UNIVERSIDADE DE BRASÍLIA
FACULDADE CIÊNCIA DA INFORMAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA INFORMAÇÃO

Multimodal Information Architecture and Artificial Intelligence: applicability and architectural models

George Hideyuki Kuroki Júnior

Advisor: Prof. Dsc. Cláudio Gottschalg-Duque

Brasília
2023

George Hideyuki Kuroki Júnior

Multimodal Information Architecture and Artificial Intelligence: applicability and architectural models

Proposes Multimodal Information Architecture models. Check the applicability of these models in artificial intelligence problems

Advisor: Prof. Dsc. Cláudio Gottschalg-Duque

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

KK96m

Kuroki Júnior, George Hideyuki
Multimodal Information Architecture and Artificial Intelligence: applicability and architectural models / George Hideyuki Kuroki Júnior; orientador Cláudio Gottschalg-Duque. -- Brasília, 2023.
387 p.

Tese(Doutorado em Ciência da Informação) -- Universidade de Brasília, 2023.

1. Multimodal Information Architecture. 2. Artificial Intelligence. 3. Architectural Applications. I. Gottschalg Duque, Cláudio, orient. II. Título.

Universidade de Brasília

FOLHA DE APROVAÇÃO

Título: "Multimodal Information Architecture and Artificial Intelligence: applicability and architectural models "

Autor (a): George Hideyuki Kuroki Júnior

Área de concentração: Gestão, Organização e Comunicação da Informação e do Conhecimento

Linha de pesquisa:

Tese submetida à Comissão Examinadora designada pelo Colegiado do Programa de Pós-graduação em Ciência da Informação da Faculdade de Ciência da Informação da Universidade de Brasília como requisito parcial para obtenção do título de **DOUTOR** em Ciência da Informação.

Tese aprovada em: 17 de fevereiro 2023.

Dr. Claudio Gottschalg Duque (presidente)

Dr. Marcio de Carvalho Victorino (PPGCINF/UnB)

Dr. Richard Marciano (University of Maryland)

Dr. Anderson da Silva Soares (UFG)

Dr. André Porto Ancona Lopez (suplente)

sei! assinatura eletrônica

Documento assinado eletronicamente por **Claudio Gottschalg Duque**, **Pesquisador(a) Colaborador(a) Pleno(a) da Faculdade de Ciência da Informação**, em 06/03/2023, às 16:55, conforme horário oficial de Brasília, com fundamento na Instrução da Reitoria 0003/2016 da Universidade de Brasília.

sei! assinatura eletrônica

Documento assinado eletronicamente por **Anderson da Silva Soares**, **Usuário Externo**, em 10/03/2023, às 08:00, conforme horário oficial de Brasília, com fundamento na Instrução da Reitoria 0003/2016 da Universidade de Brasília.

sei! assinatura eletrônica

Documento assinado eletronicamente por **Richard Marciano**, **Usuário Externo**, em 10/03/2023, às 16:25, conforme horário oficial de Brasília, com fundamento na Instrução da Reitoria 0003/2016 da Universidade de Brasília.

sei! assinatura eletrônica

Documento assinado eletronicamente por **Marcio de Carvalho Victorino**, **Professor(a) de Magistério Superior da Faculdade de Ciência da Informação**, em 10/03/2023, às 16:29, conforme horário oficial de Brasília, com fundamento na Instrução da Reitoria 0003/2016 da Universidade de Brasília.

Referência: Processo nº 23106.010057/2023-39 SEI nº 9259593

Joi:
"Mere data makes a man.
A and C and T and G
The alphabet of you.
All from four symbols.
I am only two: 1 and 0"
K:
"Half as much, but...
... twice as elegant, sweetheart."
(Fancher, Green and Dick, *Blade Runner 2049*)

Abstract

Assertiveness and effectiveness of a set actions in an information context depends on the subject ability to produce and adapt representations about reality. The process of building these representational models begins with the proper combination of learning methods and the selection of what is presented to this subject, whom perception capacity is limited. With regard to machine learning, it appears that Computer Science historically adhered to the syntax of the relations between Computational Subject and Object of analysis: it produces algorithms for calculating incidence and proximity between the properties of texts, images, sounds and others perceptible forms of manifestation. The following work aims to position Multimodal Information Architecture as the counterpart of Information Science in the semantic study of manifestations to be presented to a Computational Subject in its development of an artificial intelligence neural network.

Keywords: Multimodal Information Architecture, Artificial Intelligence.

Resumo

A assertividade e propriedade das ações de um sujeito perante um contexto informacional depende da sua capacidade de produzir e adaptar suas representações sobre a realidade. Construir estes modelos representacionais parte da combinação entre métodos de aprendizagem aliados ao que se apresenta a este sujeito que, por sua vez, possui capacidade limitada de percepção. No tocante ao aprendizado de máquinas, verifica-se que a Ciência da Computação se ateve até então à sintaxe das relações entre Sujeito Computacional e Objeto de análise: produz algoritmos para cálculo de incidência e proximidade entre as propriedades de textos, imagens, sons e outras formas de manifestação perceptíveis. O trabalho aqui apresentado visa posicionar a Arquitetura da Informação Multimodal como a contrapartida da Ciência da Informação no estudo semântico das manifestações a serem apresentadas ao Sujeito Computacional o qual se quer desenvolver uma rede inteligente.

Palavras-chaves: Arquitetura da Informação Multimodal, Inteligência Artificial, Lógica Modal.

List of Figures

List of Tables

Abbreviations and acronyms

ACC: Accuracy

AI: Artificial Intelligence

ANN: Artificial Neural Network

BERT: Bidirectional Encoder Representation for Transformers

BOW: Bag-of-Words

CBOW: Continuous Bag-of-Words

CNN: Convolutional Neural Network

DAG: Directed Acyclic Graphic

DBN: Deep Belief Network

DNN: Deep Neural Network

DCNN: Dynamic Convolutional Neural Network

DL: Deep Learning

GDI: general Definition of Information

GDR: Generalized Delta Rule

GPU: Graphic Processing Unit

HTML: Hypertext Markup Language

IDF: Inverse Document Frequency

KO: Knowledge Organization

KOP: Knowledge Organization Processes

LIS: Library and Information Science

LMS: Least-Mean Square

LSTM: Long Short-Term Memory

MB: Megabit

MIA: Multimodal Information Architecture

MCTI: Brazilian Ministry for Science, Technology and Innovations

ML: Machine Learning

MUC: Message Understanding Conferences

NER: Named Entity Recognition

NLP: Natural Language Processing

NLTK: Natural Language Toolkit

NN: Neural Network

NNLM: Neural Network for Language Modeling

NSE: Neural Semantic Encoder

PTM: Pré-trained Language Model

SL: Supervised Learning

SSL: Self-Supervised Learning

RD&I: Research, Development and Innovation

RBM: Restricted Boltzmann Machine

RCS: Reinforcement Control System

RNN: Recurrent Neural Network

TC: Text Classification Task

UG: Universal Grammar

UL: Unsupervised Learning

VDCNN: Very Deep Convolutional Neural Network

QA: Questions and Answers

WaC: Web-As-Corpus

XOR: Exclusive-Or function

Contents

1 Introduction

Organization and knowledge are two concepts with intimate relation within Information Science. Hjørland (2008) proposes two intersection points on these concepts, which can be divided into a technician view and scientific view. As the author said:

> In the narrow meaning Knowledge Organization (KO) is about activities such as document description, indexing and classification performed in libraries, bibliographical databases, archives and other kinds of "memory institutions" by librarians, archivists, information specialists, subject specialists, as well as by computer algorithms and laymen. KO as a field of study is concerned with the nature and quality of such knowledge organizing processes (KOP) as well as the knowledge organizing systems (KOS) used to organize documents, document representations, works and concepts. Library and Information Science (LIS) is the central discipline of KO in this narrow sense (although seriously challenged by, among other fields, computer science). (HJØRLAND, 2008, p. 86).

At the time that Hjørland referred to Computer Science as a challenger, it developed a new paradigm to face the limitations of machines.

Hinton, Osindero and Teh (2006) proposed an algorithm capable of enabling machines to get in the universe of semantic inferences in a given context of analysis. The authors describe the existence of several *"hidden layers"* within the universe of beliefs (or knowledge), which would make learning difficult. In their own words:

> Learning is difficult in densely connected, directed belief nets that have many hidden layers because it is difficult to infer the conditional distribution of the hidden activities when given a data vector. Variational methods use simple approximations to the true conditional distribution, but the approximations may be poor, especially at the deepest hidden layer, where the prior assumes independence. Also, variational learning still requires all of the parameters to be learned together and this makes the learning time scale poorly as the number of parameters increases.(HINTON; OSINDERO; TEH, 2006, p. 1527.).

It gave birth to one of the definitions of *Deep Learning*, and started the pursue for how deep and hidden layers must be in order to achieve an interpretation, not only of syntactic propositions, but semantics as well. Even though advances were noted on Computer Science, a critical issue addressed by Hjørland still an open topic:

> There exist many separated communities working with different technologies, but very little research about their basic assumptions and relative merits and weak sides. The problem is not just to formulate a theory, but to uncover theoretical assumptions in different practices, to formulate these assumptions as clearly as possible in order to make it possible to compare approaches.(HJØRLAND, 2008, p. 87.).

Wason (2018) undertakes a survey of initiatives on the use *Deep Learning* in semantic problems which presented significant results compared to human performance. The author defines two primary requirements for any system that aims to achieve such a task: first, the ability to recognize and process complex patterns, just like the human brain; second, the need for a great deal of information. (WASON, 2018, p. 701)

The author also cites that machine learning initiatives grew particularly well in effectiveness from 2006 (more precisely after Hinton, Osindero and Teh (2006), she states) and, since then, has been massively used in a wide range of domains such as voice recognition, no matter the sound source; recurrent neural networks; handwriting recognition; deep belief networks; auto-encoders; acoustic modeling; classification feature detectors; calligraphy synthesis; language modeling; improvement and development of models among others. She adds that some features identified on these techniques facilitate their use, such as acting in a highly complex environment, separating information from noise; train algorithms through examples to identify patterns and integrate the information into some kind of visual display; perform data analysis to reveal patterns and valuable information; being able to easily classify unstructured data through Convolutional Neural Networks or Deep Belief Networks; and imitate the human brain through artificial neural networks and progressively learn to solve a given problem in a human manner. (WASON, 2018, p. 702)

All these forms of action are linked to a learning method that uses non-linear modules organized in several layers that transform information from a lower level of abstraction, layer after layer, into abstract higher levels. As the transformation process develops, these layered networks are capable of learning complex functions without resources given by humans: the phenom occurs automatically, through a generalized learning procedure. Nonetheless Wason (2018) mentions that this scientific branch is still far from an error-free approach, having a wide range of challenges to overcome.

LeCun, Bengio and Hinton (2015) mention a modality called *Supervised Learning* as being the most common for machine learning activities. This supervision takes place in the neural network training process based on a large amount of inputs from the object to be learned. For example, a large amount of images of cars, people and dogs, if the purpose of the image classification network is to identify such objects. Large amount of bank transaction records in order to classify customer into risk levels. It appears, therefore, that variety and quantity of different representations has a great influence on the intended result.

Among all scientific fields that has Information as object of study, Physics may have shown one of the greatest dilemmas on attempting to manipulate and order this concept. A theory called *Maxwell's Demon*, developed by the physicist and mathematician James Clerk Maxwell in 1867, initially questioned the Second Law of Thermodynamics, which states that the entropy of a closed system tends to increase over time, until it reaches a maximum value. Entropy, in this case, would be analogous to the concept of disorder – molecules endowed with more heat would freely mix within the system among molecules of less heat.

According to Maxwell, the Second Law would only have statistical applications. He proposes the existence of an "intelligent microscope being", equipped with a thermal insulation "door" between two closed systems with a considerable temperature difference. To avoid entropy increase on both systems, this "being" would control the output of more "agitated" molecules (with greater energy, producing more heat) to the lower energy environment, thus maintaining thermal differences, bypassing the Second Law.

This restrictive premise of Information Management can be adapted to the context of Information Science. The never ending production and assimilation of information and knowledge on multiple scientific environments detected by Vannevar Bush (1979) surpassed academical limits – any relationship between two beings can be registered, collected, cataloged, classified and retrieved, whether in documents, books or memories of whom experienced a phenomenon.

In this sense, any *Order* imposition must focus on the main elements intended by the intelligent agent who manipulates the information set in question. That is, any intended order will be bound to all precepts assumed by the subject that operates the transformation, imprinting his perceptions about the stimuli perceived with crucial relevance of the informational environment to which he was introduced.

Therefore, the natural search for relevance and the wide variety of stimuli presented in a given phenomenon are key factors on the concept of Multimodal Information Architecture, defined by Kuroki Jr. (2018) as

onstruction and distinction of Architectural Worlds, through the assumption of Relational Models grouped by space-time contexts of correlated or uncorrelated Information States. (KUROKI JR., 2018, p.108)

Where Architectural Worlds are nothing more than *Modes*, as conceived by (KRESS, 2009): social and cultural resources produced to construct meaning. In a deeper perspective, the author differentiates two spaces of analysis while constructing Architectural Worlds: relationships perceived in a space-time context, and real objective relationships, but not relevant to the subject at a particular space-time context.

2 Problem, Objectives and Justification of the Research

2.1 Problem

Undeniable are the advances made by Computer Science on conceiving pattern recognition and learning tools. Over the years, technological development barriers that limited information processing and storage have quickly disappeared, following Moore et al. (1965) law. In this sense, considering a technical implementation scope, little importance has been given to information volume and spectrum used on learning networks: as voluminous and broader the sample is, greater tendency to assertiveness is postulated.

In a slightly different manner, long texts have always been a great challenge for Deep Learning algorithms: not only assembling the necessary quantity of examples is a difficult task, but designing methods and algorithms to gain intelligence from these examples also is (WASON, 2018; MINAEE et al., 2021).

Information Science seems to be sidelined on discussions about conception methods for these networks and their ways of handling information when compared to identified patterns. It became a mere consumer of techniques though having on its corpus a body of knowledge that can contribute on artificial intelligence development.

Looking forward, the problem to be explored in this thesis is how to develop theoretical-practical foundations based on Knowledge Organization approaches, implementable in *Deep Learning* paradigms using Multimodal Information Architecture as a theoretical framework, and apply these techniques on text classification problems?

2.2 Objectives

2.2.1 Main Objective

To design theoretical-practical constructs for an Information Management approach based on Multimodal Information Architecture, which can be implemented in neural networks based on *Deep Learning* techniques.

2.2.2 Specific Objectives

(a) Identify possible assumptions or directives from the concept of Multimodal Information Architecture that can collaborate on the development of artificial intelligence networks.

(b) Propose structuring rules, correlations and definitions for the construction of Information Architecture products aiming artificial intelligence needs.

(c) Apply the proposed ruling scheme to a problem that can be treated through a network based on *Deep Learning*.

2.3 Justification

Seems unjustifiable for Information Science to relish on Artificial Intelligence practices made up exclusively by Computer Science paradigms with no technical or methodological involvement in deeper levels. Specially when considering recent contributions on relevance analysis by Multimodal Information Architecture.

Although computation capacity continues to grow as Moore et al. (1965) states, it is reasonable to consider that the appearing of new record categories, that is, the increasing complexity of analysis as the range of perceivable phenomena grows, can overcome all advancements. The infinity of contexts formed by wide variety of Subjects facing the same situation makes the task always greater than available resources in the specific cases. Denying the fact would be the same as admitting that all problems have the same resources available while pursuing resolution, which is unrealistic.

Not only computational processing power can be listed as critical constraining factor on developing artificial intelligence, as adjusting all learning variables to as much general instances of a particular problem requires considerable quantity of examples of it. Facing this issue, the quality and variety of data is another facet to be addressed.

Hardware availability, high volume and high quality data can be classified as the perfect scenario of development, but it may not be assumed as an imperative condition. In matter of fact, the lack of it may be the perfect set up for Multimodal Information Architecture to operate better results on low expectations contexts.

3 Methodology

3.1 Research classification

Aiming proper classification for the research intended is an optimal manner for aligning expectations on its results. Three categories will be used on this matter: according to purpose, nature and methodological approach:

– **On purpose**, it is an explanatory research according to Bhattacherjee (2012), as it seeks explanations of observed phenomena, problems, or behaviors, pursuing answers to "*why*" and "*how*" type of questions. It attempts to "connect the dots" in research, by identifying causal factors and outcomes of the target phenomenon.

– **According to the Methodological Approach**, a quantitative method according to Cresswell (2003) seems to be the appropriate choice. The investigator primarily uses post positivist claims for knowledge development (cause and effect reasoning, reduction to specific variables, hypotheses, use of measurement and observation, theory testing), employs strategies of inquiry such as experiments and surveys, collecting data with predetermined instruments that yield statistical values.

– **According to the Nature of the Research**, an applied research is proposed according to the view of Kothari (2009) which explains that it aims at finding a solution for an immediate problem facing a society or an industrial/business organization.

3.2 Research Method

Intending to observe impacts of Multimodal Information Architecture on learning results of an artificial intelligence network, is proposed a comparative analysis of effectiveness between two subsets of data extracted from the same database, considering unaltered both collection time and human classification of results (treated as an *expert classification*). The difference between them resides in how each subset is conceived:

S_1. Formed by a gathering of randomly chosen attributes (like columns on a spreadsheet) from the main data set considering either completeness of each instance or volume (quantity) of information on each attribute;

S_2. produced and separated after a simple relevance analysis based on Multimodal Information Architecture methods.

The assertiveness of a neural network is measured by its accuracy rate (for some authors also called error rate, as one can be obtained from the other by simple percentage complementary), and its value will guide the analysis.

Two key aspects with critical impact on learning effectiveness will be kept unaltered: network architecture (number of layers, type of propagation flow) and activation function.

The first simulation, hereinafter called **pre-conditioned**, would consider a spectrum of data empirically defined as more significant based on volume and completeness as stated before. Accuracy rate will be registered for posterior comparison.

The second simulation, hereinafter called **post-conditioned**, will apply an organization method at the informational context to be processed. A particular definition on the concept of order, according to Abbagnano (2015), is any kind of relation between objects that may be expressed by a rule. To construct this set of rules, it is proposed the guidance of a methodological path of World View based on M^3 created by Van Gigch and Moigne (1989), an adaptation of Thomas Kuhn (2003) ideas described on *The Structure of Scientific Revolutions*:

> The Information Systems (IS) discipline lacks a paradigm to guide its work. A metasystemic approach is taken to explore a metatheory with potential to develop into a paradigm for the discipline. Sources of knowledge, the object of study, representative metaphors, activities, methodologies, and purposes of the schools of thought which constitute the discipline are reviewed and discussed in an effort to define the paradigm.(Van Gigch; MOIGNE, 1989, p.128)

The proposal adopts an knowledge construction procedure with three levels that keep intimate relationships between them: a metaphysical level, prior to the formalization of the object; the level of the object of knowledge itself and the level of application of the constructed knowledge, expressed through figure 1.

The first level, called meta-level, aims to define epistemological bases to construct knowledge. It proposes a set of postulates about reality and takes an epistemological position that will serve as a platform for key issues to be addressed at lower levels. As said by Van Gigch and Moigne (1989):

> the metalevel formulates and solves the metamodeling problem of the discipline. It is influenced by the assumptions and worldviews (inputs) of its actors and produces paradigms and metaphors (outputs) which are used by the science of IS inquiring system at the object level of inquiry (Van Gigch; MOIGNE, 1989, p. 129.).

The second, or scientific inquiry level, presents research theories and practices to delineate the problem and its likely explanations. Aiming at defining explanatory constructs of reality as well as probable theorems resulting from them, Van Gigch and Moigne (1989) list the most critical classes:

a. Person/psychological type;

b. Type of problem;

c. Organizational context;

d. Evidence/presentation mode;

e. Logical basis;

f. Rationality.

On the third level, or praxis, resides technology development founded on theories and theorems produced on the scientific level. It aims to design methods and tools to guide the subject of knowledge actions in the domain of the problem.

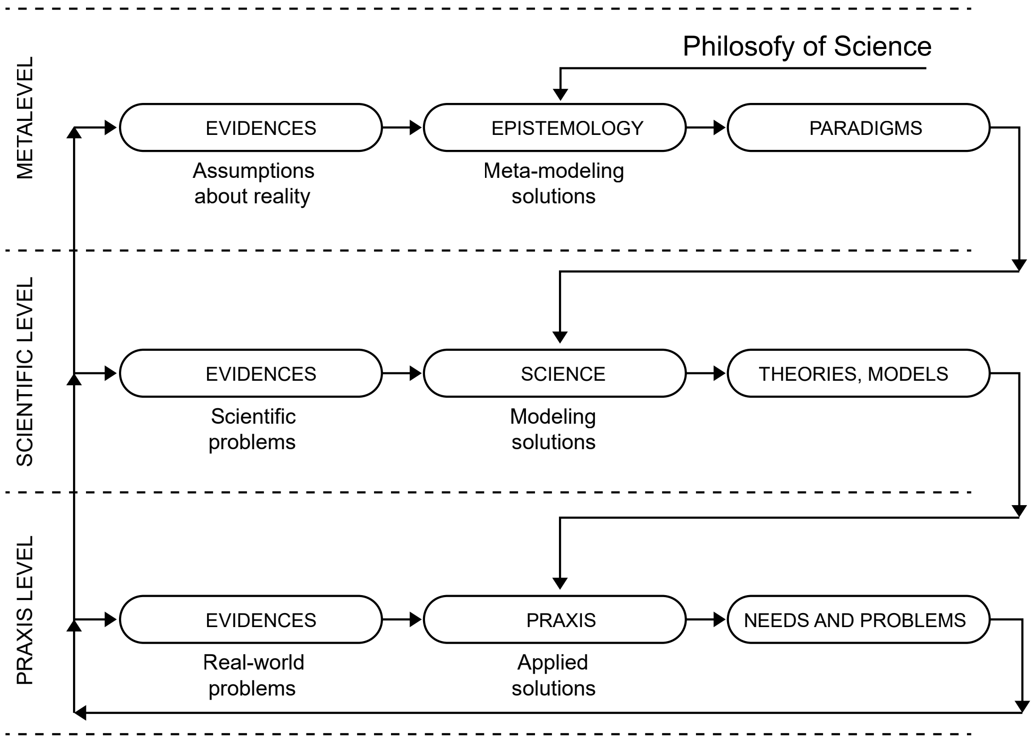


Figure 1 – (M^3) Metamodeling methodology

Source: Adapted from Van Gigch and Moigne (1989)

This thesis will adopt the methodological path presented as follows:

1. At the **epistemological level**, to undertake a pursue for Artificial Intelligence, Artificial Neural Networks and Natural Language Processing origins and development, as well as Multimodal Information Architecture worldview, its guidelines for analyzing reality and its possibilities for modeling contexts.

2. At the **scientific level**, to define guiding constructs for grounded analysis of informational contexts, based on the worldview obtained at the epistemological level. The

constructs classes proposed by van Gigch and Pipino (1986) will initially be divided into two strands of study:

Table 1 – Initial division of study of construct classes proposed by van Gigch and Pipino (1986)

Group 1	Group 2
Person/Psychological type	Evidence/Presentation mode
Type of problem	Logical basis
Organizational context	Rationality

Fonte: Adapt from van Gigch and Pipino (1986)

3. At the **technological level**, based on the constructs defined in groups 1 and 2, along with the possible relationships identified between these definitions, an information configuration will be proposed towards effectiveness gain on artificial intelligence analysis on a given problem.

3.3 Data sampling techniques

The objective of this research is to apply theoretical-practical constructs based on Multimodal Information Architecture in artificial neural networks. In this sense, it seems more appropriate the use of primary data inherited from data collection methodologies, aimed at valued analysis with semantic classification on boolean variables such as "yes/no ", “approved/failed” or any excluding dichotomous pair. This choice is justifiable given that the research result is not based on how raw data is obtained, but on the configuration records that compose the informational context to be analyzed.

Kothari (2009) divides the ways of obtaining primary data into two macro-categories: questionnaires and experiments. Also differentiates the two categories according to the following view:

> An experiment refers to an investigation in which a factor or variable under test is isolated and its effect(s) measured. In an experiment the investigator measures the effects of an experiment which he conducts intentionally. Survey refers to the method of securing information concerning a phenomena under study from all or a selected number of respondents of the concerned universe. In a survey, the investigator examines those phenomena which exist in the universe independent of his action. (KOTHARI, 2009, p.97)

In this sense, data obtained through surveys will be preferred, since it seeks to identify a common behavioral pattern, not the particular impressions of an individual.

3.4 Data collection methods

All data comes from collecting analyses carried out by several groups of individuals (specialists on each semantic domain) where assembled in data sets according to temporal distinction. Semantic boolean classification was used as section 3.3 described.

3.5 Data analysis methods

The applied character of this research aims a statistical test based on *Inferential Analysis*, as defined by Bhattacherjee (2012), which defines them as statistical procedures used to reach conclusions about associations between variables. They differ from descriptive statistics as they are explicitly designed to test hypotheses.

The aforementioned author cites a technique called Two-Group Comparison, which compares post-test results of a treated group and those obtained on a control group after inserting a variable in the informational environment of the treated group. A simple example given is the impact on the performance of students who enroll in a special mathematics program compared to those who remain restricted to a traditional curriculum.

The assertiveness results of the neural network will be compared before and after the construction of an informational configuration obtained through an architectural model based on multimodal information.

4 Relevant concepts for a theoretical model

As a first step towards designing a Multimodal Information Architecture model applied to Artificial Intelligence, it is necessary to review themes that are related to the desired objective. Identifying MIA's assumptions suitable to machine learning context is a primary task.

In the same sense, current context of development and applications of Artificial Intelligence are also characterized as a primordial part of the study, as well as a chronological verification aiming to verify practices that were eventually discontinued.

Additionally, the inclusion of other relevant scientific or philosophical currents throughout the research is not ruled out. As an example, it can be noticed, from the beginning, some epistemological and scientific developments made by some linguistic currents in text processing activities.

4.1 Intelligence and Artificial Intelligence

Can machines think (TURING, 1950)? Such question has become increasingly complex with the dissemination and evolution on how ease became implementing algorithms that aim to simulate man actions.

The history of this challenge, at technological implementation levels, began with McCarthy et al. (2006) when the authors proposed a study aiming to demonstrate that all aspects of intelligence, as well as learning processes, can be described so precisely that a machine could be constructed to simulate this phenomenon through the use of language, conceiving abstractions and concepts in order to improve itself (MCCARTHY et al., 2006, p. 12). For this, they proposed some aspects to be faced:

i. Automatic computers, capable of executing machines job automatically. The major obstacle is not lack of machine capacity, but the inability to write programs taking full advantage of what we have;

ii. Self-improvement machines, as the probability of intelligence may grow with the capacity of finding better solutions for variations of the problem;

iii. Calculation complexity measurement methods, in order to avoid the need to calculate all probabilities on a given problem;

iv. Linguistic generalizations, as a large portion of humans thought may consist of dealing with words, reasoning rules and conjecture analysis;

v. Neural nets that can be arranged to form concepts;

vi. Methods for constructing abstractions;

vii. Controlled randomness and creativity of thoughts.

The list proves to be extremely heterogeneous, so that the number of scientific disciplines that can be permeated while attempting to solve these issues gradually increases over the course of human technological development. The authors coined the term Artificial Intelligence to name the unborn object. It is undeniable that Computer Science has a leading role in the development of this activity where such artificiality is materialized through mathematical equations recorded in an electronic device. However, some of the questions listed by McCarthy et al. (2006) are not addressed by the area.

Linguistic generalizations, syntax simplifications and the necessary apparatus to achieve these new rules seem to be closer to Linguistics, just as the concept of neuron networks would be closer to neuroscience. Two postulates remain rarely discussed: how to produce methods for constructing abstractions? In this case, are these abstractions fragments of real contexts or mental projections? Finally, how would randomness control and creativity of thoughts would be on machines?

A humankind characteristic that separates it from all other beings is the ability to transcend simpleton relationship with the world: the power to interpret and apprehend attributes makes it possible to draw inferences about things, modify contexts and build models of reality. In the same sense, McCarthy and Hayes (1969) suggest that the machines ability to act intelligently would be linked to the quality of the given general representation model of the world, in terms of defining which manifestations would be interpreted. Thus, regardless the general definition of intelligence, the authors propose to define an intelligent entity if it has an adequate model of the World, capable of answering a wide variety of questions based on that model and be able to apprehend additional information from the context and perform actions on it, according to goals and abilities. (MCCARTHY; HAYES, 1969, p.12)

The problem of developing artificial intelligence was divided into two questions: an epistemological one, which deals with world representation; and a heuristic one, which deals with resolution mechanisms based on available information. McCarthy (1981) addresses the epistemological issue, stating that:

> The epistemological part of AI studies what kinds of facts about the world are available to an observer with given opportunities to observe, how these facts can be represented in the memory of a computer, and what rules permit legitimate conclusions to be drawn from these facts. It leaves aside the heuristic problems of how to search spaces of possibilities and how to match patterns.(MCCARTHY, 1981, p. 459.)

A method to produce a reasoning program for machines would be based, initially, on a model of reality. The question then would be deciding how to build this model: conceiving a simplified structure of the world (or sub-sampling on such degree that would be possible to represent all characteristics of the selected subset), and all changes that may occur on it, either being on informational or ruling contexts, also considering the relations between them.

Such definition makes the task too comprehensive, whereas expressing knowledge about the totality of the world, objectively speaking, may be considered impossible. This paradox was addressed by McCarthy and Hayes (1969), whom make a comparison with the understanding of gas dynamics. For the authors, such conceptualization is linked to the representation of the entity *"gas"* as a large portion of molecules moving in space, which would make it possible to derive mechanical, thermal, electrical and optical properties of gases. In the same sense, the entity's physical state at a given moment could be determined by position, velocity and excitation of each molecule. However, this representation would be nothing more than an abstract representation of reality. As the authors says:

> However, we never actually determine the position, velocity or excitation of even a single molecule. Our practical knowledge of a particular sample of gas is expressed by parameters like the pressure, temperature and velocity fields or even more grossly by average pressures and temperatures. From our philosophical point of view this is entirely normal, and we are not inclined to deny existence to entities we cannot see, or to be so anthropocentric as to imagine that the world must be so constructed that we have direct or even indirect access to all of it.

The authors describe three categories of adequacy for representations of the world in artificial intelligence: metaphysical, epistemological and heuristic. At the metaphysical level, a representation would be adequate if the conceived abstraction could actually exist without contradicting facts that are pertinent to common knowledge. Such adjustments would have a primary role in the construction of general theories, so that they start from high-level abstract conceptions of things. On an epistemological level, a representation would be adequate if it can be used by a person or machine to express facts perceived to describe aspects of the world. At a heuristic level, a representation would be adequate if the logical argumentation processes carried out to solve a problem can be expressed through a language.

It is possible to perceive the authors efforts to conceive a logical path to build the necessary means to build machines that can relate to the world around them in an analogous way (or at least a simulacrum) to how mankind does. Initially, a vision of the world is built, an automaton intelligence is inserted in this vision and it is allowed the ability to relate with the projected entities, expressing its apprehensions through a language.

In another aspect of facing the fundamental problem of Artificial Intelligence, Russell and Norvig (2010) characterize two dimensions of analysis. The first deals with thinking and reasoning; a second deals with acting and behaving. These two dimensions can be measured in terms of human behavior simulation or an ideal reasoning of a situation, according to a certain world representation context that the machine has, that is, "*to do the thing rationally*". From these definitions, four approaches to AI have been proposed.

The first refers to acting humanly, focused on meeting what Allan Turing (1950) proposed – a computer that could impersonate a human when interrogated by another human. Several skills would be needed to succeed in this endeavor: sensory simulations such as sight and touch; motor reflexes like moving objects; iterative cycles of learning through experience; representation and manipulation of acquired knowledge; linguistic coordination. For Russell and Norvig (2010), these skills represent six Artificial Intelligence disciplines:

- Natural language processing;
- Knowledge representation;
- Automated reasoning;
- Machine learning;
- Computer vision;
- Robotics.

The authors complement stating that these six disciplines cover most of Artificial Intelligence discipline, however, emphasize that most scientists have not devoted considerable time in attempting to provide a solution to Turing's test, focusing on establishing intelligence principles at the expense of duplicating an intelligent exemplar, even comparing a bird flight with a machine one: The quest for "*artificial flight*" succeeded when the Wright brothers and others stopped imitating birds and started using wind tunnels and learning about aerodynamics. (RUSSELL; NORVIG, 2010, p.3)

The second approach addresses thinking humanly. It is more focused on simulating the ways of human reasoning, that is, understanding how humans think in order to produce programs that simulate the same method. Collect as much information as possible about mind process and then design computerized mental models. Russell and Norvig (2010) defines Cognitive Sciences as the encounter of computer models of AI with experimental techniques from psychology, aiming to build verifiable theories about the human mind. This interdependence of areas (Cognitive Sciences and AI) has proofed to be beneficial for both: a new computerized implementation (a new algorithm) enables a new testable mental model, that provides feedback to AI which can provide new computerized test modalities.

The third approach deals with thinking rationally, formalized through propositional logic. Computer programs are extremely efficient in solving problems that may be represented by logical arguments. The authors cite two obstacles to this approach: first, the difficulty of translating informal knowledge into formal expressions, especially when this knowledge is not an absolute truth (which, in theory, would apply to the majority of human spectrum); second, there is a big difference between solving a problem in theory then in practice. The amount of computational resource needed to calculate scenarios of small hundreds of variables can be unfeasible in some aspects (RUSSELL; NORVIG, 2010, p.4).

Finally, the approach of acting rationally. The authors differentiate it from the previous one by describing that acting rationally goes beyond the simple correctness of inferences. In fact, consistency of problem solution on a logically validated reasoning system is part of a rational action. However, correct inferences are not always rational. In some scenarios, it is not possible to obtain correctness proof of a rationalized solution.

Russell and Norvig (2010) characterizes Cognitive Sciences as the intersection between computer models of AI with experimental techniques from psychology in order to build verifiable theories about the human mind. This interdependence of areas (Cognitive Sciences and AI) has proven to be beneficial for both: a new computerized implementation (a new algorithm) enables a new mental model, providing feedback to AI, enabling a new computerized test approach.

Russell and Norvig (2010) stand as in favor of an agent-rational approach at the expense of other views due to its generalization capacity (it covers both logically verifiable and empirically verified inferences), as well as its flexibility and adaptation to scientific development in detriment of approaches based on human behavior or human thinking.

Minsky (1961) also stands for the non-existence of a general theory of intelligence. The author mentions that this conclusion was obtained through conversations with several authors who deal with Artificial Intelligence. Instead, the author says that it would be possible to divide the problem into five areas: search, pattern recognition, learning, planning and induction.

From the perspective of search, an artificial intelligence cannot be driven into mapping all possible cases to obtain a solution to a given problem. "Trial and error" strategies, other than being not practical, are mathematically questionable: problems now considered trivial, such as building an intelligence to play chess, should consider up to 10^{120} movement possibilities. Such magnitude of calculus cannot be based solely on computational power to obtain a solution. It is necessary to have strategy and update methods according to the results obtained.

As an alternative to "trial and error strategy", Minsky (1961) argues in favor of using heuristic concepts. For the author, this approach would need constant improvement of general performance in problem solving, which also would increase success rate on other situations, although with an acceptable failure rate. Considering this scenario, pattern recognition should also provide intelligence with the ability to classify problems into categories associated with more effective resolution methods. These classification methods can be as simple as comparing the current question with previous ones, as far as property analysis and identification through testing. Therefore, the pursue for relevant properties to build a pattern recognition system becomes extremely important. In this sense, a pattern can be defined as a set of properties identified in a group of objects that make each instance of the group suitable to similar and useful treatment.

The same heuristic analysis strategy fits the concept of learning systems. For Minsky (1961), when starting to solve a new problem, it is common and understandable to use strategies already known and proven to be effective, used in apparently similar contexts. One way to implement this systematic procedure is through the use of reinforcement models to achieve right (or better, at least) decisions. The definition of reinforcement relates to increased use (or disuse) of certain aspects of the learning system. This is not about penalizing the system, but about increasing relevance where is due or extinguishing a step based on ineffectiveness.

However, when facing real problems, it must be considered that the situation presented is not always atomic, in a sense that it is not complete on it self: in general, a problem is composed of several interrelated sub-problems and, in addition, it is common that each instance has different characteristics and properties that makes it unique when compared to the whole. To investigate and solve the totality of components identified in real situations can culminate in a metaphysical discourse about the very nature of the problem, that is, in order to reach a resolution for an insignificant fraction of reality, we are forced to build a complete model of the world. In order to face this limitation, the development of some method of problem evaluation and selection through each step of the search for solutions becomes imperative. Only a small part should be selected, based on criteria such as complexity estimation and relevance analysis of the part that will be treated in comparison to the global problem.

For Minsky (1961), it does not matter how many heuristic layers of analysis, selection of strategies or definition of key problems are implemented in an intelligence system: every operation of this entangled and complex configuration will come down to a series of routines placed in a sequenced and repetitive mode that, at their lowest level, are resumed to trivial operations of comparison. The last level described by the author is based on induction and inference strategies, where these operations must be tested in real and complex situations. At that time, the most promising approach was called "grammatical induction", based on manipulating *languages*, defined as:

> (...) We will take language to mean the set of expressions formed from some given set of primitive symbols or expressions, by the repeated application of some given set of rules; the primitive expressions plus the rules is the grammar of the language (MINSKY, 1961, p. 27)

4.1.1 Rational agents

On philosophy, one definition of agent is whom or what takes initiative of acting or from whom or what an action emanates or derives from. Is part of a dichotomous relation with a patient, which is whom or what undergoes the action (ABBAGNANO, 2015, p.21).

An agent, for Russell and Norvig (2010), would be any entity that can be synthesized by means of receptors that perceive the environment in which it is inserted and act through actuators. The entity itself only becomes an agent when is given the ability of perceiving the environment. Throughout interactions, it starts to produce a sequencing of perceptions, which will guide other choices for each new perceived manifestation. It is never influenced by unknown insights.

Perception, according to Abbagnano (2015), has three main definitions. First, in a very general manner, characterize it as any type of cognitive activity; second, in a more restricted way, defines it as a cognitive act or function to which a real object is presented; finally, in a more technical sense, designates an operation determined by humans while perceiving an environment. A stimuli interpretation, either constructing their meaning or revisiting it (ABBAGNANO, 2015, p. 876-880).

An agent who perceives an environment is able to make inferences about what was perceived. It emphasizes the recurrent aspect of the expression "constructing or revisiting of meaning" – no apprehension or designation of meaning is absolute when manifestation review can occur at each new interaction with any object. Even considering that no inference made can be taken as absolute, it is necessary to have guidelines for analysis and apprehension of perceptions, otherwise, the intelligent aspect of this agent would be questionable.

Although the definition of the concept of Intelligence is not the scope of this thesis, it is necessary to define at least what can be characterized as intelligent. Engelbrecht (2007), aimed at a definition of Computational Intelligence, describing it as the ability to understand, comprehend and benefit from experiences, to make assumptions with intelligence, having the ability to think and reason. The author exemplify keywords that describe other aspects of intelligence: creativity, competence, conscience, emotion and intuition (ENGELBRECHT, 2007, p. 3). Two words stand out as possible ways to design agents that express intelligence: whom or what can *think* and who or what can *reason*. At first sight, these terms may seem analogous, but philosophically, there is some differences between them.

The concept of *Thought*, for Abbagnano (2015) has four distinct meanings:

a. Any mental or spiritual activity;

b. Any activity obtained from intellect or reason, in opposition to senses and will. On one hand distinguishes from sensitivity, on the other hand from practical activity;

c. Discursive activity, as part of propaedeutic sciences (arithmetic, geometry, astronomy and music), and a path towards intuitive thinking;

d. Intuitive activity identified with the object, a direct view of what is intelligible.

For the author, the most traditional view of thought comprises definitions on "b." and "c.". Through the combination of both, there is an understanding that the concept is linked to a specific activity of a certain faculty of the human spirit, more precisely the one which higher (non-sensible) cognitive activity belongs. (ABBAGNANO, 2015, p. 874)

On a slightly different path, *Reasoning* is defined as any procedure of inference or proof; but also can be expressed as an argument, conclusion, inference or analogy (ABBAGNANO, 2015, p. 982).

To obtain practical results through a rational agent (which acts according to reasoning) appears to be more viable than obtaining results from thinking agents (which acts on the basis of a human faculty). Rationality implies a systematic analysis of efficiency on the actions taken to reach a goal, based on factual evidences and then, describing the experience observed through general explanatory principles.

Russell and Norvig (2010) are consistent with this concept, however, leave as an open question the definition of what an ideal performance would be. For the authors, every performance measure is linked to the desired results under certain environmental circumstances, exemplified in an situation where the rational agent is a vacuum cleaner:

> A rational agent can maximize this performance by measuring by cleaning up the dirt, then dumping it all on the floor, then cleaning it up again, and so on. A more suitable performance measure would reward the agent for having a clean floor. For example, one point could be awarded for each clean square at each time step (perhaps with a penalty for electricity consumed and noise generated). As a general rule, it is better to design performance measures according to what one actually wants in the environment, rather than according to how one thinks the agent should behave. (RUSSELL; NORVIG, 2010, p. 37)

Four questions would be taken into account in defining rationality at a given moment: performance measurement or goal that defines success; prior knowledge of the agent; actions that the agent can perform; sequence of perceptions apprehended by the agent up to that moment. Thus, the authors conclude on a definition of rational agent as:

> For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has. (RUSSELL; NORVIG, 2010, p. 37)

Clearly the authors focus their construction of rational action based on sets of perceptions. First, the one presented right before the action to be taken. Second, those that have been recorded as knowledge acquired by the agent. About the first grouping, two actions performed by the agent are of extreme relevance: exploring the environment and collecting information from it. If these actions are repeated in a cyclic manner, it should culminate in a process of apprehension of perceived configurations, even though there is relevant prior knowledge about the environment in which the agent is acting.

It would be questionable and flawed to assume that an agent's rationality would accomplish complete understanding of all relationships between things and entities placed in a given environment. It's a measure of *autonomy* compared to what can be called *pre-assumed perceptions* or *projected meta-environment*. As an example, such assumptions can come from a previous experience (the agent has already performed actions within the environment in question) or through indirect projection (description and/or modeling acquired from another rational agent or conscious being).

An agent who deprives himself of reconstructing his perceptions and interpretations about them as they change or reveal themselves, becomes poorly adaptive and extremely linked to non-dynamic models: he lacks learning autonomy. This autonomy enables better adapting to a wider range of contexts and problems.

This gathering of sequenced learning is what Russell and Norvig (2010) call *knowledge base*, characterized by the set of assertions (or sentences) expressed through a language that represent some aspect of the world. The authors defend that such constructions are not exclusively conceived through sensory mechanisms, but also through reasoning processes that operate the internal representations of knowledge.

This base, as already described, cannot be static: it is imperative to insert (or produce) new sentences, as well as to search for a sentence that has already been apprehended in the list of assertions. These actions relies on inference processes, where, through sentence analysis, one concludes (or constructs) the other, every time the agent is required to have some type of interaction with the world or with another agent.

4.1.2 Environments

From the perspective listed by McCarthy and Hayes (1969), the environment in which an agent performs its actions refers to the epistemological part of the AI. At the other hand, the counterpart of analysis performed by the agent can be fulfilled in the form of heuristic strategies cited by Minsky (1961). Although the construction of an intelligence system is focused on better problem solving strategies, it cannot be denied that every action of this intelligence will suffer great influence from the context that surrounds it.

McCarthy and Hayes (1969) list some questions to be considered about the environment that surrounds an intelligence system:

a. What kind of generalized representations of reality will enable the capacity of incorporating specific observations on the knowledge base, as well as new scientific laws as they are discovered?

b. Besides the representation of the physical world, what other types of entities should be considered? For example, mathematical systems, goals and states of knowledge;

c. How should observations be used to get knowledge about the world and how other kinds of knowledge are obtained? In particular what kinds of knowledge about the system's own state of mind need to be provided?

d. In what kind of internal notation should system knowledge be expressed?

For the authors, designing systematized intelligence must start with the representation model of a world to act in. Characteristics such as its structure (the objective reality of things), how this structure manifest itself in pieces of information and the rules that guide eventual changes must be clearly presented for the agent. Therefore, a world representation has epistemological adequacy if an agent can used it to express facts about aspects of the real world. Aiming to formalize the concept, an epistemologically adequate system is proposed through definitions expressed in mathematical expressions.

Beginning with the expression **Situation**, a particular situation "*s*" is characterized by the configuration of all components of the Universe at a given moment in time. In turn, "*Sit*" would demonstrate the totality of all situations, as a space-time continuum. Considering that describing the exact configuration of the Universe is impossible, any model should only describe facts about certain situations. Through sequencing of facts it would be possible to approximate projected reality to objective reality, however, being necessary to provide instruments that makes possible to apprehend at least part of the information about situations.

Fluents are functions whose domain is the space "*Sit*" of situations (MCCARTHY; HAYES, 1969, p. 470). If the scope of the function is *(true, false)*, it is called *propositional fluent*. If the scope is "*Sit*", it is called *situational fluent*. Fluents usually are the values of functions. For example, a fluent over *rain* at a certain location can be expressed as $rain(x, s)$, if in fact it is raining at location (x) in situation (s). The notion of *fluent* enables the construction of expressions with mathematical representation:

> $At(p, x)(s) \wedge raining(x)(s)$ to describe that person p is at location x and it is raining at x; or
>
> A most common mathematical expression $At(p, x, s) \wedge raining(x, s)$; or
>
> $[At(p, x) \wedge raining(x)](s)$ to express that fluents can perform logical operations on fluents, such as;
>
> $(f\ [\mathrm{op}]\ g)(s) = f(s)\ [\mathrm{op}]\ g(s)$.

Causality can be expressed through a fluent $F(\pi)$ where π is itself a propositional fluent. Such function describes a situation s that will be followed, at some point, by another situation that satisfies the fluent π. Causal relationship, for example, can be expressed through a logical equation:

$$\forall x.\forall p.\forall raining(x) \wedge at(p,x) \wedge ()outside(p) \rightarrow F(wet(p)) \tag{4.1}$$

which expresses that for anyone p at a place x where is raining, and the person p is in place x on open air, fluent F conditions implies that person p get wet.

Actions are intentional or non-intentional goals of a Subject p that can be summarized through the fluent

$$[result](p, \sigma, s) \tag{4.2}$$

which expresses that Subject p can perform an action σ in a situation s. The value of this equation is the situation when p carries out σ, starting on situation s. Actions can be concatenated, sequenced or even canceled by each other, giving rise to the concept of **Strategy**.

Strategies are defined as the combination and/or sequence of actions, as long as they are procedural remote calls: an instance does not suffer interference from another during its execution. This notion of independence allows variables to influence only the operational set in which they act, not being directly transmitted from one action to another, for example, a variable called s in an action σ has no procedural relationship with the same variable s in action ω. In a broader sense, strategies are generally used to achieve a particular goal. By selecting the best action to be undertaken in a situation s, a rational agent implements a strategy according to the objective. At this moment, concepts of **Knowledge** and **Ability** take place on the discussion.

To illustrate the context, McCarthy and Hayes (1969) propose a situation where a person p is supposed to open a safe. If he/she has a set of potential keys c that open the vault sf, this strategy could be expressed through the following expression.

$$has(p, k, s) \wedge fits(k, sf) \wedge at(p, sf, s) \rightarrow open(sf, results(p, opens(sf, k), s)) \tag{4.3}$$

It would be necessary for person p to have the **Ability** *open* within his list of possible actions in context s. On the other hand, if the safe is not opened by a key, but by means of a numerical code such as *28101983*, the need for the **Knowledge** *code* to carry out the action would be added to the context of *open*, which would lead to an expression as follows

$$open(sf, result(p, open(sf, code(sf)), s)) \tag{4.4}$$

It opens up a discussion of how to formalize complex notions of knowledge, time, obligations and many others expressions that are typical of the human mind. McCarthy and Hayes

(1969) mentions a common sense where the definition of Artificial Intelligence would be the study of methods for constructing programs that could predict sequences formed from simple classes of laws, in some cases, probabilistic. The model, according to the author, seems to be metaphysically adequate, but epistemologically inadequate. What is known about the world is divided into knowledge groups comprising aspects about it, taken separately and with low level of interaction. Another relevant point is that human knowledge is not used to predict determined sequences of experiences: as situations are presented, context perception changes. An example would be to observe a person predicting the result of a sports competition match: he/she does not conceive individuals perceptions of each visual sensation of the context; all predictions made take into consideration factors that help to better describe a plausible behavior in the future, for example, a performance decrease of one team due to the apparent fatigue of players. However, this kind of reduced analysis still have highly probabilistic nature and tend to be poorly formalized.

Addressing this questions, the authors mention Modal Logic as an initial path to avoid implications paradoxes, for example, a false proposition implying any proposition, obtained through truth tables. The initial idea was to segregate truths into two categories: necessary and contingent. A proposition would not simply be judged as true or false, but as a measure of possibility. Saul Kripke (1963) led the development of a theory that implements propositional calculus to deal with the concepts of necessary and possible truths. He proposed the existence of several coexisting worlds, resulting in the possibility of different truth-values for the same proposition. Thus, a value is necessary when it has to be true in all possible worlds.

4.2 Interactions between Agents and Environments

The construction of intelligence models demands collecting perceptions and inferences made by an agent about the environment, expressing them through a language (MCCARTHY; HAYES, 1969; MCCARTHY, 1981; MINSKY, 1961; RUSSELL; NORVIG, 2010). The problem could be divided into three study lines. The first deals with agents and their structures to obtain, apprehend and produce meaning, with focus on psychoneurological issues of knowledge. A second, dedicated to analyzing objective properties of the world, the description of entities and relationships between them. Finally, a third one which would aim to formalize and rule how agents describe the world. The description makes the object of study so embracing, that almost every aspect of human knowledge, its development and relations would be considered. Therefore, it is not objective of this work to deal with detailed general physiological aspects of a group of individuals, nor conceiving axiomatic systems to describe properties of entities, neither obtaining agnostic linguistic models applied to any form of human expression.

4.2.1 Neural Networks: a model of brain functions

According to Rosenblatt (1961), as "brain model" we shall mean any theoretical system which attempts to explain the psychological functioning of the brain in terms of known physics and mathematics, and known facts of neuroanatomy and physiology. (ROSENBLATT, 1961, p.3). In essence, this theoretical model could be described as a system with known properties, prepared to analyze situations which the main goal is to incorporate essential characteristics of a system with unknown or ambiguous properties.

As being a simplification of the brain, the author suggests two fundamental issues to be considered. First one is that the essential properties of the brain are topology and dynamics of impulse propagation through a network of neurons. It is from correctness of connection mapping and part states (neurons) that a better description of the whole is obtained. Taken individually, parts do not show psychological functions such as memory, attention or intelligence. These properties are materialized through organizing and activating the network as a whole. The second one is considering the existence of a common sense that enable the conception of devices with information manipulation capabilities typical of biological networks, independent of any living force for their functioning.

Two approaches to construct theoretical models of the brain would stand, based on different views of how it would work. One strand propose a definition based on classic digital computers, endowed with algorithms that are inherent to its own existence. Other based on non-algorithmic methods that bears little resemblance with logical or mathematical rules (typical of digital implementation), tending to rely on probabilistic analysis and adaptability of values.

The first approach Rosenblatt (1961) denominate as *Monotypic Models*, which generally starts with defining, as assertive as possible, how accurate the model should be, that is, primarily defining what has to be achieved and how assertive must the strategy be. A system then will be built based on these parameters, using modular switching devices which are analogous to biological neurons in their properties, forming a nerve-net. As first application of the concept, the author points out to McCulloch and Pitts (1943), who adduce that any psychological phenomenon can be understood and analyzed in terms of activities in a network formed by binary state devices (all-or-none), where each part of the network could be mathematically represented. In their own words:

> Many years ago one of us, by considerations impertinent to this argument, was led to conceive of the response of any neuron as factually equivalent to a proposition which proposed its adequate stimulus. He therefore attempted to record the behavior of complicated nets in the notation of the symbolic logic of propositions. The "all-or-none" law of nervous activity is sufficient to insure that the activity of any neuron may be represented as a proposition. Physiological relations existing among nervous activities correspond, of course, to relations among the propositions; and the utility of the representation depends upon the identity of these relations with those of the logic of propositions (MCCULLOCH; PITTS, 1943, p. 116-117.).

Rosenblatt (1961) names the fundamental unit of an axiomatic representation of psychological behavior through a network of logical propositions as "*McCulloch-Pitts neuron*", which according to his saying, was the basis of several brain models. Additionally, lists five attributes of this neural representation:

a. Neuron activity is an "all-or-none" process;

b. A certain fixed number of synapses must be excited within the period of latent addition in order to excite a neuron at any time, and this number is independent of previous activity and position on the neuron;

c. The only relevant delay within the nervous system is synaptic delay;

d. The activity of any inhibitory synapse would absolutely prevent excitation of the neuron at that time;

e. The structure of the net does not change with time.

The Monotypic model, despite being logically well grounded, lacks relevance on achieved goals. According to Rosenblatt (1961) after the first flood of proposed models, further progresses were disappointingly trivial, and returns seemed to diminished rapidly. The promised biological "explanations" were particularly reduced. In the writer opinion, there were at least five main reasons for this:

(1.) Lack of sufficiently well defined psychological functions as a starting point. The approach requires essentially full knowledge of input-output relations for the behavior of an organism, and such knowledge is not available for any biological species .

(2.) Disparity between the designed neural constructions and the known conditions of neuroanatomy and neuroeconomics. The number of neurons needed usually exceeds the number of neurons in biological nervous systems, and logical organization usually requires precision in its connections, a need that does not apply to the brain. In some cases, a wrong connection can make the system inoperable;

(3.) Models fail to produce general laws of organization. A monotypic model is usually overdetermined, corresponding at best to a biological phenotype rather than a species as a whole. Its specification in the form of a detailed "wiring diagram" often misses a multitude of details. Generally, unique solutions for the proposed functions are lacking and a huge variety of models can be generated seeming to solve the same problem equally well. Therefore, unless the system is actually tested against its biological counterpart, nothing is gained from a detailed construction of the model except further confirmation of an existence theorem that is already well established.

(4.) Models lack predictive value. Once a specific model has been proposed, further analysis can reveal little beyond what is included in the initial functional description.

(5.) Models are not biologically testable in detail. Specific connections in nervous tissue cannot be traced with sufficient precision to prove whether or not a specific wiring diagram is implemented accurately. Consequently, models are destined to remain purely speculative unless histological techniques are improved to a highly improbable degree of definition.

A second strand of brain models cited by Rosenblatt (1961) is called *Genotypic*. Unlike Monotypic models, where the properties of network components (neurons) as well as axiomatic relationships and network topology are specified in detail, Genotypic models describe only the components in detail, leaving the network organization to an hybrid model: part is specified at the beginning, another part obeys constraints and probabilistic distributions that generate system classes at the expense of specific designs. Thus, the author states that:

> The genotypic approach, then, is concerned with the properties of systems which conform to designated laws of organization, rather than with the logical function realized by a particular system. (ROSENBLATT, 1961, p. 20.)

Greater emphasis is then placed on statistical properties of systems classes generated through applying organization rules rather than pure propositional symbolic logic. Another difference is regarding the objectives of each implementation. In monotypic models, functional properties of the network are the starting point for construction – beginning from a functional description in order to achieve an accuracy value. In genotypic models, functional properties of the network are actually the objective to be achieved, starting from a physical model of statistical values of classes. Ordinarily, psychological functions detailing is not mandatory during model conception. In fact, it was expected that genotypic models could collaborate in search for answers to open psychological problems.

The use of genotypic models at the time Rosenblatt (1961) developed his theory was prejudiced due to the absence of mathematical implementation tools that could adapt to the proposed problems. Hence, further development of monotypic models at that time is justifiable, since they are based on mathematical tools already unfolded in computers and system controlling theories. Relevant influence of Psychology and Neuroanatomy in genotypic models is mentioned, in detriment of Engineering based sciences recalling that, throughout the 19th century, advances in descriptive anatomy supported further studies on the plasticity of the nervous system to reorder neurological functions due to injuries to the cortex. However, a gap could be noticed in the production of theoretical models for the representation of this brain organization.

4.2.2 Rosenblatt (1961) Perceptron

Rosenblatt (1961) proposes a genotypic implementation called *Perceptron*, endowed with a memory mechanism that allows learning stimuli in different types of experiments. In each case, the object of analysis is an experimental system composed of the *Perceptron*, a defined environment and a training procedure or facilitator. With the results of this analysis, it would be possible to compare them to experiments carried out on animals or humans in order to obtain comparable parameters between the model and compatible psychological functions.

The author explains that the objective is not to construct a detailed copy of any particular nervous system, but a simplification designed to allow the study of rules and relationships between network organization, environment arrangement and psychological performance. Additionally, *Perceptrons* can be part of deeper networks in biological systems, as well as making it possible to ask questions and obtain relevant answers about certain types of contexts, hypothetical memory mechanisms and neuron models.

The design is based on some physiological and psychological foundations already pacified at the time:

(a.) **neurons and nerve impulses:** each unit of a neural network, regardless of whether it is specialized in a particular function or not, is activated through a conjunction of impulses coming from other units. Only when a certain level of "excitation" is accumulated (a threshold level) in a pre-trigger area, an electrical signal is then sent to neighbor units. There is also variance in the propagated signal strength based on the frequency at which the triggers are activated, which can also increase or decrease the activation sensitivity of these connections.

(b.) **Topological network organization:** the human brain consists of a network formed by billions of neurons of all types, where each sensory modality (vision, hearing, touch, etc.) has a corresponding predominant area in its organization, that is, even if a signal from a particular sensory receptor propagates through several neurons groupings, there is an area on the cerebral cortex more prone to receive and process this signal.

(c.) **Function location:** even though there is a distribution of motor functions and mental faculties (intelligence, religiosity, combativeness, among others) to certain areas of the cerebral cortex, there is also plasticity on these functions and faculties where rearrangement can be made due to injuries and lacerations compromising these areas. However, sensory functions such as vision apparently do not share the same plasticity and adaptability.

(d.) **Innate computational functions:** certain behavioral patterns and perceptual abilities present in several species are due to computational mechanisms that are often unknown.

(e.) **Learning and forgetting phenomena:** knowledge apprehension processes observed in psychological experiments tend to demonstrate some general laws of learning, however, for building brain models such concepts do not seem to be useful, being more assertive to treat each problem individually.

(f.) **Field phenomena in perception:** any perceptual phenomenon, in any sensorial modality, will be influenced by the environment in which phenomenon takes place.

(g.) **Choice-mechanisms in perception and behavior:** selection of attention and psychological focus are largely determined by the context in which behavior occurs, as well as being influenced by objectives or purposes of the intelligent being in question, taking them as guidelines for sub-decisions that contribute to an activity.

(h.) **Complex behavior sequences:** rational behaviors or goals, such as driving a car or conducting research on a subject, can be considered a group of coordinated actions directed by attention and psychological focus. Their ordering through a sequence alternated with decisions can be viewed as computer programs.

It is clear that the author made an effort to conduct his definition based on non-controversial ideas on multiple areas of human knowledge, while also recognizing that certain themes could not yet be considered more than speculation at the time, such as definitions (g.) and (h.). The author reasons that part of the exposed questions do not exert any influence on the construction of a perceptive unit that allows emulating intelligence. A clear example is the spatial location of the mental faculty of memory, where the implemented view in the proposal would remain committed to simplifying the model, in order to enable the reduction of extensive knowledge for its realization, in the words of the author:

> The question of localization is of less importance for a functional model of the brain than is the question of mechanism; as long as we assumes that it is the network topology, rather than the actual anatomical position of neurons, which is important in determining the brain's logical properties, there is no reason for requiring that a brain model resembles the biological system in its spatial organization. The indirect implications of the different theories of localization are of considerable importance, however. For one thing, the view that the brain contains its memories in a widely dispersed, intermingled form, suggests a mechanism in which the same cells participate in a great variety of different, and perhaps totally unrelated, memory organizations.(ROSENBLATT, 1957, p.59)

The author acknowledges that, combining (c.) and (h.), it is likely that the phenomenon of information storage and recovery in the human nervous system involves coordinated activities of several parts of a complex structure. However, the model should focus on defining which psychological properties can be emulated in systems which memory is located in a single set of connections, with minimal structural differences. This simplification would also affect other issues addressed.

Following the path of simplification and pursue of predominant characteristics of psychological properties, an informational structure of things in the brain would not present itself as an isomorphic representation of the object in question: it does not inquiry a way of conceiving an exact depiction of an external expression, either in physical (leading to an unreasonable isomorphism) or logical form (leading into mapping ways to deconstruct and reconstruct properties). The focal point is on strategies to obtain a model consistent with psychologically proven phenomena able to adapt perceptual mechanisms such as silhouette recognition, completeness of partially presented objects, among other observable phenomena in theories such as *Gestalt*.

In order to carry out this simplification, Rosenblatt (1961) lists an extensive list of terminological definitions to better support his purpose. For the purpose of this thesis, it is important to highlight:

Definition 1. A signal may be any measurable variable, such as a voltage, current, light intensity, or chemical concentration. A signal is typically characterized by its amplitude, time and location.

Definition 5. A signal transmission network is a system of signal generating units, linked by connections.

Definition 6. A sensory unit (S-unit) is any transducer responding to physical energy (e.g., light, sound, pressure, heat, radio signals, etc.) by emitting a signal which is some function of the input energy. The input signal at time t to an S-Unit s_i from the environment W is symbolized $\alpha'_{Wi}(t)$. The signal which is generated by s_i at time t is symbolized $s'_i(t)$

Definition 7. A simple S-unit is an S-unit which generates an output signal $s_i = +1$ if its input signal α_{Wi} exceeds a given threshold O_i, and O otherwise.

Definition 8. An association unit (A-unit) is a signal generating unit (typically a logical decision element) having input and output connections. An A-unit a_j responds to the sequence of previous signals c'_{ij} received by way of input connections c_{ij}, by emitting a signal a'_j.

Definition 9. A simple A-unit is a logical decision element, which generates an output signal if the algebraic sum of its input signals, α_i is equal or greater than a threshold quantity $\Theta > 0$. The output signal a'_i is equal to $+1$ if $\alpha_i \geqslant \Theta$ and 0 otherwise. If $a'_i = +1$, the unit is said to be active.

Definition 10. A response unit (R-unit) is a signal generating unit having input connections, and emitting a signal which is transmitted outside the network (i.e. , to the environment, or external system). The emitted signal from unit r_i will be symbolized by r'_i.

Definition 11. A simple R-unit is an R-unit which emits the output $r'_i = +1$ if the sum of its input signals is strictly positive, and $r'_i = -1$ if the sum of its input signals is strictly negative. If the sum of the inputs is zero, the output can be considered to be equal to zero or indeterminate. (A physical unit which oscillates in response to a zero signal would have the required properties.)

Definition 15. The phase space of a network is the space of all possible memory states, for a given network. In general, if there are N variable-valued connections in the network, the phase space may be represented by a region in Euclidean N-space, each coordinate corresponding to the value of one connection. The memory state of the system at any specified time can be characterized by a point in this phase space, and the history of the system by a directed line, or path, followed by this point.

Definition 16. The interaction matrix for a network of S, A, and R units is the matrix of coupling coefficients, v_{ij}, for all pairs of units, u_i and u_j. If there is no connection from u_i to u_j, is defined as zero. Specifying an interaction matrix is equivalent to specifying a point in the phase space.

Definition 17. A perceptron is a network of S, A, and R units with a variable interaction matrix V which depends on the sequence of past activity states of the network.

Definition 18. The logical distance from unit u_i to $u_j$is equal to the number of connections in the shortest path by which a signal can be transmitted from u_i to u_j.

Definition 19. A series-coupled perceptron is a system in which all connections originating from units at logical distance d, from the closest S-unit terminate on units at logical distance $d + 1$ from the closest S-unit.

Definition 20. A cross-coupled perceptron is a system in which some connections join units of the same type (S , A or R) which are at the same logical distance from S -units, all other connections being of the series-coupled type.

Definition 21. A back-coupled perceptron is a system in which at least one A or R unit at a distance d_1 from the closest S-unit is the origin of a connection back to an S-unit or to an A -unit at a distance $d_2 > d_1$ from the closest S-unit; i.e. , this is a system with feedback paths from units located near the output end of the system to units closer to the sensory end.

Definition 22. A simple perceptron is any perceptron satisfying the following five conditions:

i. There is only one R-unit, with a connection from every A-unit;
ii. The perceptron is series-coupled, with connections only from S-units to A-units, and from A-units to the R-unit.
iii. The values of all sensory to A-unit connections are fixed (do not change with time);

iv. The transmission time of every connection is either zero or equal to a fixed constant, $\mathcal{T}$;

v. All signal generating functions of S , A , and R-units are of the form $u_i(t) = +(\alpha_i(t))$, where $\alpha_i(t)$ is the algebraic sum of all input signals arriving simultaneously at the unit u_i.

Definition 26. A <u>stimulus-sequence world</u> (or stimulus-sequence environment) is any set of stimulus sequences, each consisting of an ordered series of stimuli from the set $\mathcal{W}$. (For example, if the image of a printed word is a stimulus, and $\mathcal{W}$ consists of all words in a dictionary, then the set of all English sentences would comprise a stimulus sequence world.)

Definition 27. A response function is any assignment of R-unit output signals to stimuli in W. For a simple perceptron, the response function $R(W)$ is a vector from n elements (R_1, R_2, R_3, R_n) indicating the value of the response for each of the stimuli, S_1, S_2, S_3, S_n in the environment.

Definition 28. A <u>classification</u> is an equivalence class of response functions. Two response functions are considered equivalent if their corresponding elements agree in sign. For any perceptron with one simple R -unit, a classification, $C(\mathcal{W})$ divides $\mathcal{W}$ into two classes: a <u>positive class</u> consisting of all stimuli for which $r' = +1$ and a <u>negative class</u>, consisting of those stimuli for which $r' = -1$.

Definition 29. A <u>response-sequence function</u> is an assignment of sequences of R-unit output signals to stimulus sequences in a stimulus-sequence world. This is a generalization of the concept of a response function to include a time dimension;

Definition 30. A <u>solution</u> to a response function (or classification) is said to exist for a given perceptron if there is a point in the phase space of the perceptron such that the response R_i (specified by the function) will occur if the stimulus S_i is shown, for all S_i in $\mathcal{W}$.

Definition 31. A <u>reinforcement system</u> is any set of rules by which the interaction matrix (or memory state) of a perceptron may be altered through time.

Definition 32. A <u>reinforcement control system</u> is any system or mechanism external to a perceptron which is capable of altering the interaction matrix of the perceptron in accordance with the rules of a specified reinforcement system;

Definition 33. <u>Positive reinforcement</u> is a reinforcement process in which a connection from an active unit u_i which terminates on a unit u_j has its value changed by a quantity $\Delta v_{ij}(t)$ (or at a rate $\frac{dv_{ij}}{dt}$) which agrees in sign with the signal $u_j(t)$.

Definition 34. Negative reinforcement is a reinforcement process in which a connection from an active unit u_i which terminates on a unit u_j has its value changed by a quantity $\Delta v_{ij}(t)$ (or at a rate $\frac{dv_{ij}}{dt}$) which is opposite in sign from, $u_j(t)$.

Definition 35. A monopolar reinforcement system is a reinforcement system in which the values of all connections terminating on a unit u_j remain unchanged at time t unless $u_j(t)$ is strictly positive.

Definition 36. A bipolar reinforcement system is a reinforcement system in which the values of connections are subject to change regardless of whether the output of the terminal unit is positive or negative.

Definition 37. Alpha system reinforcement is a reinforcement system in which all active connections c_{ij} which terminate on some unit u_j (i.e. , connections for which $u_i'(t - \mathcal{T}) \neq 0$) are changed by an equal quantity $\Delta v_{ij}(t) = \eta$ or at a constant rate while reinforcement is applied, and inactive connections ($u_i'(t - \mathcal{T}) = 0$) are unchanged at time t. A perceptron in which α-system reinforcement is employed will be called an α-perceptron. The reinforcement will be called quantized if the change is a ($|\Delta v| = |\eta|$) or non-quantized if the value may change by an arbitrary magnitude.

Definition 38. Gamma system reinforcement is a rule for changing the values of the input connections to some unit, whereby all active connections are first changed by an equal quantity, and the total quantity added to the values of the active connections is then subtracted from the entire set of input connections, being divided equally among them.

Definition 39. A response-controlled reinforcement system (R-controlled system) is a training procedure in which the magnitude of η is constant, and the sign of η is entirely determined by the current response, r', regardless of the current stimulus, s. In general, unless otherwise specified, this term implies that the reinforcement is always positive (i.e., the sign of η agrees with the sign of r', in a simple perceptron);

Definition 40. A stimulus-controlled reinforcement system (S-controlled system) is a training procedure in which the magnitude of η is constant, and the sign of η is determined entirely by the current stimulus, s, and a predetermined classification, $C(\mathcal{W})$; the current response of the perceptron does not influence either the sign or magnitude of η.

Definition 41. An error-corrective reinforcement system (error correction system) is a training procedure in which the magnitude of η is 0 unless the current response of the perceptron is wrong, in which case, the sign of η is determined by the sign of the error. In this system, reinforcement is 0 for a correct response, and negative (see definition Definition 34.) for an incorrect response, or, more generally, $\eta = f(\mathcal{R}' - r')$, where $\mathcal{R}'$ is the required response, r' is the obtained response, and f is a sign-preserving monotonic function, such that $f(0) = 0$

Furthermore, it is possible to graphically describe perceptrons in a wide variety of ways, however, three types of diagrams are the most common: *network diagrams*, *set diagrams* and *symbolic diagrams*. The use of a particular diagram type depends on the level of specificity desired in the representation:

(1) Network diagrams are more complete and indicate each connection and signal unit individually. Arrows indicate the direction of signal transmission along the connections.

(2) Set diagrams represent all S-Units as a single set, connected to the set of A-Units (or association system) which is represented by a Venn diagram, which are subsets connected to different R-Units. For the author, these diagrams are useful in carrying out analyses.

(3) Symbolic diagrams only indicate the types of connections existing in a perceptron, namely, [S] to [A], [A] to [R] and [S] to [S].

Figure 2 shows the graphic representations.

Finally, the author defines the concept of experimental system, consisting of a perceptron, a world $\mathcal{W}$ with stimuli and a reinforcement control system. The latter can be an automatic regulation device (for example, a thermostat) or a human operator, capable of responding to the perceptron responses and environmental stimuli, applying the appropriate reinforcement rules, changing the perceptron's memory state.

The Reinforcement Control System can be considered a specialized part of the environment in terms of its relationship to the perceptron, although it may belong to the physical construction of the perceptron itself. In an R-Controlled System (where reinforcement orientation is through perceptron response analysis), the information channel from $\mathcal{W}$ to the R.C.S. is not functional, while in an S-controlled System (where reinforcement orientation is given through the analysis of the stimuli presented to the perceptron) the information channel from $\mathcal{W}$ to R.C.S. is not functional and, in an error correction system, both channels are essential for boost control. In digital simulation programs, the R.C.S. is the part of the program concerned with reinforcing the simulated perceptron, whereas in experiments with hardware systems, it is usually a human operator.

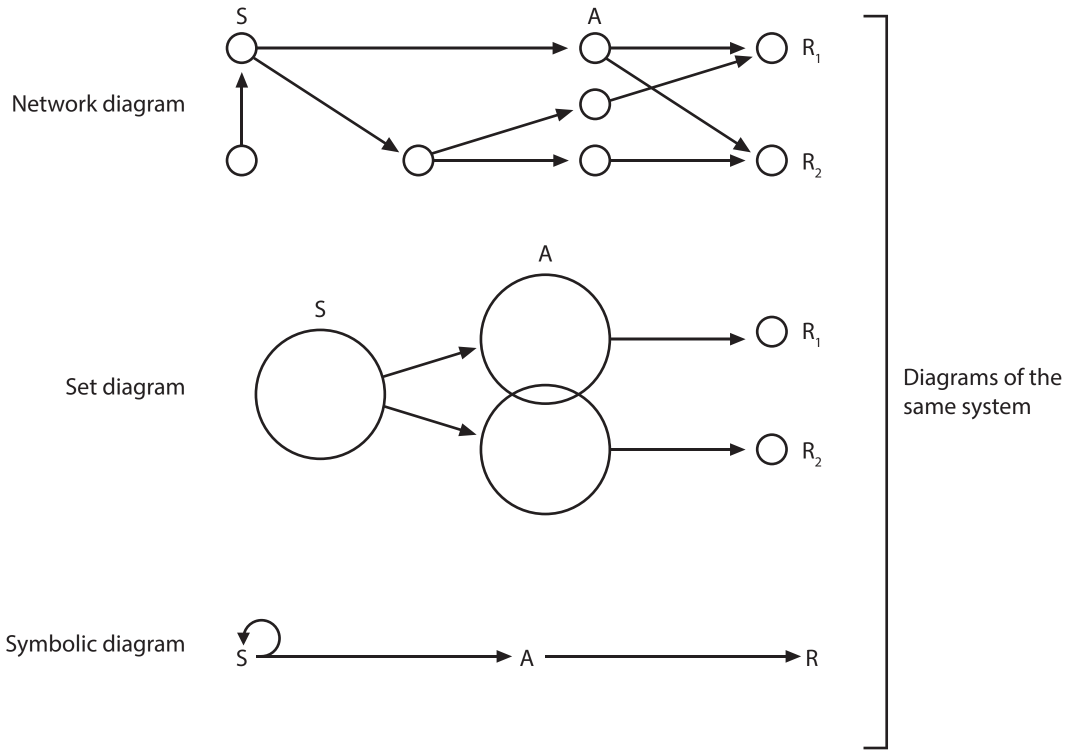


Figure 2 – Rosenblatt (1961) diagrams

Source: Adapted from Rosenblatt (1961)

An experiment involves an experimental system, a training procedure and a procedure to test the perceptron or measure its performance, expressed graphically in figure 3.

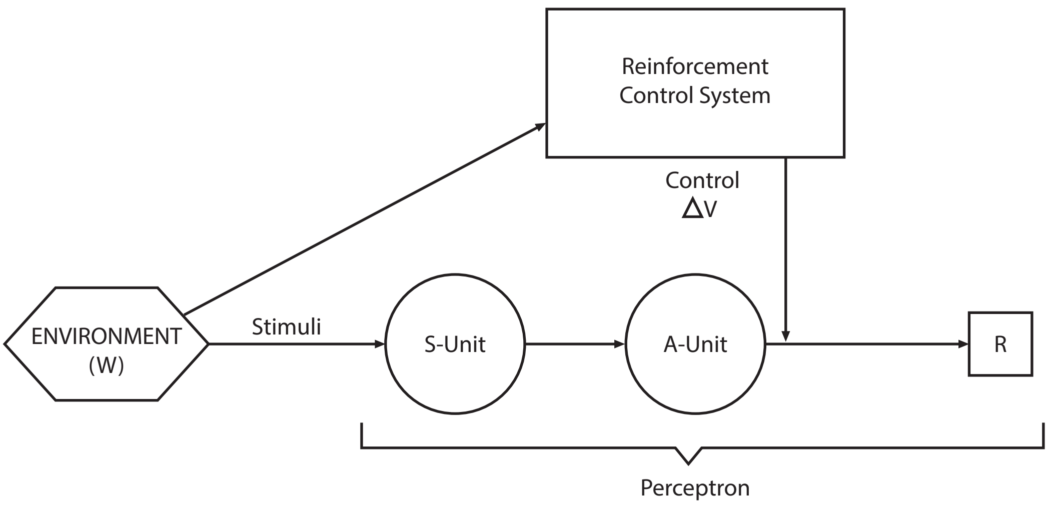


Figure 3 – Rosenblatt (1961) experimental system

Source: Adapted from Rosenblatt (1961)

Rosenblatt (1961) starts his experiments with three-layered perceptron models with serially connected units, with the topology $S \Longrightarrow A \Longrightarrow R$, that is, in each of the levels (Sensorial

- Association - Response), there are s_n, a_n and r_n units as described in the definitions Definition 6., Definition 8. and Definition 10., respectively. Afterwards, using multilayer models, increases the number of levels of association, either serially (becoming a four-layered model) or cross-coupling (retaining three layers, but with non-serial connections between some units). Figure 4 presents a generic scheme of the experiments performed by the author.

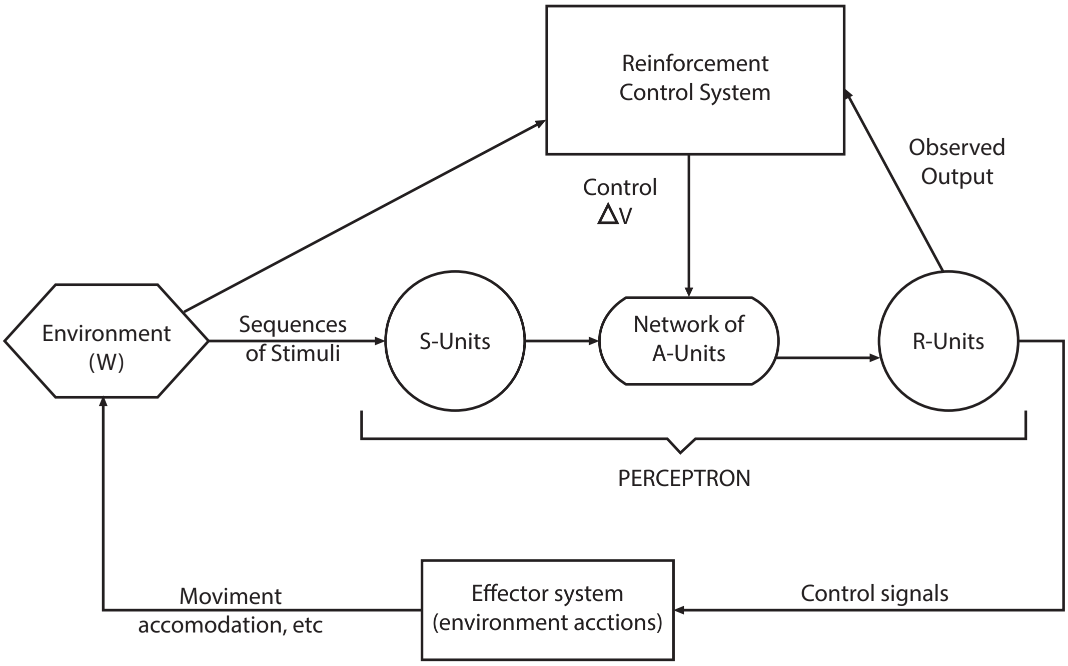


Figure 4 – Rosenblatt (1961) general model

Source: Adapted from Rosenblatt (1961)

Over 16 experiments, the author points out that as the complexity of the perceptron organization increases, new psychological properties are observed. Among the conclusions obtained, the following can be highlighted:

(1) A three-layer series-coupled perceptron is the minimal system capable of learning to discriminate arbitrary classes of patterns or sequences of stimuli. Any problem of discrimination can, in principle, be solved by this system, and any arbitrary response function can be attributed to stimuli from a given universe.

(2) The generalization capabilities of series-coupled three-layer systems are poor, and in "pure generalization" experiments (where the test stimuli have no sensory points in common with the training stimuli), there is basically no generalization capability .

(3) By means of an alpha system with reinforcement through error correction, a perceptron of three serial layers with simple A-Units and fixed pre-terminal network can always be taught the solution to any problem of classification or function of answer to which there is a solution.

(4) Four-layered and cross-coupling systems with adequate rules to modify their connection values are able to learn a group of transformations that occurred in stimulus sequences and, later, recognize the similarity of stimuli that are equivalent in the observed transformation group. This phenomenon occurs "spontaneously", without any external influence on the perceptron, other than the occurrence of stimuli.

(5) In rear-coupled perceptrons, selective attention to familiar objects in a complex field is possible. It is also possible that this perceptron selectively observes objects that move setting itself apart from its background.

(6) Several speculative models that are likely to learn sequential programs, analyze speech in phonemes, and learn "meanings" of nouns and verbs with simple sensory references have been presented. Such systems represent the upper limits of abstract behavior in the perceptrons considered at that time. They are handicapped by a lack of satisfactory "temporary memory", an inability to perceive abstract topological relationships in a simple way, and an inability to isolate significant figurative entities or objects except under special conditions.

A point to be highlighted during the experiments performed is the identification of the need for "memory" along the perceptron network, if the approach is oriented to sequential programs, that is, the subsequent step of any unit depends directly on the result from the previous transmitting unit. In this sense, the greater the complexity of the program in question, greater probability will be that a later step will depend on information extracted from an earlier step, which would generate greater storage capacity for these unit states.

4.2.3 Minsky and Papert (1988): comments on Perceptrons

In 1969, Minsky and Papert (1988) released the first edition of their work *Perceptrons*, which is described as being focused on a deeper understanding of concepts related to general theory of computation and parallel computation, getting further details on classes that make decisions through duly weighted evidence. Among a variety of the work's target readers, special remarks were addressed to psychologists and biologists who seek some kind of mathematical-computational foundation in research related to the functioning of the brain and the processing of thoughts. In addition to them, it is also directed to any audience interested in delving into pattern recognition theories.

The use of the name *Perceptron* is a recognition to the pioneering work of Frank Rosenblatt (1961), given the existence of a wide range of machines whose primary objective is similar: making decisions based on how similar (or not) an event is compared to a pattern, supported by evidence obtained through several small experiments. This foundation, although simple, is basal to construct more complex decision-making apparatus. Therefore, the term *Connectionism* is coined based on the grow and use of networks based on the *Perceptron* design, antagonistically to what was called *Symbolist*. This dichotomy was applied not only to computing, but also to writers, therapists, educators, and philosophers when it came to models of mental functions. Most people shared the characteristics of these classifications diametrically opposite, as shown in table 2.

Table 2 – Symbolist and Connectionist Dichotomy

Symbolist	Connectionist
Logical	Analogical
Serial	Parallel
Discrete	Continuous
Localized	Distributed
Hierarchical	Heterarchical
Left-brained	Right-brained

Source: Adapted from (MINSKY; PAPERT, 1988, p. viii)

This division cannot be taken as absolute, since the attributes in question can be seen as independent of each other. In their words:

> (...) the very same system could combine symbolic, analogical, serial, continuous and localized aspects. Nor do many of those pairs imply clear opposites; at best they merely indicate some possible extremes among some wider range of possibilities. And although many good theories begin by making distinctions, we feel that in subjects as broad as these there is less to be gained from sharpening boundaries than from seeking useful intermediates. (MINSKY; PAPERT, 1988, p. viii.)

First studies on perceptrons were extremely broad and voluminous, however, the vast majority suffered from scientific value, a fact accentuated by the definition of the term as "learning machines" or "pattern recognizing machines". Computer science and cybernetics surged, in their opinion, surrounded by a certain romanticism. On the other hand, collaborative work of scientific communities provided relevant contribution to its development, when considering that greater rigor and precaution could significantly slow down steps took towards improvement.

As an opposition to the vast majority of authors at the time, their work did not focused on building perceptrons or on how learning could be performed. Instead, tried to elucidate problems that would be presented to these machines. In this sense, efforts were aimed at relationships between pattern recognition activity and parallel architectures designs capable of recognizing these patterns.

Among the studies with relevant contributions, Donald Hebb (1949)'s work stands out, which is based on the principle of function distribution of perceptive faculties: processing received signals is distributed through a network, not centralized in isolated areas or disconnected from each other. Neural activity would take place through the junction of perceptive generalization, persistence of learning and attention, which, in the incidence of repeated exposure to the same stimulus by specific receptors, would form a "cluster" of cells in association areas that can act as a closed system after stimulus presentation is ceased. This prolonged permanence allow structural change of knowledge and would represent the simplest instance of a representative process (image or idea). (HEBB, 1949, p. 60)

Samuel (1959), throughout research on development of machine learning through checkers, pointed out the need to drive efforts towards designing computer programs in order to enable them to learn through experience, reducing programming effort to adapt into different scenarios of data processing. This author indicates the existence, at the time, of two distinct general methods for machine learning:

> One method, which might be called the NeuralNet Approach, deals with the possibility of inducing learned behavior into a randomly connected switching net (or its simulation on a digital computer) as a result of a reward-and-punishment routine. A second, and much more efficient approach, is to produce the equivalent of a highly organized network which has been designed to learn only certain specific things. The first method should lead to the development of general-purpose learning machines. A comparison between the size of the switching nets that can be reasonably constructed or simulated at the present time and the size of the neural nets used by animals, suggests that we have a long way to go before we obtain practical devices.f The second procedure requires reprogramming for each new application, but it is capable of realization at the present time. [p. 211.](SAMUEL, 1959)

Samuel (1959) openly admits limitations in his experiment, on the order of physical feasibility of implementing more complex and plastic neural networks. Despite such limitations, the author suggests two fundamental questions to be faced:

(1) Credit valuation: given a certain configuration of variables, how to determine contribution extension of each when a positive achievement is made?

(2) Development of new properties: if existing variables are inadequate, how can new ones be produced?

According to Minsky and Papert (1988), Rosenblatt (1961)'s implementation satisfactorily addresses the first question. Crediting each part proportionally to their contribution overcomes dispersiveness of each nuclei involved. On the authors own words:

> When the answer is obtained, in effect, by adding up the contributions of many processes that have no significant interactions among themselves, then the best one can do is reward them in proportion to how much each of them contributed. (Actually, with perceptrons, one never rewards sucess; only punishes failure. (MINSKY; PAPERT, 1988, p. xi)

For the second question, Rosenblatt (1961) provides the simplest possible answer: it would not be necessary to design new variables, if the initial supply is sufficient given the scope of the problem. As the use of perceptrons advanced, it became clear that such an approach only apply at certain circumstances.

In 1988 the authors decided to update what had been proposed nearly two decades before regarding theories related to *Perceptrons*, parallel computing, pattern recognition, knowledge representation and learning. They realized that little had been added during this time, making only a few critics on the results obtained on those years.

One of the aspects strongly addressed was the fact that perceptrons have limited learning capabilities: only lower complexity problems presented themselves as subject to pattern mapping by perceptrons at the time. Minsky and Papert (1988) characterize as lower order properties that have a linear relationship with the achieved result, that is, tendency to proportional relations, whether direct or indirect, between the property in question and the output of the perceptron. In this kind of problem, one could, in fact, create properties randomly and select those that influence the result.

These limitations to the spectrum of treatable patterns indicated to the authors the existence of unknown issues that had been little treated so far. Most theoretical studies focused only on the mathematical structure of what could be considered of common learning, culminating in theories far too general and weak in order to explain why perceptrons only identify some types of patterns. The authors defend that research focuses were wrong: it was not about identifying learning patterns, but about the own perceptron architecture, given the characteristics of the problem. Limitations occurred when there was no adequate form to represent the object in question, that is, enabling machine learning would not be limited to construct methods that allow "learning", needing also to include ways to understand the nature of the object and represent it somehow. Therefore, the authors propose two strands of analysis: “theory of learning” and “theory of representation”. Quoting:

> Perceptrons could learn anything that they could represent, but they were too limited in what they could represent. (MINSKY; PAPERT, 1988, p. 256)
>
> Multilayered networks were less limited in what they could represent, but they had no reliable learning procedure. (MINSKY; PAPERT, 1988, p. 256)

Proportionately to the impacts that such statements had on the development of perceptron use researches, there was a growing number of criticisms on how these issues were addressed, in emphasis, McClelland et al. (1986) took a tougher stance, stating that the limitations described for single-layered perceptrons by no means could be applied to more complex networks. In fact, Minsky and Papert (1988) recognize that McClelland et al. (1986)’s proposal makes part of the conclusions taken by the authors in 1969 clearly mistaken, however, remember the question of computational cost and scalability of the problems. Quoting:

> This observation shows most starkly how we and the authors of *PDP* differ in interpreting the implications of our theory. Our "pessimistic evaluation of the perceptron" was the assertion that, although certains problems can easily be solved by perceptrons on small scales, the computational costs become prohibitive when the problem is scaled up. (MINSKY; PAPERT, 1988, p. 253-254.)

Another discussion point raised by McClelland et al. (1986) is the use of a method called *Generalized Delta Rule* - GDR, which implements a way to measure the participation of each unit of analysis of the network in its success or failure when processing an input. The reasoning undertaken is based on a limitation pointed out by Minsky and Papert (1988):

> In their famous book Perceptrons, Minsky and Papert (1969) document the limitations of the perceptron. The simplest example of a function that cannot be computed by the perceptron is the exclusive-or (XOR), illustrated in Table 1. It should be clear enough why this problem is impossible. In order for a perceptron to solve this problem, the following four inequalities must be satisfied.
>
> $0 \times w_1 + 0 \times w_2 < \theta \longrightarrow 0 < \theta$
>
> $0 \times w_1 + 1 \times w_2 > \theta \longrightarrow w_1 < \theta$
>
> $1 \times w_1 + 0 \times w_2 > \theta \longrightarrow w_2 < \theta$
>
> $1 \times w_1 + 1 \times w_2 < \theta \longrightarrow w_1 + w_2 < \theta$
>
> Obviously, we can't have both w_1 and w_2 greater than θ while their sum, $w_1 + w_2$, is less than θ.

The authors propose a graphical way to present this limitation. In a geometrical map of assertions (inputs) and results (outputs), as shown in figure 5, inputs are placed at each vertex of a polygon and outputs at each internal angle of the representation. Table 3 expresses, in a structured way, the relationship between vertex and angle.

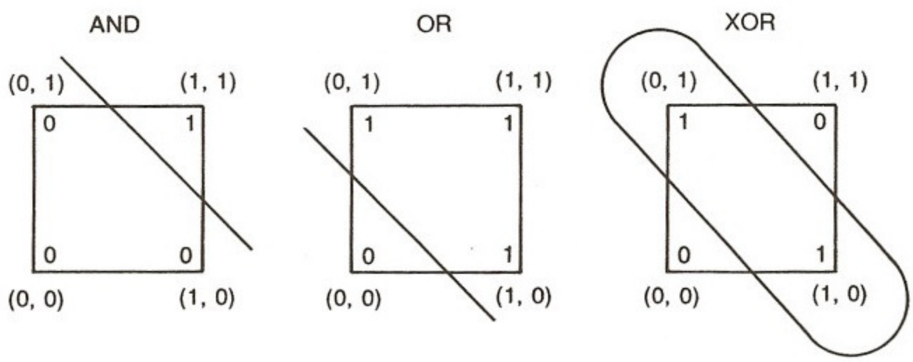


Figure 5 – Graphic demonstration of perceptrons constraints

Source McClelland et al. (1986)

Table 3 – XOR operation results

Input	Output
00	0
01	1
10	1
11	0

Source: Adapted from McClelland et al. (1986, p. 123)

A perceptron would be able to solve any function in which, based on a graphical model like figure 5, it is possible to draw a line which separates all “0” outputs on one side from all outputs “1” on the other side. The figure shows that it is totally possible for functions *AND (AND)* and *OR (OR)*, but not for *XOR*. Geometrically expressable functions which also present a graphical solution for separating results are called *linearly separable*.

For this limitation, McClelland et al. (1986) propose the following situation: a third dimension is added to the two dimensions that define the function *XOR*, which is nothing more than inserting a function *AND* between the initial two, generating table 4 below.

Table 4 – XOR with AND operation results

Inputs	Outputs
000	0
010	1
100	1
111	0

Source: Adapted from McClelland et al. (1986, p. 125)

Adding a third dimension makes it possible to insert a plane inside the cube formed from the union of vertices compared to entrances and exits values. Figure 6 below graphically expresses the assertion proposed by the authors.

Discussions could be summarized into ways of finding out which properties should be considered to solve the problem at hand, or, in short, a method for learning intermediate layers should be provided, which would be quite challenging, given that the original perceptron learning process only applies to a single layer for analysis.

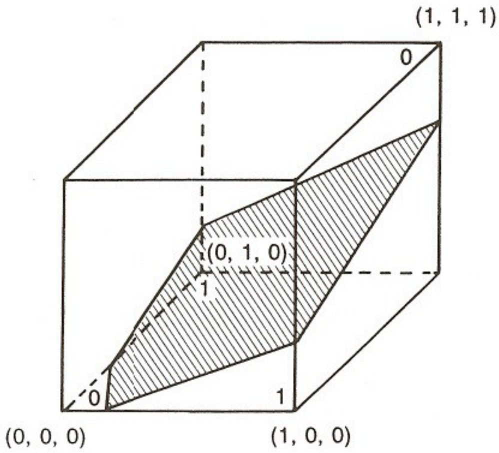


Figure 6 – XOR with AND solution according to McClelland et al. (1986)

Source: McClelland et al. (1986, p.125)

GDR uses a learning procedure called *Least-Mean Square - LMS*, which takes into account the sum of the square of the difference between the expected outputs and the outputs obtained for each input presented: that is, the total error presented is the sum of the squared deviation between the expected results and the obtained results, as shown in the expression below.

$$E = \sum_p \sum_i (t_{pi} - o_{pi})^2 \tag{4.5}$$

The objective would be to obtain a combination of values for each analysis unit relevance weight in order to reduce total error through the network. To operate such scaled reduction, *LMS* uses a method called *gradient descent*. After analyzing an input, the error produced is computed and the weight of each analysis unit is modified according to its deviation from the expected result: if it is more relevant, weight value is increased; if it is less relevant, weight value is reduced. To apply this method to a multilayer network, a technique called *Backpropagation* is used, which defines two distinct moments of action. The first processes an input in a forward propagation direction, where each unit analyzes the input and makes its prediction. Individual errors are computed and total error is obtained. The second action is to verify and adjust the contribution of each unit on the deviation, taking the opposite direction (back to the first analysis layer - hence the name *Backpropagation*) - through *gradient descent*. Figure 7 demonstrates a single evaluation of a network influence weight compared to its general error, while figure 8 shows the complexity of handling two influence weights on the general error of the same network.

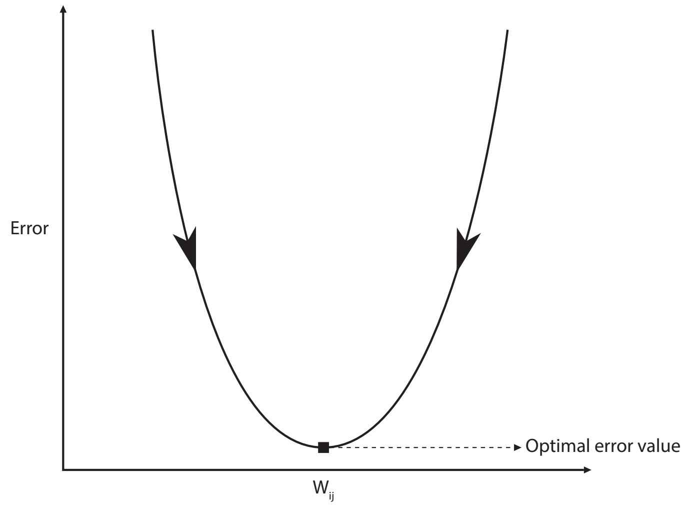


Figure 7 – Single weight error compared to general error

Source: Adapted from McClelland et al. (1986, p.127)

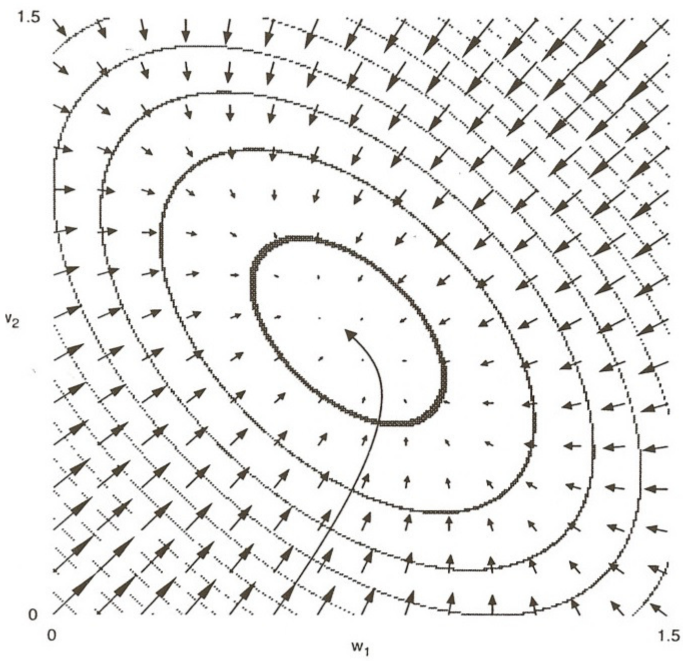


Figure 8 – Two errors weights compared to general error

Source: McClelland et al. (1986, p.129)

Minsky and Papert (1988) address the strategy under two point of views: sample variance and solution scaling. Once again they point out the issue of problem complexity dealt by McClelland et al. (1986), citing that the situations analyzed were too simplistic (as in being possible to collect and present all input stimulus configurations) and sample noise was eliminated whenever possible. They complement by stating that no person or animal is faced with a situation so simple and configured in such favorable manner that one can go through learning cycles in a fluid way (MINSKY; PAPERT, 1988, p.264). Fatefully, transcending these limits to more real situations, an exhaustive collection of stimuli would become impracticable. The alternative goes through statistical sampling which, consequently, bring noise to properties presented by each instance.

Scaling issue is based on computational cost to produce perceptrons with sufficient number of layers for certain problems. The authors do not deny that relevant order perceptrons could, in principle, represent any finite property. However, the search for mapping main characteristics of a property cannot be reduced to brute force methods: it is necessary to have a strategy for this purpose. For Minsky and Papert (1988), the examples formulated by McClelland et al. (1986) mostly deal with situations where the reduced number of variables creates an atmosphere in which it would be possible to reproduce the same result on a larger scale, which, in principle, would not be verifiable, since the exponential increase of weights to be calculated were ignored.

It is mentioned that overcoming these limitations cannot be based on developing a domain agnostic general theory for neural networks. It is necessary to carry on studies on neural networks models as specialized as possible, fitting the reality of the mental faculty they are intended to exercise. Therefore, it is deduced that the geometric recognition skill would not be transposed to another problem domain, for example, color recognition: they are different modes of visual expression. On the other hand, these specializations work together in the human brain. In the aforementioned visual context, representation of an object's image will take place through the conjunction of results from both networks, although being possible to separate them into different analysis models: questioning the object's color apart from its geometric shape.

We return, then, to the duality explained in the table 2 between Symbolism and Connectionism: which analysis system is more efficient and assertive? Symbolism is based on the construction, by a subject, of compact representations of more complex objects. By nature, it opens up the possibility of obtaining several symbols for the same object, since each subject can produce its own simplification within its mind, as well as an object representing several simplifications at the same time. Connectionism is based on the inexistence of a central element on some component. Object representation comes from a series of contributions that work together simultaneously. On the other hand, modifications in a given representation will require changes in a large number of components which, reflexively, an isolated change of a component will have minimal impact (or even none) in different circumstances.

Minsky and Papert (1988) do not position themselves for or against any system presented. Just point out that none of the alternatives proved to be decisive in solving the problems exposed. Quoting:

> This observation shows how we and the authors of *PDP* differ in interpreting the implications of our theory. Our "pessimistic evaluation of the perceptron" was the assertion that, although certain problems can easily by solved by per-

> ceptrons on small scales, the computational costs become prohivitive when the problem is scaled up. (MINSKY; PAPERT, 1988, p. 253-254.)

In this sense, the authors indicate that the notion describing the brain as a large uniform highly interconnected network of units related to one another would not be assertive. It would be more correct to interpret it as a large grouping of networks, endowed with distinct architectures and control systems. A concept called by the author as *Society of Mind*, describes what would be a large number of “agents” working together that, if taken individually, would treat no more than a minor problem. Relationships between these parts take place through multiple layers, organized by levels of abstraction: the upper layers control and manage the lower layers, up to the level where the units of the last layer specialize in micro-tasks of less relevance, that is, do not represent a relevant stimulus alone.

4.3 Artificial Neural Networks: definitions, development and applications

Minsky and Papert (1988)’s notes posed great challenges to conceiving artificial models of intelligence. Most of the limitations found can be summarized in two strands cited by Hagan, Demuth and Beale (2014), that resumed studies in neural networks:

> At least two ingredients are necessary for the advancement of a technology: concept and implementation. First, one must have a concept, a way of thinking about a topic, some view of it that gives a clarity not there before. This may involve a simple idea, or it may be more specific and include a mathematical description.(HAGAN; DEMUTH; BEALE, 2014, p.2)

For the authors, a large part of the mathematical foundation necessary for implementing algorithms that performed intelligent functions (in a practical example, computed tomography is mentioned) became available years before the computational power needed to perform such a task. The improvement of artificial neural networks depends on advancements of these two aspects: conceptual innovations and implementation development. Although the pillars are identifiable, evolution did not take place in an orderly manner. Setbacks, revisions, denials of previously consolidated theories were constant during this process. From late 1960s until part of 1980s was a period marked by lack of new ideas and computational power available for experimentation. Throughout the 1980s, both impediments got overcame and research into neural networks increased drastically. In this sense, they explain:

> Two new concepts were most responsible for the rebirth of neural networks. The first was the use of statistical mechanics to explain the operation of a certain class of recurrent network, which could be used as an associative memory. This was described in a seminal paper by physicist John Hopfield.

> The second key development of the 1980s was the backpropagation algorithm for training multilayer perceptron networks, which was discovered independently by several different researchers. The most influential publication of the backpropagation algorithm was by David Rumelhart and James McClelland. This algorithm was the answer to the criticisms Minsky and Papert had made in the 1960s.(HAGAN; DEMUTH; BEALE, 2014, p.1-4)

Another view for interest growth in the area was due to conceptual changes when approaching certain problems. Hassoun et al. (1995) cite in their preface that issues such as pattern classification, voice recognition, dialogue synthesis, adaptive interfaces between humans and complex physical systems, predictive analysis, associative memory and nonlinear systems modeling are subject to treatment by computational models based on neural networks. Two new views were responsible for such leverage, according to the authors:

> A very important feature of these networks is their adaptive nature, where "learning by example" replaces traditional "programming" in solving problems. This feature makes such computational models very appealing in application domains where one has little or incomplete understanding of the problem to be solved but where training data is readily available.
>
> Another key feature is the intrinsic parallelism that allows for fast computations of solutions when these networks are implemented on parallel digital computers or, ultimately, when implemented in customized hardware. (HASSOUN et al., 1995)

The most interesting facet presented was a concept modification on problem solutions. It would no longer be a matter of building algorithms for treating previously identified cases, previously established rules or any other past approach elicited through requirements. For Sommerville (2011), a requirement can be defined as:

> The requirements for a system are the descriptions of what the system should do — the services that it provides and the constraints on its operation. These requirements reflect the needs of customers for a system that serves a certain purpose such as controlling a device, placing an order, or finding information. (SOMMERVILLE, 2011, p.83)

Different levels of abstraction of these needs are possible: from the most elementary level of unitary operations to concepts and definitions in natural language used in the system's application context. Therefore, the author divides requirements into two broad categories: user requirements and system requirements.

> 1. User requirements are statements, in a natural language plus diagrams, of what services the system is expected to provide to system users and the constraints under which it must operate.
>
> 2. System requirements are more detailed descriptions of the software system's functions, services, and operational constraints. The system requirements document (sometimes called a functional specification) should define exactly what is to be implemented. It may be part of the contract between the system buyer and the software developers.(SOMMERVILLE, 2011, p.83)

This definition of requirements does not apply in artificial neural networks implementations. Such a conclusion can be proved comparing the paradigms defined by Hagan, Demuth and Beale (2014) and Hassoun et al. (1995) with those of Sommerville (2011). First ones start from records about what actually happens to discover ways in which results are obtained. Second one begins defining how to do something, to obtain an expected result. Nielsen (2015) explicit such idea:

> Neural networks are one of the most beautiful programming paradigms ever invented. In the conventional approach to programming, we tell the computer what to do, breaking big problems up into many small, precisely defined tasks that the computer can easily perform. By contrast, in a neural network we don't tell the computer how to solve our problem. Instead, it learns from observational data, figuring out its own solution to the problem at hand.(NIELSEN, 2015, p.iii)

A direct reference to Rosenblatt (1961)'s perceptron was made, describing it as a starting point for developing artificial neural networks in modern models. However, points out that literal use of the basis proposed has little applicability. Thus, makes small comments about what could be called as *the basic principle of perceptron functioning*, according to his interpretation:

> A way you can think about the perceptron is that it's a device that makes decisions by weighing up evidence. Let me give an example. It's not a very realistic example, but it's easy to understand, and we'll soon get to more realistic examples. Suppose the weekend is coming up, and you've heard that there's going to be a cheese festival in your city. You like cheese, and are trying to decide whether or not to go to the festival. You might make your decision by weighing up three factors:
>
> 1. Is the weather good?
>
> 2. Does your boyfriend or girlfriend want to accompany you?
>
> 3. Is the festival near public transit? (You don't own a car).
>
> (NIELSEN, 2015, p.3 Tradução livre.)

One way to understand the influence of each of the variables is to think of a perfect case: the day is perfect for a walk, you will have company and there is public transport at the event's entrance door. In theory, the probability that you will attend the event is one hundred percent, or, mathematically speaking, 1. However, it must be considered that ideal situations are rare and certain variables have more influence than others. The author summarizes the functioning of a perceptron in an objective way by means of a mathematical expression that evaluates the value obtained through the sum of the influences of each variable against an activation threshold: an algebraic value meaning that, even if there are adverse conditions, the sum of all of them makes the result viable or unachievable. Laurene Fausett (1994) calls this boundary *activation* or *activity level*, and describes it as the *internal state* of a neuron. A resumed expression of the equation is presented below.

$$result = \begin{cases} 0 & if \quad \sum_j w_j x_j \leq threshold \\ 1 & if \quad \sum_j w_j x_j \geq threshold \end{cases} \tag{4.6}$$

Assume that it is impossible for you to attend the event if it rains and that the activation value for the function *"go to event"* is 0,5. In this case, a set of influence weights (w_1, w_2, w_3), could have the values (0,6; 0,2; 0,2) , that is, any value assigned to w_1 that does not activate it, results in a situation where the outcome is not going to the event.

Considering this scenario, it would be possible to generate several decision-making models, changing weights or function threshold values. An example: by changing the activation value to 0,3 would be possible to activate the "trip to the event" if conditions 2 and 3 are true. In this case, there is a greater probability to attend the event — required conditions level decreases. On the other hand, updating the weights to (0,35, 0,35, 0,3) and keeping the activation limit at 0,5 sets a situation where only a combination of two conditions would activate the function in question — required conditions become more complex.

Obviously, perceptrons do not have the same complexity as the human decision-making system: it is just an extremely reduced expression of this mechanism. Its main attribute is the ability to analyze variable influence in a given situation.

Fausett (1994) describes that an analysis unit analogous to Rosenblatt (1961)'s perceptron would work synchronized with other units: although it is possible to transmit only one signal at a time, each unit does it to several other units, forming a signal analysis network, interconnected and interdependent with each other. Thus, the author defines an artificial neural network as:

> An artificial neural network is an information-processing system that has certain performance characteristics in common with biological neural networks. Artificial neural networks have been developed as generalizations of mathematical models of human cognition or neural biology, based on the assumptions that:
>
> 1. Information processing occurs at many simple elements called neurons;
>
> 2. Signals are passed between neurons over connection links;
>
> 3. Each connection link has an associated weight, which, in a typical neural net, multiplies the signal transmitted;
>
> 4. Each neuron applies an activation function (usually nonlinear)to its net input (sum of weighted input signals) to determine its output signal.
>
> (FAUSETT, 1994, p.3)

Haykin (2009) also addresses the general features of a neuron model. He describes it as an information processing unit fundamental to a neural network, with three basic elements in its conception:

> 1. A set of synapses, or connecting links, each of which is characterized by a weight or strength of its own. Specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}. It is important to make a note of the manner in which the subscripts of the synaptic weight w_{kj} are written.The first subscript (k) refers to the neuron in question, and the second subscript (j) refers to the input end of the synapse to which the weight refers. Unlike the weight of a synapse in the brain, the synaptic weight of an artificial neuron may lie in a range that includes negative as well as positive values.
>
> 2. An adder for summing the input signals, weighted by the respective synaptic strengths of the neuron; the operations described here constitute a linear combiner.
>
> 3. An activation function for limiting the amplitude of the output of a neuron.The activation function is also referred to as a squashing function, in that it squashes (limits) the permissible amplitude range of the output signal to some finite value.
>
> (HAYKIN, 1999, p.24)

Figure 9 visually presents these definitions.

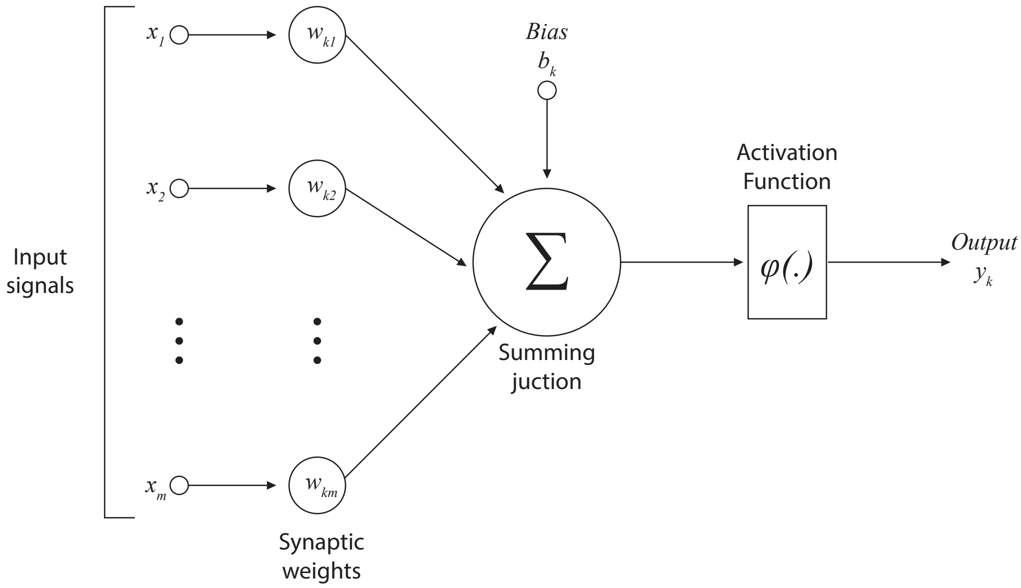


Figure 9 – Haykin (2009)'s neuron graphic model

Source: Adapted from Haykin (2009, p.11)

Simon Haykin (1999) also points out this proximity to the biological model of neural processing. For the author, a neural network can be seen as a machine that was designed as a model of how the brain performs a certain task, including learning processes. In order to obtain greater efficiency, these implementations make use of a massive amount of computational units called "neurons" or "processing units". Quoting the definition:

> A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential

> knowledge and making it available for use. It resembles the brain in two aspects:
>
> 1. Knowledge is acquired by the network form its environment through a learning process;
>
> 2. Interneuron connection strengths, known as synaptic weights are used to store the acquired knowledge
>
> (HAYKIN, 1999, p.24)

For the author, the potential to artificially obtain a measure of intelligence is on the conjunction of all processed signals influence weights performed by every element of the network. That is: an extensive and interconnected set of signal strength values.

For Basheer and Hajmeer (2000), influence weights are the measure of intelligence of an artificial system. Learning consists on the process of changing these value to obtain a more assertive configuration while facing a given set of external stimuli captured in order to undertake a task. This includes several kinds of changes: processing units arrangement, connections between these units and their activation rules.

On similar path, Hassoun et al. (1995) credits significant relevance to the ability of a neural network to learn through interaction with the environment or with a given information set. This is usually achieved through an adaptive process, known as a learning rule or algorithm, whereby network weights are incrementally adjusted to improve a predefined measure of performance over time.

A *learning algorithm* is characterized by the function that makes it possible to modify weights, in order to obtain the desired configuration. In the same sense, the connection mode between each analysis unit (each neuron) also becomes relevant, since it is through these connections between parts that analysis weights are obtained. Two points then become fundamental for developing neural networks: architecture design and method to obtain influence weights values.

4.3.1 Influence Weights and Activation Functions

Obtaining the arrangement of influence weights applied to each signal sent to a processing unit is one of the foundations for the development of a model of rational actions given a set of stimuli. The more assertive the evaluation model is given the objective, based on the set of values obtained, the more "intelligent" would the solution be.

Fausett (1994) define as "training" the method to find these values, characterizing it as the distinction point between different neural networks (FAUSETT, 1994, p.15). In the author's words:

> Many of the tasks that neural nets can be trained to perform fall into the areas of mapping, clustering and constrained optimization. Pattern classification and pattern association may be considered special forms of the more general

> problem of mapping input vectors or patterns to the specified output vectors or patterns.(FAUSETT, 1994, p.15)

Distinguishes then two major categories of training: supervised and unsupervised. Also adds the existence of weights that are not conceived through an iterative training process, having pre-established fixed values. She also mentions some ambiguity in binary classification of training methods into supervised and unsupervised, citing some authors who consider it useful to create a third category, called self-supervised. Proposes that it is possible to carry out, in general, a useful correlation between the training category to be adopted considering the type of problem to be solved, based on some characteristics of each method.

4.3.1.1 Supervised learning

Initial essays on construction of processing units began with classification problems: facing a set of stimuli, it would fit or not in some condition.

For Fausett (1994), such practice is the most typical neural net setting, training is accomplished by presenting a sequence of training vectors, or patterns, each with an associated target output vector (FAUSETT, 1994, p .15,). The author mentions such method as being called *supervised training*. In the author's words:

> Some of the simplest (and historically earliest) neural nets are designed to perform pattern classification, i.e., to classify an input vector as either belonging or not to a given category. In this type of neural net, the output is a bivalent element, say, either 1 (if the input vector belongs to the category) or -1 (if it does not belong).(FAUSETT, 1994, p.15)

Haykin (2009) addresses this practice, calling it *learning with a teacher* or *input-output mapping*. He defines it as a popular learning process that involves modifying the synaptic weights (influence weights) of a neural network by applying a set of properly classified *test examples* or *training examples*, which represent the teacher knowledge. Each example consists of a unique *input signal* and its desired corresponding *output*. As the network is presented a randomly chosen example from the set, the synaptic weights (also called **free parameters**, due to the possibility of being freely changed throughout the learning process) of the network are modified in order to minimize the difference between the desired response and the actual response produced by the input signal according to a specific statistical criterion. (HAYKIN, 2009, p.35)

Also Hassoun et al. (1995) cites supervised learning as a synonym for *learning with a teacher* or *associative learning*, characterizing it as a process where each input signal or pattern received from the environment is associated with a specific target pattern desired.

For Basheer and Hajmeer (2000), there are six relevant general characteristics to be considered when classifying an artificial neural network:

a. The function that the ANN is designed to serve (e.g., pattern association, clustering);

b. the degree (partial / full) of connectivity of the neurons in the network,

c. the direction of flow of information within the network (recurrent and nonrecurrent), with recurrent networks being dynamic systems in which the state at any given time is dependent on previous states;

d. the type of learning algorithm, which represents a set of systematic equations that utilize the outputs obtained from the network along with an arbitrary performance measure to update the internal structure of the ANN;

e. the learning rule (the driving engine of the learning algorithm);

f. the degree of learning supervision needed for ANN training.

Regarding the level of supervision, the authors make a comment on supervised learning:

> Supervised learning involves training of an ANN with the correct answers (i.e. target outputs) being given for every example, and using the deviation (error) of the ANN solution from corresponding target values to determine the required amount by which each weight should be adjusted. (BASHEER; HAJMEER, 2000, p.12)

Engelbrecht (2007) stands similarly to Haykin (2009), citing the need for a test data set, with input vectors associated with target vectors, which are used to measure the assertiveness level of the neural network learning process, as well as a way to guide influence weights adjustments in order to reduce error spread. For the author, supervised learning implementations can be classified in big groups according to how temporal distinctions are treat through learning processes.

> Feedforward NNs such as the standard multilayer NN, functional link NN and product unit NN receive external signals and simply propagate these signals through all the layers to obtain the result (output) of the NN. There are no feedback connections to previous layers. Recurrent NNs, on the other hand, have such feedback connections to model the temporal characteristics of the problem being learned. Time-delay NNs, on the other hand, memorize a window of previously observed patterns.
>
> (ENGELBRECHT, 2007, p.27)

4.3.1.2 Unsupervised learning

Not every human brain rational ability can be learned through predetermined classifications. In this sense, Engelbrecht (2007) refers to an Aristotelian observation that describes human memory ability to connect items (such as objects, feelings and ideas) that are similar or contradictory, that occur in proximity or in succession. This association building technique between various stimuli without guidance (a tutor) is called unsupervised learning. The author defines associative memory neural networks those that implement this characteristic of the human mind.

Russell and Norvig (2010) cites clustering tasks as the most common use of unsupervised learning, describing it as a stimulus group formation activity, exemplifying its application in a situation where an intelligent agent could classify the day's traffic in "heavy transit days" and "light transit days", without the need for a information set previously classified by a tutor.

In the same sense, Hassoun et al. (1995) also characterizes unsupervised learning as a process of grouping (or detection of similarities) of unmarked patterns in a given training set. The idea is to optimize (maximize or minimize) the performance criterion or function defined in terms of output activity performed by units in the network. It is expected that the weights and outputs of the network converge into representations that capture statistical regularities noticed on input data.

A divergent definition was presented by Fausett (1994), explaining the concept of self-organized neural networks, which performs vectors grouping by similarity without using a training set endowed with pre-existing classifications. Only one set of input vectors is provided without any output vector indication. The network, then, modifies its weights so that clusters are created, that are identified later through a representative vector (as a consolidated example of characteristics found on that cluster).

4.3.1.3 Activation functions

Both learning methods are based on influence weights adjustments on each unit of analysis (a neuron) when evaluating an input stimuli to its respective output signal, whether predetermined (in cases of problems that can be applied to supervised learning techniques) or associated by clusters (in typical problems of unsupervised learning).

For Haykin (2009), this influence calculus procedure can be described as a *squashing function*, as it *squashes* (or limits) output range amplitude of each unit of analysis. He calls this reducing function *activation function*, expressing its influence value as a finite value [0,1] or, alternatively, [-1,1].

Engelbrecht (2007) extends the responsibilities of activation functions to network initialization, in addition to signal intensity regulation. For the author, a neural network collects all input signals and computes a *net signal*, as the summation of weights from each signal individually. This *net signal* is used as input to the *activation function*, which calculates the output signal of the neural network. In the same sense, Agatonovic-Kustrin and Beresford (2000) cites being the sum, duly calculated, of the inputs of a neuron.

Hagan, Demuth and Beale (2014) and Basheer and Hajmeer (2000) match the definitions presented, just naming the function *transfer function*, as it is responsible for signal intensity transfer between one neuron and another in the network.

Fausett (1994) describes that each unit of a network, a neuron, has its *internal state*, called *activation* or *activation level*, which can be expressed as the sum of all the inputs it receives, effected through an output function, also called the activation function.

There is a convergence in the authors' understanding in the sense that the adaptability of a neural network depends on the way a activation function acts on its influence weights. Fausett (1994) makes a small comment regarding the number of activation functions applied to a neural network. The author cites that, ordinarily, only one activation function is applied in a network, and this rule is not mandatory. It extends the discussions towards a broad classification of activation functions into linear and nonlinear.

For the author, an example of *linear function* presents the characteristic of *identity function*. Identity functions can be represented by the equation $f(x) = x$ and graphically demonstrated through figure 10. It is verified that for each input value in x, this value is mirrored even in $f(x)$.

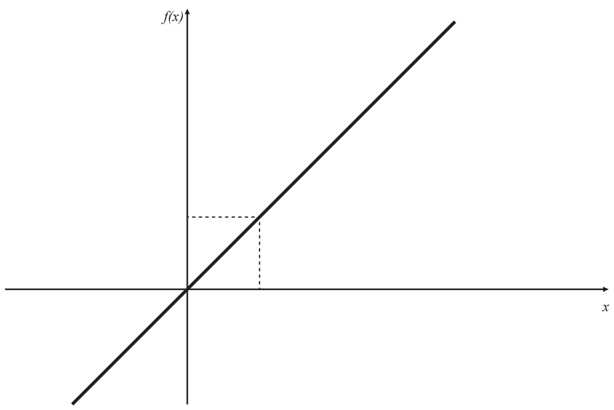


Figure 10 – Graphic representation of an identity function

Source: Adapted from Fausett (1994, p.17)

Step functions, widely used in single-layer networks, can present binary (1 or 0) or bipolar (1 and -1) behavior while propagating signals. Its behavior is related to the concept of limit, a given value to which the input signal x is compared. No output signal is propagated until this value is exceeded. The equation below mathematically demonstrates the behavior of a binary function. Figure 11 presents the graphical representation of the same equation.

$$f(x) = \begin{cases} 1 & se \quad x \geqslant \theta \\ 0 & se \quad x < \theta \end{cases} \tag{4.7}$$

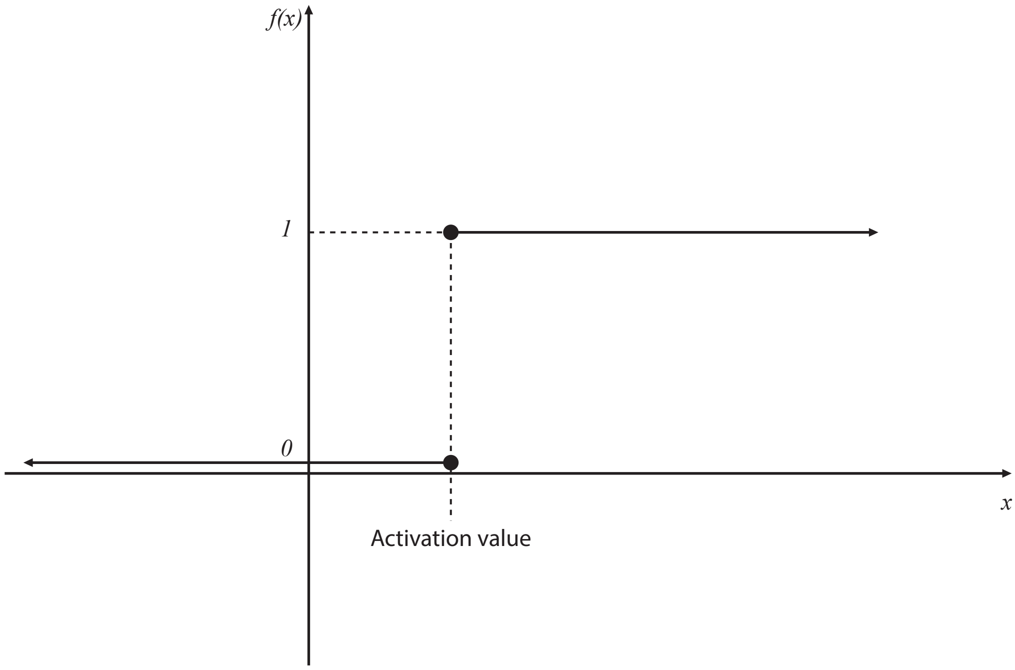


Figure 11 – Graphic representation of a binary step function

Source: Adapted from Fausett (1994, p.17)

Haykin (2009) calls *stair functions* as *boundary functions*, citing also that it is commonly addressed as *Heaviside function*, in reference to the mathematician and engineer *Oliver Heaviside*.

Engelbrecht (2007) describes a conjunction made from the linear and step functions cited by Fausett (1994). It starts by defining how to calculate the signal of a neural network, as being, in general, the sum of all input signals. The equation below mathematically demonstrates the author's definition, where *net* refers to the total input signal of an artificial neural network.

$$net = \sum_{i=1}^{I} z_i v_i \tag{4.8}$$

To demonstrate an artificial neuron functioning while correlating it with total input signal, the author proposes a graphic model reproduced in figure 12, where the set

$$\mathcal{Z} = (z_1, z_2, ..., z_i) \tag{4.9}$$

refers to the input vector formed by the i signals that compose it. For each z_i signal an influence weight v_i is associated, which can enhance or neutralize that signal.

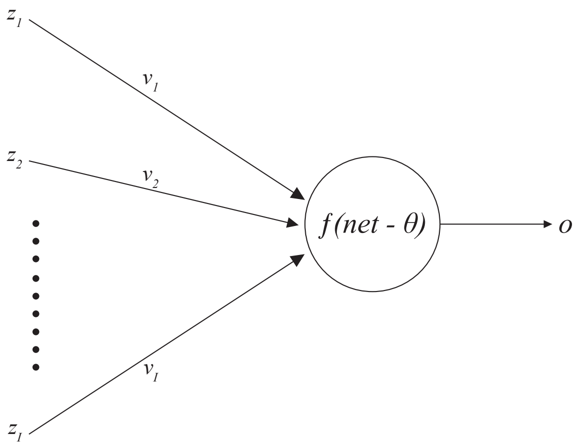


Figure 12 – Engelbrecht (2007) graphic representation of an artificial neuron

Source: Adapted from Engelbrecht (2007, p.17)

The final result of any activation function processing can be demonstrated through the expression ($net - \theta$), where θ is the activation value of the function, that is, the output signal will be the value of the subtraction between the general input signal of the network and the activation threshold. The author adds the concept of slope to the definition of linear function, so that an identity function has a 45-degree slope, which ensures that for each input value on the signal axis, it reflects the same output value for the function activation. The mathematical equation of linear functions can then be written as follows, where λ is the slope of the function:

$$f_{AN}(net - \theta) = \lambda(net - \theta) \tag{4.10}$$

For Engelbrecht (2007) a *ramp function* is described by means of an equation that delimits a space of input values where the behavior of the function resembles a linear function amid a step function. What differs ramp functions from step functions is that the slope between threshold values $(\lambda, -\lambda)$ is an angle less than 90 degrees typically found in step functions. The following equation mathematically demonstrates the ramp function, while figure 13 presents its graphical form, where

$$f_{AN}(net - \theta) = \begin{cases} \lambda & if \quad net - \theta \geqslant \in \\ (net - \theta) & if \quad - \in < net - \theta < \in \\ -\lambda & if \quad net - \theta \leqslant \in \end{cases} \tag{4.11}$$

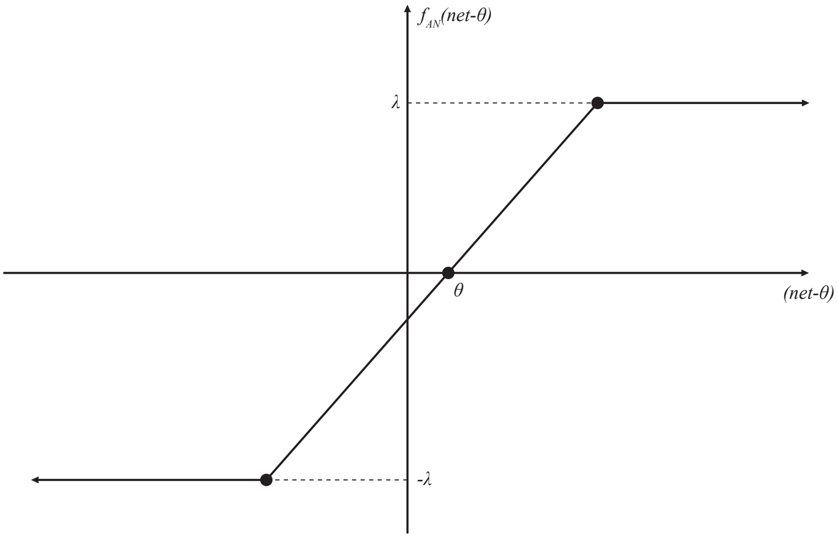


Figure 13 – Representação gráfica de uma função de rampa

Source: Adapted from Engelbrecht (2007, p.19)

Fausett (1994) describes the behavior of *sigmoid functions*, also referred to as “S”-shaped curves, as being less burdensome during the training process of a network using back-propagation techniques. This difference is due to the relationship between the value of the function at a given point and the value of the derivative at the same point.

Another category described by the author is *logistics functions*, also named *binary sigmoids* or *logistics sigmoids*. Its output behavior is similar to *step functions*: the range of values lies between (0, 1). At the other hand,Engelbrecht (2007) refers to *sigmoids functions* as a continuous version of *ramp functions*, where the general signal is comprised between (0, 1), that is, $f_{AN}(net - \theta) \in (0, 1)$. Both authors agree with the existence of an slope variable in *sigmoid functions*, differing only in the symbols used in their equations. The expression below is described by Engelbrecht (2007), while figure 14 presents its graphic representation

$$f_{AN}(net - \theta) = \frac{1}{1 + e^{-\lambda(net-\theta)}} \tag{4.12}$$

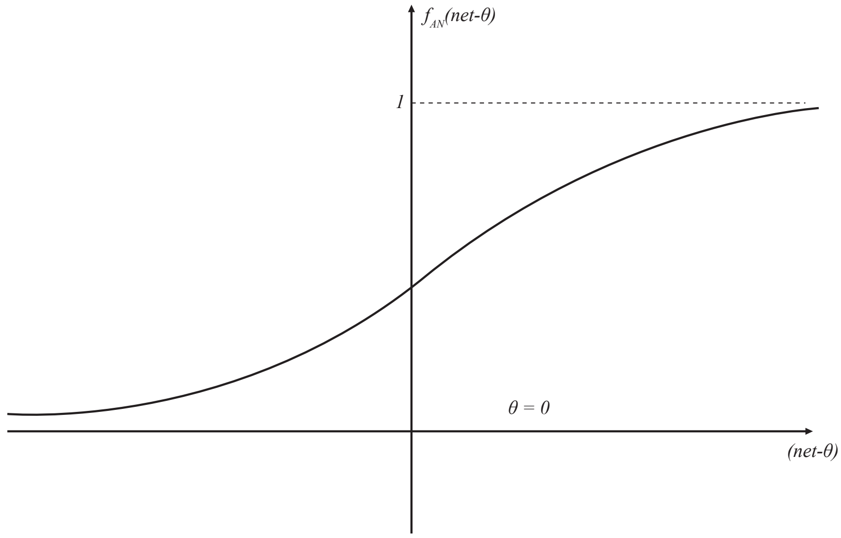


Figure 14 – Sigmoid function graphic representation

Source: Adapted from Engelbrecht (2007, p.19)

Another category described by Fausett (1994) is the *bipolar sigmoid*, which Engelbrecht (2007) calls *hyperbolic tangent*. In the same sense as a *bipolar step function*, its activation spectrum lies between (-1, 1). The mathematical expression described by this author

$$f_{AN}(net - \theta) = \frac{e^{\lambda(net-\theta)} - e^{-\lambda(net-\theta)}}{e^{\lambda(net-\theta)} + e^{-\lambda(net-\theta)}} \tag{4.13}$$

can also be reduced to

$$f_{AN}(net - \theta) = \frac{2}{1 + e^{-\lambda(net-\theta)}} - 1 \tag{4.14}$$

and its graphic representation can be observed on figure 15.

Engelbrecht (2007) describes the *gaussian function*, as being determined by a symmetrical distribution of values in relation to the center of the curve, whose value will always be $(net - \theta)$, and the variable σ is the standard deviation of the Gaussian distribution. Figure 16 graphically demonstrates the function, as the equation below does it mathematically.

$$f_{AN}(net - \theta) = e^{-(net-\theta)^2/\sigma^2} \tag{4.15}$$

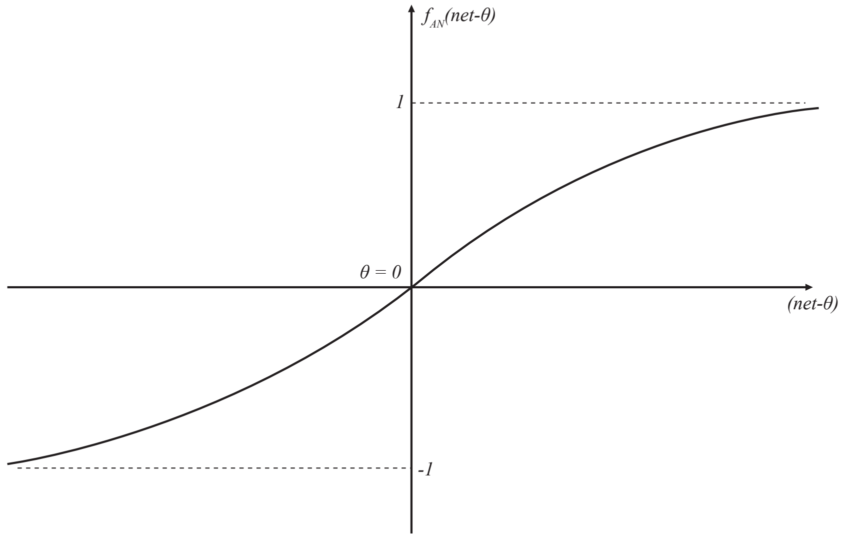


Figure 15 – Hyperbolic function graphic representation

Source: Adapted from Engelbrecht (2007, p.19)

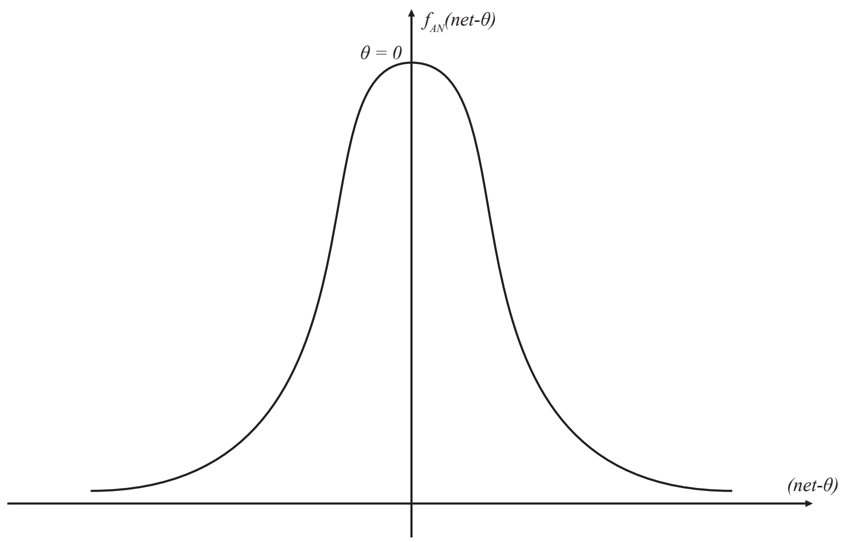


Figure 16 – Representação gráfica de uma função gaussiana

Source: Adapted fromEngelbrecht (2007, p.19)

Engelbrecht (2007) finally performs an analysis on the role of activation functions in what he called *artificial neuron geometry*. Based on a Cartesian plane, where the abscissa axis refers to the neuron output value and the ordinate axis refers to the neuron input signal value, the role of activation functions would be dividing the plane into three distinct spaces: input values where the output will be negative, input values where the output will be null, and input values where the output will be positive. Figure 17 presents such geometric division.

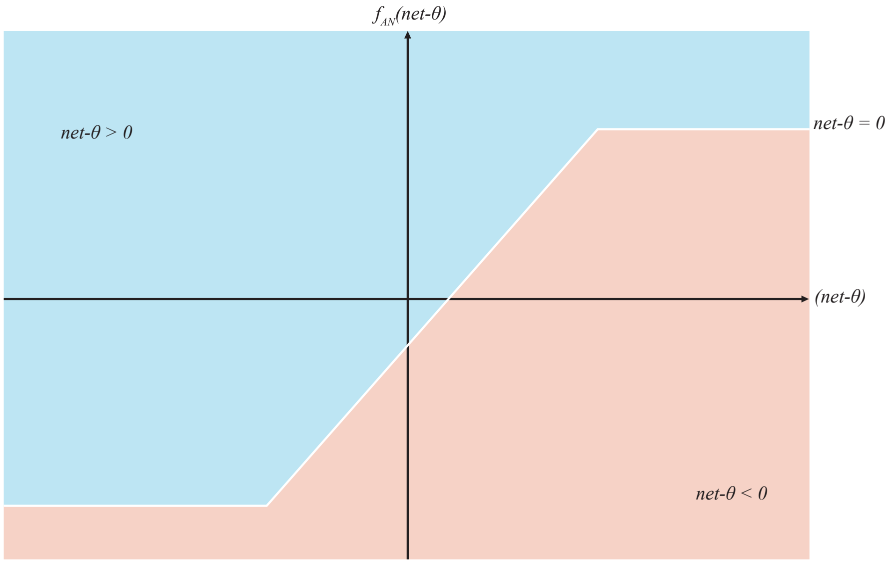


Figure 17 – Activation function geometry representation

Source: Adapted from Engelbrecht (2007, p.21)

The author also stands in the same sense as McClelland et al. (1986) when treating complex problems: a single analysis unit would not be able to draw a mathematically calculable line through an equation in order to separate all negative and positive values in different spaces. Through an *XOR* function, single neuron accuracy would be 75% at the most. To enable linearly non-separable functions analysis, a larger number of neurons is needed.

4.3.2 Neural Networks Architectures

Boosting individual neurons analysis abilities presents itself as the way to obtain better results. Even though several input signals are presented to a single unit of analysis, model assertiveness may be unsatisfactory (HAGAN; DEMUTH; BEALE, 2014, chapt.2 p.9). In this sense, an ordered set of neurons must be arranged so that their analysis capabilities can be leveraged together.

Fausett (1994) defines a neural network architecture as a the orderly arrangement of its units in analysis layers, the connections between the units of each analysis layer, as well as the connections between the analysis layers. In a similar sense, Haykin (2009) characterizes the term as the structure of connections of neurons that compose it.

4.3.2.1 Neural networks classifications

The most common way of classifying neural network architectures considers the number of layers and the signal propagation flow direction. (FAUSETT, 1994; ENGELBRECHT, 2007; HAYKIN, 2009). As for the number of layers, they are classified into singlelayer networks and multilayer networks. As for the propagation of signals along the network, we can classify them as feed forward propagation networks and recurrent propagation networks.

Single-layer networks

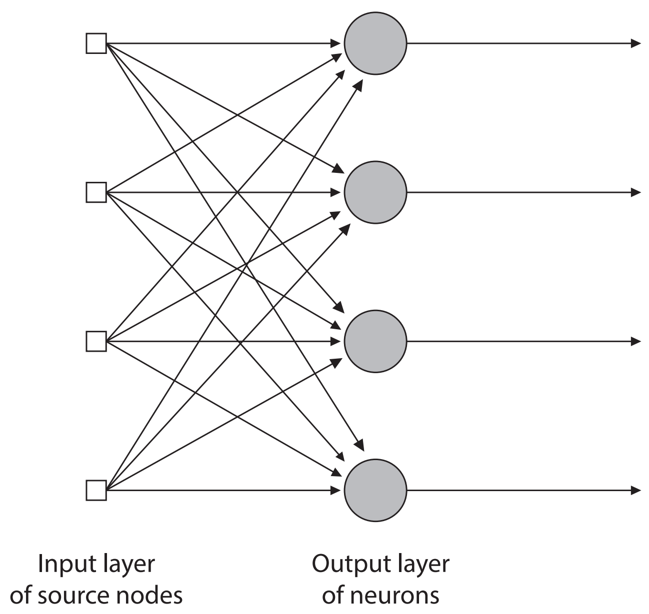


Figure 18 – Single-layer graphic representation

Source: Adapted from Haykin (2009, p.21)

The simplest form of analysis units arrangement is through an input layer of source nodes that projects directly onto an output layer of neurons (computation nodes), but not vice versa.(HAYKIN, 2009, p. 21). Although the existence of the first layer, that of stimuli, is argued, it does not perform any computation, resulting in its non-counting.

Analogous definition presents Fausett (1994), when describing a single-layer network as an array that has only one layer of connection weights. Input units are connected to output units and no other connections are presented in this configuration. Figure 18 graphically illustrates the configuration of a singlelayer network.

Multilayer networks

Unlike singlelayer networks, multilayer networks do not directly connect input units to output units. There is a composition of one or more analysis layers, called *"hidden layers"*. Such hidden layers processes signals coming from other layers, that is, they refine results looking for hidden properties of the input signals. In Haykin (2009) words:

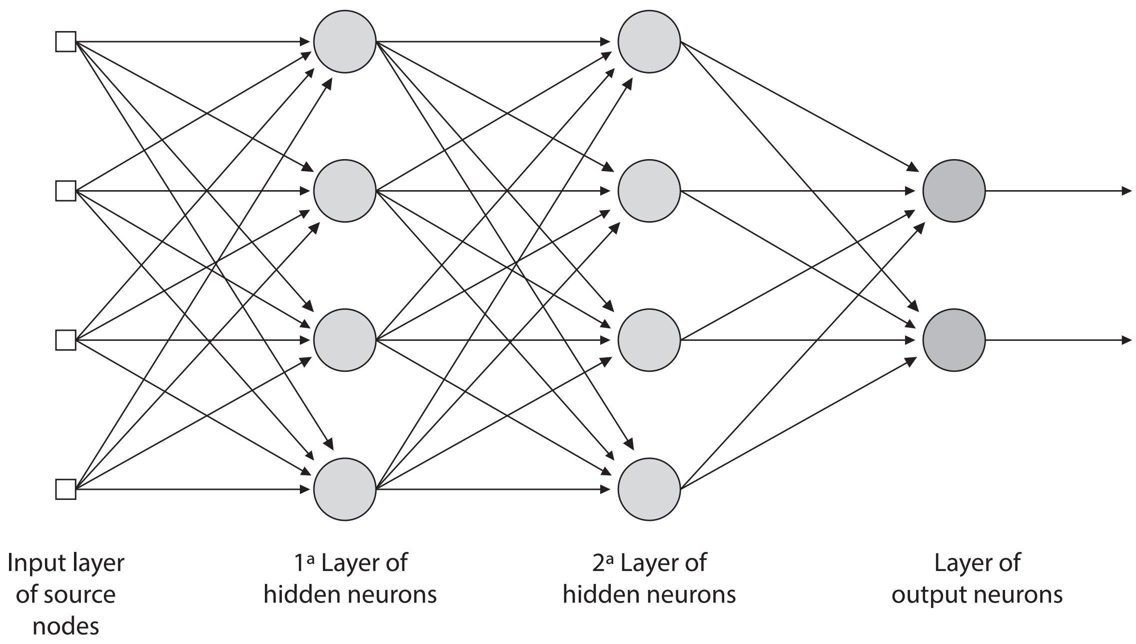


Figure 19 – Multilayer network representation

Source: Adapted from Haykin (2009, p.22)

> The hidden neurons act as feature detectors; as such, they play a critical role in the operation of a multilayer perceptron.As the learning process progresses across the multilayer perceptron, the hidden neurons begin to gradually "discover" the salient features that characterize the training data.(HAYKIN, 2009, p.126)

In this sense, Laurene Fausett (1994) cites that the presence of these hidden layers enable resolution capabilities to more complex problems, which single-layer networks are not able to solve, however, training multilayer networks can present greater difficulty.

Hagan, Demuth and Beale (2014) characterizes hidden layers as any layer other than the one that produces the neural network output. Additionally, it is noteworthy that multilayer networks make possible the use of different activation functions in each of the layers, giving greater flexibility to the performed analyses.

Haykin (2009) defines hidden layers as a matter of visibility of computations performed according to the environment in which the neural network operates. Any layers that do not interact directly with the environment are considered hidden. This feature can provide more flexibility, depending on the type of arrangement to be used. In the author's words, in a *Boltzmann Machine*[1]:

> During the training phase of the network, the visible neurons are all clamped

[1] According Memisevic and Hinton (2010), a Restricted Boltzmann Machine is a simple learning module in which a layer of *visible* units, representing the data observed is connected to a layer of *hidden* units that learn to extract properties from this data.

> onto specific states determined by the environment. The hidden neurons, on the other hand, always operate freely; they are used to explain underlying constraints contained in the environmental input vectors. (HAYKIN, 2009, p.598)

Figure 19 presents a graphical model of a multilayer network with two hidden layers.

Feedforward networks

According to Haykin (2009), Feedforward networks are those whose input signals pass through the network's neurons in a one-way direction, that is, the outputs of each layer of neurons only feed the layers in front of it. The author divides this category into singlelayer or multilayer feedforward networks, concatenating the two definitions mentioned above.

Another characteristic mentioned by the author refers to the connections between the neurons of the analysis layers, which can be classified as fully or partially connected. Quoting:

> The neural network in Fig. 19 is said to be fully connected because every node in each layer of the network is connected to every other node in the adjacent forward layer. If, however, some of the communication links (synaptic connections) are missing from the network, we say that the network is partially connected.(HAYKIN, 2009, p.23)

The author also mentions the concept of backpropagation, as a popular method of training multilayer networks, which basically consists of two phases:

> In the forward phase, the synaptic weights of the network are fixed and the input signal is propagated through the network, layer by layer, until it reaches the output. Thus, in this phase, changes are confined to the activation potentials and outputs of the neurons in the network.(HAYKIN, 2009, p.123.)
>
> In the backward phase, an error signal is produced by comparing the output of the network with a desired response.The resulting error signal is propagated through the network, again layer by layer, but this time the propagation is performed in the backward direction. In this second phase, successive adjustments are made to the synaptic weights of the network.(HAYKIN, 2009, p.124.)

In this sense, we can identify two types of signals propagated through a neural network: function signals and error signals. Figure 20 presents both concepts graphically.

Function Signals are input signals (stimuli) that begin their path through the network on the input stimulus layer, being forward propagated layer after layer, all the way to the output layer. According to Haykin (2009), the name "function signals" comes from: *a)* playing a useful role in obtaining the output signal of the network and; *b)* at each neuron of the network through which a function signal passes, the signal is calculated as a function of the inputs and associated weights applied to that neuron(HAYKIN, 2009, p.125)

Error signals An error signal originates at an output neuron of the network and propagates backward (layer by layer) through the network, that is, they start the path in the output layer. They are referred to as “error signals” because its computation by every neuron of the network involves an error-dependent function in one form or another.(HAYKIN, 2009, p.125)

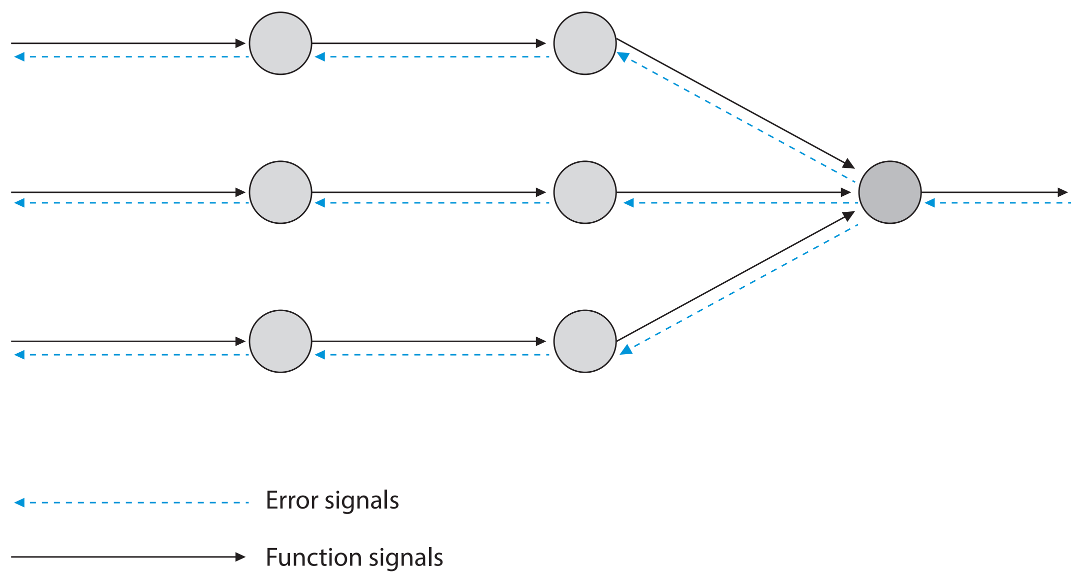


Figure 20 – Signal flow representation for a backpropagated network

Source: Adapted from Haykin (2009, p.125)

Recurrent Networks

One of the issues raised when dealing with multilayer networks is treating temporal variations of the different moments which the understanding of a given problem changes throughout each step of the analysis. In a synthetic way, by reasoning that each layer presents results on the understanding of the problem, a result obtained in a further layer of the neural network could, in theory, influence the results of previous one.

Engelbrecht (2007) describes simple recurrent networks through the presence of feedback connections that add the ability to learn the temporal characteristics of the data set.

Hagan, Demuth and Beale (2014) complements citing that this feedback is a connection between analysis units, where an output signal can become an input signal, in the opposite direction to the initial insertion, that is, different from backpropagation, in which the direction is unique until the last layer of the network is reached. To demonstrate this difference, the author introduces the concept of delay units.

A **delay unit** is an analysis unit where the output $\mathbf{y}(t)$ is computed through an input $\mathbf{x}(t)$, so that $\mathbf{y}(t) = \mathbf{x}(t - 1)$, acting like a constraint on the output to be initialized at time $t = 0$ where the sequencing of these moments is characterized by their discrete algebraic separation, that is, non-continuous, finite and distinguished by means of integer values. On neural networks such attribute materializes in each layer or analysis unit of the network itself. Figure 21 graphically presents the operation of a delay unit.

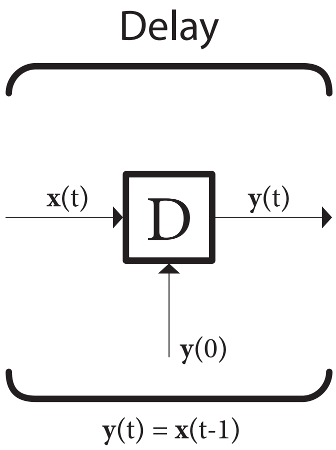


Figure 21 – Delay unit representation

Source: Adapted from Hagan, Demuth and Beale (2014, p.2.13)

The representation of a recurrent network, in a simplified way, can be demonstrated through figure 22, where (**W**) represents an analysis unit matrix; (**b**) the calculated error accumulated in previous computations; (**f**) an activation function; (**D**) the network delay blocks; (**x**) the network input signals; (**y**) the output signals and (**t**) the iterative discrete moment at each round of signal computation.

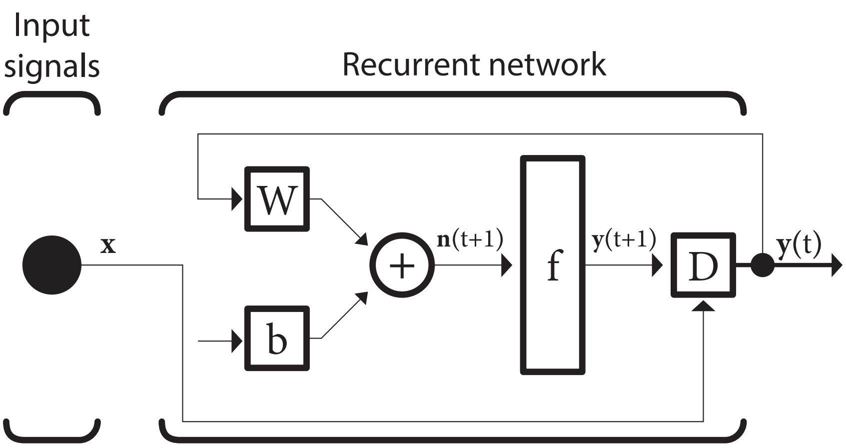


Figure 22 – Recurrent network simplified representation

Source: Adapted from Hagan, Demuth and Beale (2014, p.2.14)

Recurrent Networks are better described on section 4.4.2.2.

4.3.3 Guidelines for architectural choices in neural networks

As characterizing an artificial neural network as a set of analysis units (neurons), its arrangement describes the defined architectural design, the way in which they are connected and carry out their functions. However, signal propagation is also taken into account in order to obtain greater accuracy, therefore architectural definition alone does not tend to guarantee better results. It seems then plausible to assume the existence of techniques that can optimize assertiveness gain.

Haykin (2009) addresses the issue through a concept called *Credit-Assignment Problem*, which is summarized by *giving credit* or *attributing blame* to each of the internal decisions made by the hidden units, in the overall result obtained by the network. This phenomenon occurs due to error correction directive in multilayer networks: in order to provide a solution to a given task, it is necessary to determine different behavior patterns for each unit through specifications dictated by the error-correction algorithm. The author quotes that the associated error of an output neuron can be seen, but how to visualize it on the hidden neurons? In this matter, the concept of backpropagation plays a fundamental role.

For Engelbrecht (2007), the problem listed by Haykin (2009) refers to the *Supervised Learning Problem*. The author considers the following hypothetical situation:

(a) A finite set of input-output pairs $\mathfrak{D} = \{d_p = (z_p, t_p) | p = 1, ..., P\}$, where z_p is the input value for the intended result t_p;

(b) For each analysis of a given input z_p, an output o_p is calculated through an unknown function $\mu(z)$;

(c) The relationship between the intended results and the function $\mu(z)$ can be described by the expression $t_p = \mu(z_p) + \zeta_p$, where ζ_p are independent noises distributed in a identical way, with an overall average value of zero.

The neural network goal is to determine a function $\mu(z)$ that approximates outputs o_p to results t_p. To achieve this objective, some type of training method has to be considered. Following this premise, Engelbrecht (2007) cites the division of the finite set $\mathfrak{D}$ into three subsets, formed by the random division of its items:

(a) $\mathfrak{D}_T$ as a training subset, which performs the approximation of the function $\mu(z)$;

(b) $\mathfrak{D}_V$ as a validation subset, which calibrates the generalization of the network, that is, how general and comprehensive the network behaves as different input signals are presented;

(c) $\mathfrak{D}_G$ as a test subset, which calibrates the accuracy of the network's generalization, that is, how assertive the network is when its generalization increases;

In each function performed by subsets $\mathfrak{D}_T$, $\mathfrak{D}_V$ and $\mathfrak{D}_G$, the author cites the development of several algorithms aimed to improve the optimization of the training step, dividing them into two categories, which can be combined into hybrid optimization methods:

- **Local Optimization**, where the algorithm may get stuck in a local optimum without finding a global optimum. Gradient descent and scaled conjugate gradient are examples of local optimizers;
- **Global Optimization**, where the algorithm searches for the global optimum by employing mechanisms to search larger parts of the search space. Global optimizers include LeapFrog, simulated annealing, evolutionary algorithms and swarm optimization.

Another point addressed by Engelbrecht (2007) and Haykin (2009), is the classification of methods for adjusting influence weights according to the moment in which they are updated. Two categories are proposed by the authors:

- **Stochastic learning, or *online***, where the influence weights are adjusted after processing each input signal. In this case, the selection of the next signal to be analyzed must be random, in order to avoid error incidence due to how input signals are ordered within the training subset $\mathfrak{D}_T$;
- **Batch learning, or *offline***, where influence weights adjustments are accumulated and applied only at the end of processing all input signals of the $\mathfrak{D}_T$ subset of training, which constitute a training round.

Haykin (2009) additionally describes positive and negative points in each strategy. His conclusions are taken through the construction of a concept called *error energy*, according to the specifications below:

(a) Consider a multilayer neural network with training subset expressed by

$$\mathscr{T} = \{\mathbf{x}(n), \mathbf{d}(n\ )\}_{n=1}^{N} \tag{4.16}$$

where the pair $\mathbf{x}(n), \mathbf{d}(n)$ represent, respectively, the input signal and the expected result computed by the neural network;

(b) Defining $y_j(n)$ as the output signal produced by the neuron j in the output layer after analyzing the input signal $\mathbf{x}(n)$, the corresponding *error signal* is expressed by

$$e_j(n) = d_j(n) - y_j(n) \tag{4.17}$$

(c) Applying the definition of *LMS* quoted by McClelland et al. (1986), it is possible to obtain the *instantaneous error energy* of the neuron j through the expression

$$\mathscr{E}_j(n) = \frac{1}{2}e_j^2(n) \tag{4.18}$$

(d) The sum of all *error energy* of all the neurons in the set C when processing an input signal $\mathbf{x}(n)$, we get the *total instantaneous error energy*:

$$\begin{aligned} \mathscr{E}(n) &= \sum_{j \in C} \mathscr{E}_j(n) \\ &= \frac{1}{2} \sum_{j \in C} e_j^2(n) \end{aligned} \tag{4.19}$$

(e) Consequently, the *error energy averaged* (also named *empirical risk*) can be described as the sum of all *total instantaneous error energy* obtained through the analysis of the N input signals of the training subset $\mathscr{T}$:

$$\begin{aligned} \mathscr{E}_{av}(N) &= \frac{1}{N} \sum_{n=1}^{N} \mathscr{E}(n) \\ &= \frac{1}{2N} \sum_{n=1}^{N} \sum_{j \in C} e_j^2(n) \end{aligned} \tag{4.20}$$

For the author, **batch learning** synaptic weights adjustments are performed *round after round* of training. Thus, the learning curve will be obtained by comparing $\mathscr{E}_{av}(N)$ and the number of rounds to be executed, regarding the need for, at each round, subset $\mathscr{T}$ suffers a random rearrangement. This process is executed by obtaining the average of several training rounds, which has the following advantages:

- *Accurate estimate* of the gradient vector (the derivative of the cost function $\mathscr{E}_{av}(N)$ with respect to the weight $\mathbf{w}$), thus ensuring, under simple conditions, the convergence of the descending gradient method (Figure 7) to a local minimum;

- *Parallelization* of the learning process.

On the other hand, in a practical way, it demands greater storage capacity to accumulate information along the N input signals to be processed.

As for **stochastic learning**, Haykin (2009) says that synaptic weights update is performed after each input signal in the subset $\mathscr{T}$ is processed. In this sense, the cost function to be minimized is the instantaneous error represented by $\mathscr{E}(n)$. The random arrangement of input signals gives the stochastic (non-deterministic) aspect of the process, which gives this method some advantages:

- Reduction of local minimum trapping probability;

- Reduction of required storage space;

- Less impact of redundancies in the training subset $\mathscr{T}$, given the characteristic of constant updating of influence weights at each input signal analysis;

- Ability to track small changes in the training data subset $\mathscr{T}$, particularly when the environment responsible for generating the data is non-stationary, that is, with unpredictable behavior.

In summary, although stochastic learning has disadvantages, the author cites that it is widely used to solve pattern classification problems for two practical reasons: it is simple to implement and effective on large scale pattern classification problems with increased difficult. Thus, the author proposes a series of methods that are capable of improving the performance of the backpropagation algorithm.

1. **Maximizing information content:** quoting LeCun (1993), every training example presented to the backpropagation algorithm should be chosen on the basis that its information content is the largest possible for the task at hand. Two ways of realizing this choice are as follows:

    - Use an example that results in the largest training error;
    - Use an example that is radically different from all those previously used.

2. **Activation function:** Haykin (2009) stands by the use of sigmoid functions, given an apparent improvement in learning speed while using it. For this conclusion, he cites studies conducted by Ian LeCun (1993) and presented at the 7th Conference on Neural Information Processing Systems, which indicates the use of symmetric sigmoid functions, particularly the *hyperbolic tangent*, represented by the formula

$$\varphi(v) = a \tanh(bv) \tag{4.21}$$

    where a e b were adjusted with the following values:

    $a = 1.7159$

    $b = \frac{2}{3}$

    It is also presented a graphical representation of the hyperbolic tangent function reproduced in figure 23, through which the following useful properties can be observed, enabling relative controlled maintenance of deviations from constant target values in the range (-1, 1):

    - $\varphi(1) = 1$ e $\varphi(-1) = -1$;
    - At its origin, the slope of the curve (or effective gain) of the activation function is close to one unit:

$$\begin{aligned} \varphi(0) &= ab \\ &= 1.7159\left(\frac{2}{3}\right) \\ &= 1.1424 \end{aligned} \tag{4.22}$$

- The second derivative of function $\varphi(v)$, that is, the rate at which the rate of change of function $\varphi(0)$ changes, reaches its maximum value when v =1.

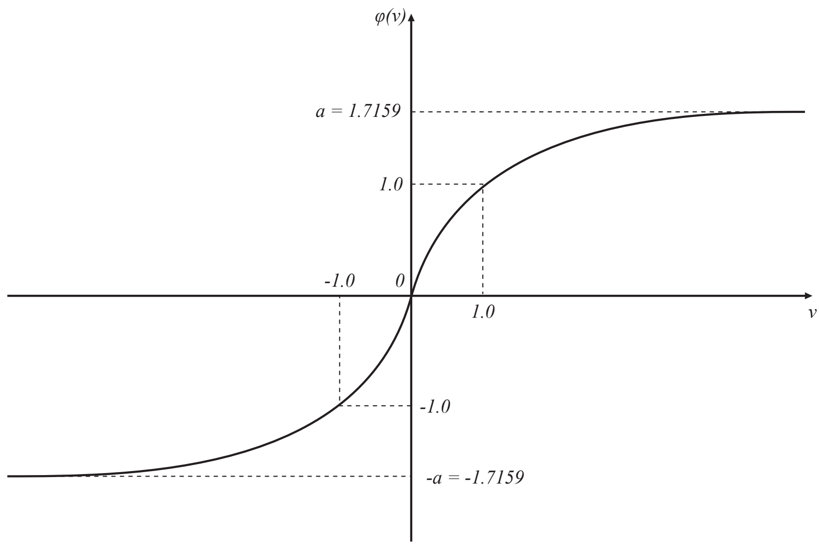


Figure 23 – Hyperbolic tangent function graphic $\varphi(v) = a \tanh(bv)$ for $a = 1.7159$ e $b = \frac{2}{3}$.

Source: Adapted from Haykin (2009, p.146)

3. **Target values:** related to the activation function, it is important that the expected value d_j of the input-result pairs (i_j, d_j) be within the scope of the activating sigmoid function. It is recommended that the target values be compensated by a factor $\mathcal{E}$ that distances them from the lower and upper limits of the sigmoid function, otherwise the backpropagation algorithm tends to take the synaptic weights to infinity, saturating the network neurons, impacting the learning speed. In the case illustrated by figure 23, considering the limit values $\pm a$, we could propose

$$\begin{aligned} d_j &= a - \mathcal{E} \\ &= -a + \mathcal{E} \end{aligned} \tag{4.23}$$

where $\mathcal{E}$ is defined as a positive constant. In the present case, for $a = \pm 1.7159$, the conveniently chosen value of $\mathcal{E} = 0.7159$ would keep the target values for d_j within the range ± 1.

4. **Normalizing the inputs:** each input signal must be *pre-processed* so that its mean value, the averaged over the entire training set approaches to zero, avoiding that the input signals culminate in predominantly positive or negative expected results. In a practical way, it would be like presenting the network only situations where the expected result is true, which would delay the learning of what is false. Figure 24 presents a scenario where input-output pairs have a high tendency to positive results. Three normalization operations are graphically presented: mean removal, decorrelation and covariance equalization.

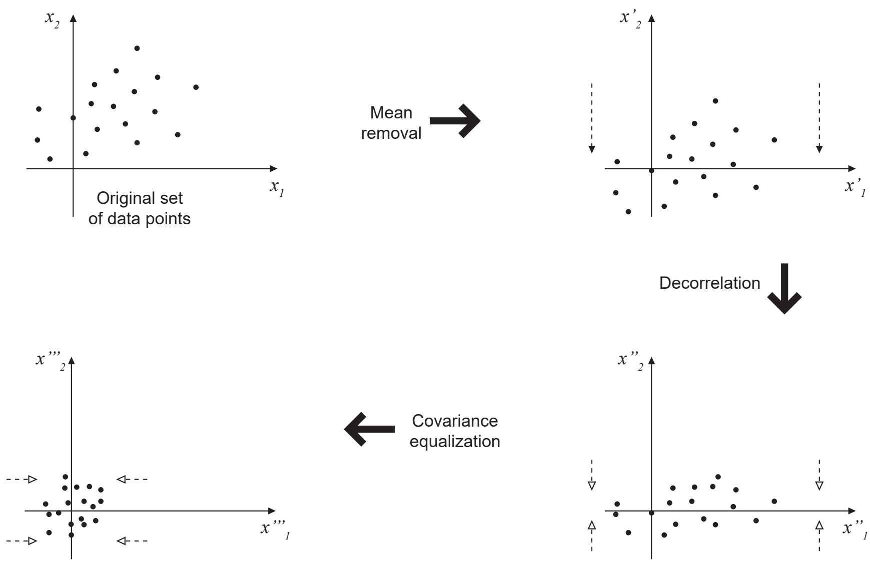


Figure 24 – Normalizing steps

Source: Adapted from Haykin (2009, p.147)

5. **Initialization:** the initial values of synaptic weights has great influence on network learning. For Haykin (2009), extremely high or extremely low initial values should be avoided as they tend to slow down the learning process. Figure 25 presents the graphical representation of a hyperbolic tangent function with markings of high extreme points [Q, R, S, T] and low-end point [P].

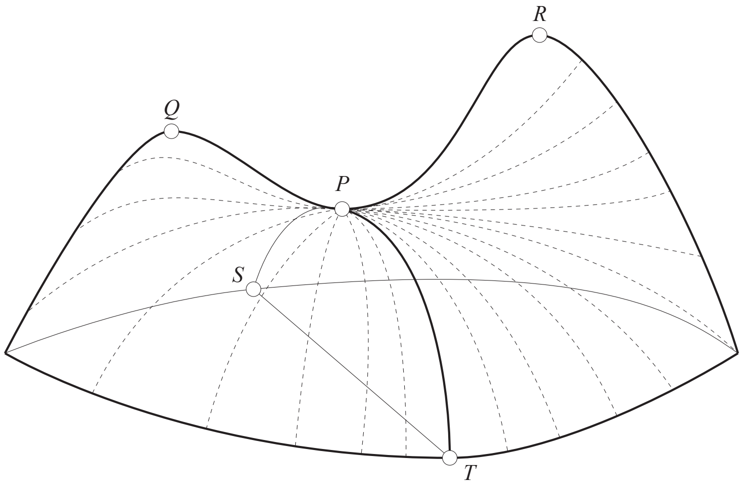


Figure 25 – Three dimensional hyperbolic tangent function

Source: Produced by the author

> Situations [Q, R, S, T] are considered at high synaptic weight point since values only tend to rise fast (in S and T) or descend fast (in Q and R). This situation will lead to a high saturation of neurons (since the high synaptic value will excite the neurons as a whole) which will slow down the learning process.
>
> Likewise, if an extremely low value is assigned (in P), the activation function's area of action will be predominantly flat as in a *saddle point*, which culminates in a low activation of neurons, also slowing down the learning process. Ideally, the initial value of the synaptic weights should fall between these two extremes.

There are some issues to be addressed in order to obtain more expressive results with artificial neural networks.

Larger the set available for maximizing the amount of available information, greater the probability that the number of input signals is equally (or exponentially) larger. Bellman (1954) identified this question in his technical report on the *Theory of Dynamic Programming*, which addresses mathematical problems endowed with multiple decision scenarios. In his own words:

> We have a physical system whose state at any time t is determined by a set of quantities which we call state parameters, or state variables. At certain times, which may be prescribed in advance, or which may be determines by the process itself, we are called upon to make decisions which will affect the state of the system. These decisions are equivalent to transformations of the state variables, the choice of a decision being identical with the choice of a transformation. The outcome of the preceding decisions is to be used to guide the choice of future ones, with the purpose of the whole process that of maximizing some function of the parameters describing the final state.
>
> Examples of processes fitting this loose description are furnished by virtually every phase of modern life, from the planning of industrial production

> lines to the scheduling of patients at a medical clinic; from the determination of long-term investment programs for universities to the determination of a replacement policy for machinery in factories; from the programming of training policies for skilled and unskilled labor to the choice of optimal purchasing and inventory policies for department stores and military establishments.(BELLMAN, 1954, p.1)

policies are a sequence of decisions or transformations. The most advantageous *policy* under some predetermined criterion is called *optimal policy*. The greater the number of possible *policies* (i.e., the more complex the scenario presented), the greater the complexity of finding the *optimal policy*. Difficulty lies in the fact that even though dimensionality growth presents linear aspects, learning tends to an exponential rate(AREL; ROSE; KARNOWSKI, 2010, p.13.). This phenomenon is referred as the ***Data Dimensionality Problem***, citing Arel, Rose and Karnowski (2010) stating that the dominant approach has been pre-processing data in order to reduce its dimensionality and enable effective processing, for example, through a classification mechanism. This procedure can be described as a *feature extraction*, different from the item 4. described by Haykin (2009), where there is a simple normalization of the expected results, without verifying the dimensionality data input.

Duda, Hart and Stork (2006) define *feature extraction* as the basic pre-processing step of *pattern classification*. They synthesize the concept as being a procedure for obtaining attributes that identify a certain pattern, with the amount mapped, in most cases, smaller than the totality of attributes necessary to describe the object as a whole, but culminating in information loss(DUDA; HART; STORK, 2006, p.11.). Also differentiate *pattern classification* from *associative memory*, in a hypothetical case of image recognition:

> In acts of associative memory, the system takes in a pattern and emits another pattern associative which is representative of a general group of patterns. It thus reduces the information memory somewhat, but rarely to the extent that pattern classification does. In short, because of the crucial role of a decision in pattern recognition information, it is fundamentally an information reduction process. The classification step represents an even more radical loss of information, reducing the original several thousand bits representing all the color of each of several thousand pixels down to just a few bits representing the chosen category(DUDA; HART; STORK, 2006, p. 11. Tradução livre)

The objective would be to select training examples with the most amount of key attributes of the problem as possible, in order to be properly mapped according to the activation parameters described in the item 3., which would culminate in obtaining a more assertive representation model of the object. The propositions are based on cyclical analysis: searching for more assertive representations requires greater range of instances of the object to be analyzed for extraction of attributes that cause data dimensionality to grow and, therefore, to be treated. Duda, Hart and Stork (2006) list what they call pattern classification sub-problems. The most relevant for this thesis are:

a. **Feature extraction:** it is possible to draw an ambiguous relationship between a pattern classifier and a property extractor. An excellent property extractor would make pattern classification tasks somewhat trivial and, conversely, an efficient pattern classifier would not need a property extractor. Authors say that this is a practical distinction: pattern extraction activity is highly dependent on the correct definition of the domain in question and the problem under analysis, which leads to the need for greater knowledge of the context in which they are inserted.

b. **Noise:** can be treated as noise any property of a perceived pattern that does not originate from the model in question, but arises from some fortuity of the context in which the problem is inserted, as well as from the receivers that apprehend the perceived signal. An important issue to be considered is the case that signal variations are not noise in itself, but rather an unknown characteristic of the object.

c. **Overfitting:** attempting to design a model that achieves close to perfection classification during the training phase of a neural network can lead to a phenomenon of little generalization capabilities, also called overfitting. The situation describes a highly complex algorithm capable of correctly identifying almost all the cases in the training set, as shown in figure 26. Formalizing an algorithm that describes the dashed line that separates the categories *a* and *b* seems highly costfull.

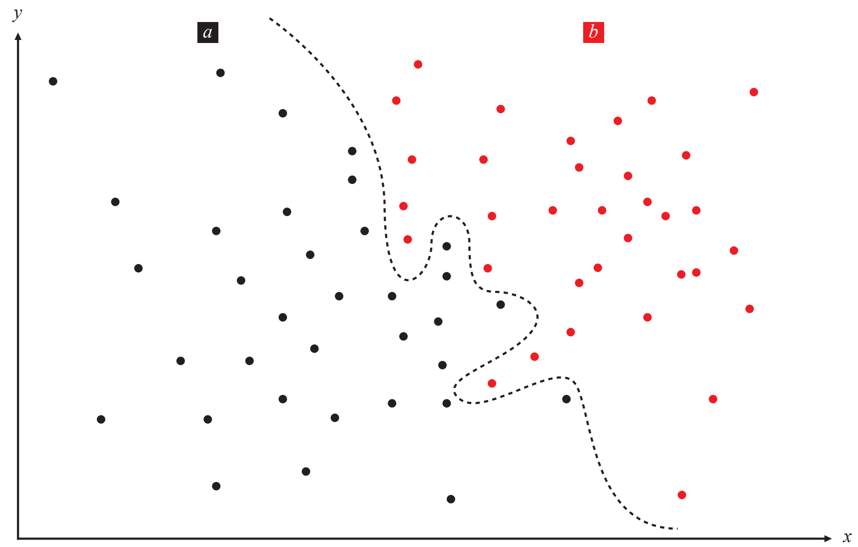


Figure 26 – Overfitting example

Source: Produced by the author

On the other hand, if the algorithm is presented with a new set of signals b_1, which belong to the same *b* pattern, as illustrated in figure 27, the accuracy drops significantly: the classifier function is over-adjusted to the training set and does not generalize well.

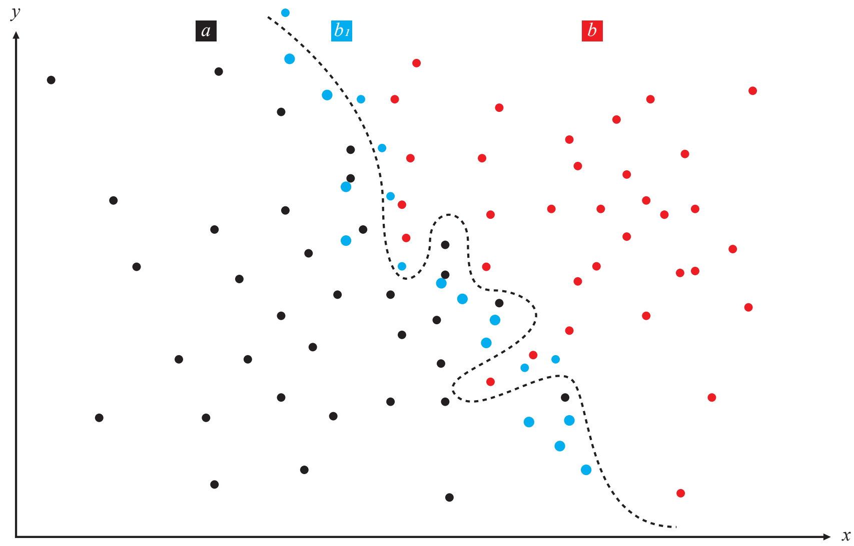


Figure 27 – Overfit function when faced with a new data set

Source: Produced by the author

d. **Prior Knowledge:** sometimes, obtaining better classification methods present the need for objective knowledge of the physical problem in question and the specific attributes of its patterns, for example, when identifying faces, there are two subpatterns for eyes and one subpattern for mouth.

e. **Missing features:** it is necessary to consider the possibility of presenting a set of input signals which do not have a certain attribute analyzed by the network. For example, in a facial identification net, part of a person's face is covered in the image under analysis.

f. **Mereology:** described as the study of mathematical relationships between parts and whole in gathering sets, subsets and supersets. In this sense, there is a need to verify how the correct groups are formed. As an example, in the set **INFORMATION** one could obtain the subsets **IN**, **FORM** or **FORMATION**. It is necessary to obtain an accurate method for grouping elements, suitable for the problem in question.

g. **Segmentation:** intimately related to Mereology. It focuses on the correct identification of the elements of the set, delimiting the end of one instance and the beginning of the next. In cases of cursive handwriting recognition, a neural network needs to find how to identify each letter within the words.

h. **Context:** described as input-dependent information, other then the intended pattern itself. More clearly, is the underlying semantic information that can be verified when certain patterns are present. The element "*persistence*" would be taken as meaningless, unless the context portrays the description of a database transaction, where it could be corrected to "*data persistence*".

i. **Invariances:** pattern recognition must seek a representation model that is invariant to the way attributes are presented. In image recognition cases, transformations such as translation, scaling, orientation or shearing should not interfere with object recognition. For speech recognition, the rhythm of sound should not interfere, as well as the tone of voice (if deeper or higher).

j. **Evidence pooling:** generally, multiple attributes are considered in pattern recognition. The ideal analysis situation is when all are presented in a certain instance. However, if only part of them are recognized, the neural network should design a high-order classifier, which combines all evidences and makes the most assertive decision. Another point to be addressed is if a minority of the analysis neurons indicate the correct classification — the high-order classifier should ignore statistical results and opt for the less appointed classification.

k. **Costs and risks:** generally speaking, pattern classifier purpose can be resumed as recommending actions to be taken, with each action having its cost and an associated risk. In a simplistic way, the associated risk is wrongly classifying an instance, and the associated cost can be described as the sum of the efforts made to design the classifier (computational design time, data computing time, data collecting).

l. **Computational complexity:** complex algorithms, with *brute-force* tendency (mapping and computing all possible combinations for a given problem) tend to be highly costly and sometimes computationally impractical. For illustrative purposes, let's take as an example the storage and computational time needed to map all possible 10^{120} patterns for character recognition presented in 20x20 binary *pixels* images. In general, computational complexity increases as a function of the number of dimensions, attributes and categories analyzed. In this sense, it is necessary to consider an ideal measure of balance between complexity and classifier performance.

4.4 Deep Learning: concepts and development

Since Frank Rosenblatt's *Perceptron* (ROSENBLATT, 1957; ROSENBLATT, 1961) has been conceived, several applications on neural network architectures have been produced. However, attempts to train these *Deep Architectures*, were mostly [2] frustrated until 2006, most notably after Geoffrey Hinton, Simon Osindero and Yee-Whye Teh studies on an fast learning algorithm in Deep Belief Networks (BENGIO, 2009; WASON, 2018).

According to Bengio (2009), *architecture depth* refers to the number of levels of nonlinear operations (polynomials rather than single variable ones) in a given function. For the author, models until 2006 were limited to ***shallow architectures***, with 1 to 3 hidden layers, while a mammal brain works with multiple levels of abstraction, each level corresponding to a specific area of the cortex. The human brain seems to process information through multiple stages of transformations and representations, more particularly in vision, which undergoes processing stages as detection of limits, basic shapes and gradually develop to more complex visual forms.

In a similar manner Arel, Rose and Karnowski (2010), points out discoveries in Neuroscience during the decade of 2010s, describing the functioning of the human neocortex as a large flow of sensory signals that propagate through a complex hierarchy of modules. Over time, it learns to represent observations based on the regularities presented on signals.

4.4.1 The Hinton, Osindero and Teh (2006) proposition

Notable was the constant search for artificial neural models based on layer depth similar to the brain (FAUSETT, 1994; ENGELBRECHT, 2007; HAYKIN, 2009; BENGIO, 2009; RUSSELL; NORVIG, 2010; AREL; ROSE; KARNOWSKI, 2010). As computational capacity availability showed considerable improvements over the years, more complex tasks became object of these artificial models. As mentioned in item 4.2.3, Samuel (1959) was one of the first experiments intending to conceive a neural network aimed at problem resolution, at the time, a *punishment/reward* routine through checkers. In short, network functioning is based on a large look-up table, where all board positions are stored and game development predictions are made at each movement.

Sutton (1988) calls Samuel (1959) approach as *temporal-differential*, where the focus resides on error or difference between successive predictions on a temporal scale, standing apart from traditional supervised learning approaches produced so far, where focus was on difference and error in actual results. Fundamentally, so-called traditional operating approaches at the time presented a data set formed by input/output pairs where the first record refers to the analysis parameter for a given prediction and the second to the expected result. The author calls this approach *one-step prediction*. On the other hand, in temporal-differential methods, prediction assertiveness is not revealed until more than one step after the prediction is revealed, attending to the fact that relevant information can be found at each analysis step. Weather forecast situation is then analyzed: forecast accuracy for Wednesday would be measured based on Tuesday and

[2] Bengio (2009), in a footnote on page 6 of his work, cites previous advances in networks with a special structure called convolutional. The same very brief remission is also found in Bengio, LeCun et al. (2007), in reference to LeCun et al. (1989) and LeCun et al. (1998)

Monday results. The author argues that this method is more efficient, primarily because it is incremental and, for this reason, easier to compute. Second, because it tend to converge faster and show better predictions.

Subsequently, Tesauro (1992) analyzes the results presented by Sutton (1988), bringing unaddressed questions in three aspects: task-dependent considerations; algorithmic considerations; and representational considerations.

a. **Task-dependent considerations:**

   Learning to predict and control simultaneously: what is the nature of the problem — simple prediction or prediction followed by action? The second case appears to be more complex and, possibly, would be better addressed through an second neural network that performs the choice of action to be taken;

   Stationary vs. changing tasks: can tasks change over time? And even if they don't, is there a possibility that the distribution of input attributes will change? In both cases, it is recommended that the network be constantly updated with these possible changes;

   Markovian vs. non-Markovian tasks: the transition between the states of the network is *Markovian*, that is, whether it depends solely and exclusively on the current state or whether it depends on the history of previous states. This point was not directly addressed, as only *Markovian* processes were analyzed. There is a remission that *non-Markovian* processes could be included in the proposal by storing information from each current state together with all relevant information from previous states. In a practical way, this would be unfeasible given the need for a large storage space;

   Multiple outcomes: simplest reinforcement tasks have binary outcome states (success/fail signal) but more complex tasks have multiple possible outcomes. The way in which these results are represented in the network can be as important as the representation of the input signals itself. Additionally, some results may be easier to obtain than others, which makes learning more difficult.

   Noisy environment: is the environment noisy or deterministic? Noise can be identified in the rules which governs state transitions, in final signal in terminal states as well as in the representation of input patterns presented to the network.

b. **Algorithm considerations:**

   Parameter tuning: it would be recommended that parameters like learning rate α and the amount of related states λ for a TD(λ) function are adjustable. For example, starting the network with a high value for λ can help achieve better results in α, but as the learning rate increases, smaller values tend to have better performance;

Convergence: TD(λ) is limited to linear networks (where the activation rate is constant throughout the network, that is, there is no activation function described in the item 4.3.1.3) and sets of linearly independent input patterns (they can be represented in dimensions that are totally independent of each other, on another words, there is no interference from one pattern or attribute on another). In more general cases, the algorithm may not converge to a local optimization (described in 4.3.3), much less to a global optimization (also described in 4.3.3);

Scaling issues: no results were presented on how speed and quality of learning provided by TD(λ) will scale with temporal length of sequences to be learned, the dimensionality of the input spaces nor the dimensionality of the network. Tesauro (1992) performs intuitive analysis in the sense that the training time must increase drastically, possibly exponentially, with the increase in the length of the temporal sequence. In the same sense, there is a possibility of deficient scheduling with the growth of the network and of the input signals, for example, in the case of a high noise incidence in the training data.

Overtraining and overfitting: theoretically, given the dynamic nature of the training data set generated through TD(λ) methods (online, generated for each state under analysis), overtraining would not be applicable. In the same sense, overfitting would not be applicable, since the number of hidden processing units in the network could be increased. However, both phenomena can occur if minimized error function used during training does not match the desired function by the user. For example, in the case used in Sutton (1988), an algorithm can make great move predictions, but may not choose the best moves to win a game. It is entirely possible for an algorithm to make predictions that are not very accurate, but make good choices especially in cases where the best move is only slightly better than the others. Another case in which the phenomena can occur is if the training set is formed by simulations where the network plays against itself and latter is put to the test in situations where it has to play against other players - there are differences in input patterns distribution, given the nature of how data is produced, in this case, the agent's game style context;

Incremental learning: influence weights adjustments can be performed at each TD(λ) analysis step. Despite being considered an advantage, at the time the analysis is undertaken, computational and storage power were capable of record considerable sequences of inputs and outputs, as long as the temporal sequences were reasonably short, although with considerable problems. In this sense, this advantage is questionable considering the expense of an increase in network performance.

c. **Representational issues:**

According to the author, the way in which input and output data are represented in connectionist approach multilayer networks is one of the most important factors to achieve successful practical applications of supervised learning procedures. Relevance of representations can also be applied to temporal-differential learning methods. Two basic forms of representation are identified:

> (a) lookup table representations, in which the network has enough adjustable parameters to explicitly store the correct output for every possible state in the input state space; and (b) compact representations, in which the number of adjustable parameters is much less than the number of states in the state space, and the network therefore has to capture the underlying regularity of the task. (TESAURO, 1992, p.262).

In the case of look-up tables, convergence for global and local optimization described by Sutton (1988) would only be possible through previously visiting all possible states that the function could assume, given that representation nature does not allow learning through estimation. On the other hand, in compact representations there is a need for greater structural complexity to represent the problem.

The author concludes by limiting effectiveness of TD(λ) methods when applied to more complex and large-scaled problems. The algorithm might not converge for prediction-only tasks, and would be highly unlikely to do it on prediction and control tasks. Even if it reaches convergence, it could be tied to a local optimization and, even if it can find good solutions, training time required to deal with problem size or temporal sequence length would be so unattractive that learning would effectively become intractable. Increased number of hidden analysis units, as well as increased network complexity could also lead to an unattractive learning time, both of them culminating on limitation of practical results.

This issue was addressed more directly by Bengio, LeCun et al. (2007) where the authors mention that *deep architectures* in 2006 were still poorly addressed in research papers. Focus majorly remained on *shallow architectures*, with two or three layers, referencing the work of Hinton, Osindero and Teh (2006) as an inflection point.

The difference in the implementation of Hinton, Osindero and Teh (2006) is its hybrid architecture nature: the first two hidden layers form an undirected associative memory while the other hidden layers form a directed acyclic graph that converts the associative memory representations into observables variables, like *pixels* of an image (HINTON; OSINDERO; TEH, 2006, p.1527-1528).

The starting point was a phenomenon called "*explain away*" that compares two rare and independent causes that become totally anti-correlated, for example, an earthquake (x) and a truck crash (y) for the perception of a house jumping (c). Figure 29 graphically presents the proposed situation.

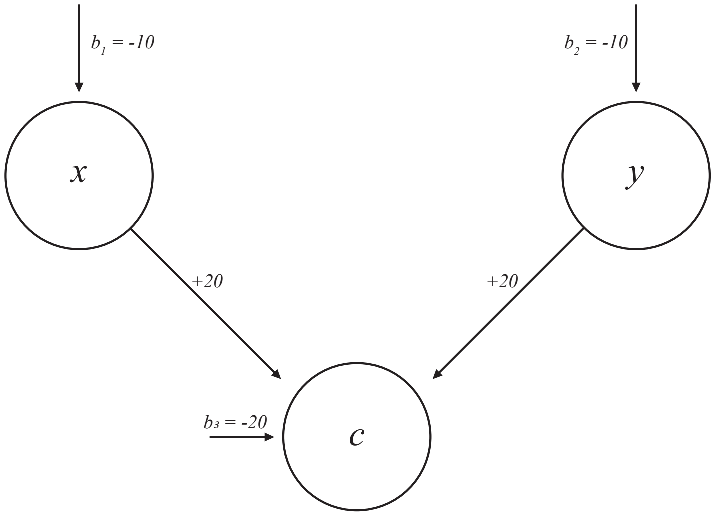


Figure 28 – Explain away example

Source: Adapted from Hinton, Osindero and Teh (2006)

Vectors b_1 and b_2 represent the activation trends of each of the causes x and y. The bias -10 means that, in the absence of any observation, the node is e^{10} more likely to be inactive than active. If the earthquake node is on and the truck node is off, the house jump node would have a total input of 0, meaning it would have an equal chance of being on. This house jump explanation is much better than relying on e^{20} chances of both causes being inactive. Practically, it would be useless to activate both, since the probability would be e^{-20}. If the earthquake node is enabled, "*explains away*" the truck node being off.

For this situation, the authors propose a concept called *complementary priors*. It is based on the results obtained by Neal (1992) on *logistics belief networks*, described as open to interpretation from two perspectives. On the one hand, it presents itself as a connectionist network with capabilities comparable to a *Boltzmann Machine*, but with improved learning performance. On the other hand, it presents how *belief networks* can be learned from empirical data as an alternative, or as a supplement to its previous specifications.

4.4.1.1 Boltzmann Machines

Ackley, Hinton and Sejnowski (1985) present the term *Boltzmann Machine* as a parallel constraint satisfaction network, involving a wide range of "*weak*" constraints. Constraint-satisfaction[3] typically use "*strong*" constraints that *must* be satisfied by any solution. Some problem domains have strong restrictions, such as the rules of a game. The authors analyzed the results obtained by Hinton (1977) in his doctoral thesis, whom states that even the best interpretations of a domain can incur in constraint violation on some degree. The domain treated by the author involved "*puppets*" of human form. Within a wide range of constraints, the author identified a group of only four that must be satisfied. In his words:

> The specific instructions which may be given as input, along with the picture, can alter the definition of the best puppet by attaching importances to the interpretation of rectangles as puppet parts, but the instructions cannot affect the four types of constraint that are listed above. So, for example, the program cannot be told to look for a one-legged or a three-legged puppet. The instructions are also unable to affect the relative proportions and the spatial relations which rectangles must have in order to depict a joint.(HINTON, 1977, p. 59.).

Ackley, Hinton and Sejnowski (1985) mention the possibility of measuring solution quality through the sum of all costs of each violated constraint, that is, measuring how implausible the referred interpretation is.

The authors present a parallel constraint satisfaction network, capable of learning the underlying constraints that characterize a domain through examples. The network modifies its connections strengths in order to build an internal *generative* model that produces other examples with the same probabilistic distribution. When a new instance is presented, it is "*interpreted*" by assigning values in the internal model that can "generate" the example. In an analogous sense, if a partial example (deprived of some attributes that characterize an object instance) is presented, values that generate the partial model would be searched and later used to generate the missing part.

They summarize this operation based on Hinton and Sejnowski (1983), defining the machine as a gathering of primitive computational elements called *units* (in a similar sense, but not identical to Rosenblatt (1961) Definition 8.), connected to each other through bidirectional links. A unit always presents a binary state, such as *on* or *off*, and adopts one of these states through a probabilistic function of neighboring units states combined with the weights attached to each of the respective connections. Weights can assume both positive and negative values. A unit being *on* or *off* means that, at that moment, the system accepts or rejects an elementary hypothesis regarding the domain in question.

The resulting structure bears a certain relationship with Hopfield (1982) Networks. It can be noticed the existence of superficial similarities with the *Perceptron* proposed by Rosenblatt (1957), however, the author is emphatic that the differences are the enablers of new results at that time. First of all, *perceptrons* modeling guidelines focus on neural connections oriented

[3] Dechter and Pearl (1988) define constraint-satisfaction problems as those that involve assigning values to variables that are bound by a set of constraints. Specifying them represents a convenient way of expressing declarative knowledge, allowing the solution designer to focus on local relationships between domain entities.

in a forward direction, such as $A \rightarrow B \rightarrow C$, since networks with strong retrograde coupling as presented below have been proven to be intractable.

$$\begin{tikzcd} A \arrow[r, leftrightarrow] \arrow[dr, leftrightarrow] & B \arrow[d, leftrightarrow] \\ & C \end{tikzcd}$$

However, better results obtained by the author came as a consequence of this high coupling. Second, most studies based on *perceptrons* put a network of neurons in direct contact with a real physical world without asking the essential questions to find the most emerging computational properties. Finally, the modeling of *perceptrons* suggests the use of synchronous neurons, while their asynchrony would bring more assertive achievements to the author's intent. Its operation is based on measuring the amount of energy linked to the correlated error in each state assumed by the model, that is, the greater the amount of accumulated energy, the greater the error linked to that state. The discovery of Hopfield (1982) is that, if the connections are symmetrical, it is possible to determine a global energy function (the sum of the quantity accumulated by each neuron in the network) that reduces the amount of energy to its minimum possible value.

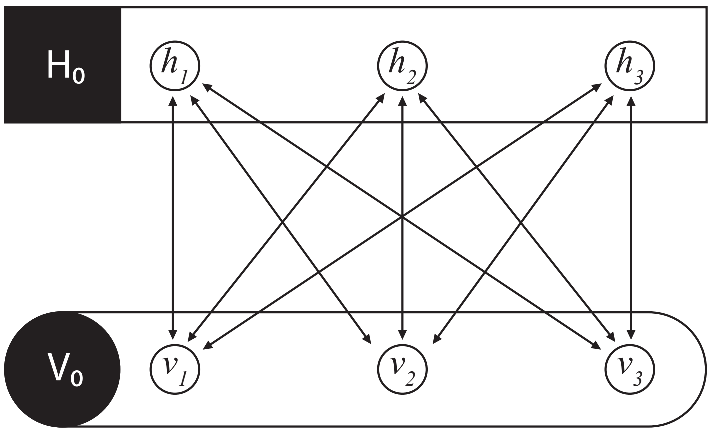


Figure 29 – Graphic representation of a Boltzmann Machine

Source: Produced by the author

Neal (1992) performed an analysis of the application of *Boltzmann* used by Hinton (1977). He concluded that the energy of a given configuration can be assumed as how critically a combination of hypotheses violates the implicit restrictions of the problem domain, leading to the conclusion that minimizing the accumulated energy of the system culminates in generating "interpretations" of inputs (in in this case, “puppets” of the human form) that gradually achieve better compliance with the aforementioned restrictions. The equation 4.24 calculates the accumulated energy in a *Boltzmann* model.

$$E(\tilde{S}) = -\beta \sum_{j>i} s_i s_j \mathcal{W}_{ij} \tag{4.24}$$

where:

- $\tilde{S}$ is a given input;
- s_i and s_j are the states of neurons i and j, respectively;
- $\mathcal{W}_{ij}$ is the connection weight value between neurons i and j. As connections are symmetric, $\mathcal{W}_{ij} = \mathcal{W}_{ji}$, since reflexive connections are absent(a neuron does not connect to itself);
- β is a constant of value 1, if the binary values assumed by neurons are 0 or 1. Its value will be $\frac{1}{2}$ if the assumed values are 1 or -1;

Energy is used to define a *Boltzmann* probability distribution across states, in which lower energy states are more probable than higher energy states. More specifically,

$$P(\tilde{S} = \tilde{s}) = \frac{exp(-E(\tilde{s}))}{Z} \tag{4.25}$$

where Z is a normalization factor which guarantees that the sum of all states probability result in 1:

$$Z = \sum_{\tilde{s}} exp(-E(\tilde{s})) \tag{4.26}$$

The model typical design encourage the use of “hidden” neurons. However, for the analyzes undertaken, only the marginal distribution of visible units is necessary. The vector $\tilde{s}$ is then considered as a pair $\langle \tilde{x}, \tilde{y} \rangle$ and, similarly, a variable $\tilde{S}$ becomes $\langle \tilde{X}, \tilde{Y} \rangle$. The distribution along the visible units becomes:

$$P(\tilde{Y} = \tilde{y}) = \sum_{\tilde{x}} P(\tilde{S} = \langle \tilde{x}, \tilde{y} \rangle) \tag{4.27}$$

Considering that the normalization factor Z can only be obtained through the sum of an exponential amount of terms, to directly calculate the probability of a particular state vector in large-scale networks becomes unfeasible. Even if such a calculation could be performed back than, the time needed to calculate the marginal probability of a visible vector, or the probability distribution for a subset of visible units from the values of the others would be exponentially greater than the number of hidden units. For these distributions in particular, the author cites the existence of a procedure called *Gibbs sampling*, also known as *Metropolis algorithm*, as its first appearance dates back to the work of Metropolis et al. (1953), which defines a simulated method of calculating properties of any substance that can be described as a composition of individual interacting particles (METROPOLIS et al., 1953, p.3). The simulation starts with the network in an arbitrary state. At each revisit cycle, each analysis unit has its value changed according to the probability distribution conditioned to the values of the other units. To produce a sample based on this distribution, the process needs to be run until a "balance" is found.

The biggest issue faced when using a *Boltzmann* machine is to adjust weights so that the probability distribution of visible units is as close as possible to the probability distribution of attributes in the real world. Adopting estimation through *maximum-likelihood* (since the goal is to achieve the realest probability according to a sample), we will have the likelihood expression:

$$\begin{aligned} V &= log \prod_{\tilde{y} \in \mathcal{T}} P(\tilde{Y} = \tilde{y}) \\ &\sum_{\tilde{y} \in \mathcal{T}} log P(\tilde{Y} = \tilde{y}) \end{aligned} \tag{4.28}$$

Where $\mathcal{T}$ is the training set, which can contain repeated instances. Since the probabilistic distribution in question addresses only the visible units, the weight of a particular unit is obtained by calculating a partial derivative, expressed as follows:

$$\frac{\partial L}{\partial \mathcal{W}_{ij}} = \beta \sum_{\tilde{y} \in \mathcal{T}} \left[\left(\sum_{\tilde{s}} P(\tilde{S} = \tilde{s} \mid \tilde{Y} = \tilde{y}) s_i s_j \right) - \left(\sum_{\tilde{s}} P(\tilde{S} = \tilde{s}) s_i s_j \right) \right] \tag{4.29}$$

Two *Gibbs* sampling phases $\left(\sum_{\tilde{s}} P(...) s_i s_j\right)$ can be observed, where the difference between them lies in the training scope $\mathcal{T}$. In the "positive" phase of the expression, it is noted that the visible units are "locked" to constant values on training set, resulting in a sample of states of the conditional state $\tilde{S}$ where $\tilde{Y} = \tilde{y}$. In the "negative" phase of the simulation no unit is "stuck", producing an equal-sized sample of the unconditioned $\tilde{S}$ distribution. For each state vector $\tilde{s}^+$ in the positive phase of the sampling, the weight $\mathcal{W}_{ij}$ is increased in quantity proportionally to $s_i^+ s_j^+$. Conversely, in the negative phase, for each vector $\tilde{s}^-$, the weight $\mathcal{W}_{ij}$ is decreased proportionally to $s_i^- s_j^-$. These two operations are repeated until convergence is reached.

Neal (1992) concludes that the need for both positive and negative phases comes from the normalization factor Z when calculating the probability of a vector state. The steepest descending direction in energy amount is not the same as the steepest ascending direction in probability. Here's why a negative sampling simulation phase is needed — it provides a mechanism to stop learning. When the increment of the positive phase is canceled by the negative phase, it is said that weight stability is reached. Although being of great importance, the negative phase has several disadvantages:

a. Increases computational volume (on greater than two factor);

b. Can make the learning procedure more sensitive to statistical errors;

c. May reduce any neurological plausibility the schema has.

4.4.1.2 Belief Networks (Bayesian Networks)

For Neal (1992), belief networks are also known as "Bayesian networks", "causal networks", "influence diagrams" or "relevance diagrams", designed to represent the probability distribution over a set of attributes (NEAL, 1992, p.77.). According to the author, the study of these networks was motivated mainly by the desire to represent specialized human knowledge.

For Pearl (1988), *Bayesian* methods provide formalism to reason about partial beliefs under conditions of uncertainty. In this formalism, propositions receive numerical parameters representing the degree of belief accorded them under some body of knowledge. Parameters are combined and manipulated according to the rules of probability theory.

The author makes a comparison with *Markov* networks, pointing out their inability to represent induced and non-transitive dependencies: two independent variables will be directly connected by a vertex, only because a third variable depends on both, which makes it impossible to represent multiple useful independencies in the network. To overcome this limitation, the author mentions that *Bayesian* networks use a richer language of *directed graphs*, where the direction of the arrows allows to differentiate genuine dependencies from spurious ones, arising from hypothetical observations. On the author's practical example:

> (...) if the sound of a bell is functionally determined by the outcomes of two coins, we will use the network *coin 1* → *bell* ← *coin 2*, without connecting coin 1 to coin 2. This network reflects the natural perception of causal influences; the arrows indicate that the sound of the bell is determined by the coin outcomes, which are mutually independent.(PEARL, 1988, p.116.).

He finalizes the definition, expressing that Bayesian networks are directed acyclic graphs — DAGs — in which the nodes represent variables, arrows denote the existence of direct influence causes between the connected variables and the strength of these influences are expressed by conditional probabilities. These conditionals come from the logical product generalization rule (equation 4.30) into conditional probabilities — $P(A|B)$ — which specify the belief in A assuming that B is known with absolute certainty. Beginning with

$$\begin{gathered} P(A,B) = \frac{P(A,B)}{P(B)}, \\ P(A,B) = P(A|B)P(B) \end{gathered} \tag{4.30}$$

we can extend an interpretation based on the *Markov chain rule*, which says that each probability depends only on the outcome of its immediate preceding, stating that in the case of a set of n events $E_1, E_2, ...E_n$, the probability of the set $(E_1, E_2, ...E_n)$ can be demonstrated by the product of each n probabilities

$$P(E_1, E_2, ...E_n) = P(E_n \mid E_{n-1}, ..., E_2, E_1)...P(E_2 \mid E_1)P(E_1) \tag{4.31}$$

extending to the core of the *Bayesian* techniques that lies in its inversion formula,

$$P(H \mid e) = \frac{P(e \mid H)P(H)}{P(e)} \tag{4.32}$$

which states that the belief attributed to a hypothesis H when obtaining an evidence "e" can be calculated by multiplying the previous belief $P(H)$ by the probability $P(e \mid H)$ that "e" will be confirmed if H is true. $P(e \mid H)$ is sometimes called posterior probability (or, in short, *later*) just as $P(H)$ is referred to as anterior probability (or *previous*).

Based on the explanations made by Pearl (1988), Neal (1992) describes that the probability of a vector state in a *Bayesian* network, called "forward conditioned probabilities", is the probability that an unit possesses a certain value conditioned to the values of the units that precede it:

$$P(\tilde{S} = \tilde{s}) = \prod_i P(S_i = s_i \mid S_j = s_j : j < i) \tag{4.33}$$

Conditioned odds are taken as given by an expert. Ordinarily, only part of the units that precede an unit "i" will be "connected" to it, and only these will be relevant in defining the forward conditional probabilities in "i" . In this case, the order of the units in the state vector also plays a key role since they determine which conditioned probabilities must be specified.

It can be seen that, contrasting with *Boltzmann* machines, in belief networks the probability of a particular state vector is strictly forward, that is, it does not have backward connections. For Neal (1992), the only plausible method of obtaining samples of conditioned distributions in highly connected accreditation networks is *Gibbs sampling*. As in *Boltzmann* machines, each step of the simulation requires the definition of a new value for unit "i" among its distribution conditioned to the values of the other units. On belief networks, the distribution proportionality is

$$\begin{aligned} P(S_i = x \mid S_j = s_j : j \neq i) \\ \alpha P(S_i = x \mid S_j = s_j : j < i) \\ \prod_{j>i} P(S_j = s_j \mid S_i = x, S_k = s_k : k < j, k \neq i) \end{aligned} \tag{4.34}$$

however, it must be considered that carrying out full calculations of forward conditional probabilities tends to be a very complex task, since specifying the distribution of S_i given the values of the predecessor units requires 2^{i-1} parameters. Even though some of the preceding units are not connected to unit "*i*", a more compact form of specification is needed.

For these situations, Pearl (1988) clarifies that, in a practical way, human mind conceptualizes causal relationships creating hierarchies of small clusters of variables and the interaction between factors in each cluster are usually categorized into prototypes of prestored structures. In Pearl (1986), he cites as examples of these structures the *noisy-OR* gate (any one of the factors is the probable cause of the result), the *noisy-AND* gate (when all the factors, together, are the probable cause of the result) and several enabling mechanisms (factors that have no other influence than the activation of other factors).

The situation faced by Pearl (1986) and Hinton, Osindero and Teh (2006) fits the situation of a *noisy-OR* gate. Neal (1992) describes that the model assumes analysis units as being binary gates (with value 0 or 1) where the input sign is the conjunction of previous units. An input signal of value 1 does not entail the obligatory assumption of the value 1 to another unit. The author mentions that there is a probability q_{ji} that even if a unit "*j*" takes on a value of 1, it will not be able to force a unit "*i*" to also assume 1 as its value. By means of this model the forward conditional probabilities can be expressed in terms of q_{ji}

$$P(S_i = 1 \mid S_j = s_j : j < i) = 1 - \prod_{j<i,\, s_j=1} q_{ij} \tag{4.35}$$

The author then describes two types of belief networks. The first is characterized as a generalization of the "*noisy-OR*" model for specifying conditioned probabilities. The second, comes from an analogy with *Boltzmann* machines, called sigmoid belief networks.

4.4.1.3 Logistic Belief Networks

Hinton, Osindero and Teh (2006) call Neal (1992)'s sigmoid belief networks as logistic belief networks, composed of stochastic binary units. When the network is used to generate data, the activation probability of an unit "*i*" is a logistic function of the states of its immediate predecessors "*j*" and the weights "w_{ij}" of the directed connections from the predecessors:

$$p(s_i = 1) = \frac{1}{1 + exp(-b_i - \sum_j s_j w_{ij})}, \tag{4.36}$$

where b_i is the bias of unit "*i*". If the logistic belief network has only one hidden layer, the prior probability distribution over the hidden variables is factorial since their binary states are chosen independently when the model is used to generate data (HINTON; OSINDERO; TEH, 2006, p.1531).

Returning to the problem of "explaining away", the non-independence in the posterior distribution is created by the likelihood term coming from the data, that is, even if they are independent in the previous state (it would be very unlikely for the house to tremble due to a truck crash after an earthquake), the posterior state can be independent (one of the causes practically eliminates the other).

The authors propose inserting more hidden layers in order to create a "complementary" prior: the added layers present the exactly opposite correlations to those in the likelihood term ("tying" the values of the influence weights), so that the product between these layers results in a posterior state that is exactly factorial. The learning algorithm is improved by “untying” the weights of a certain layer to the weights of the upper layer.

The learning situation proposed by the authors starts with the generation of data through the assumption of a directed network endowed with infinite hidden layers, starting with a random configuration in an infinitely deep layer.

Figure 30 presents the model of an infinite logistic belief network. Blue arrows represent the generative data model. The orange arrows are not part of the model, they just represent the parameters that are used to infer samples from the posterior distribution in each hidden layer of the network when a data vector is clamped in V_0.

The authors cite that this procedure gives rise to an error-free sampling as each previous level complements the next, ensuring that the posterior distribution is, in fact, factorial. This conclusion extends, due to the origin of the posterior state is true, the possibility of calculating derivatives of the logarithmic probability of the data.

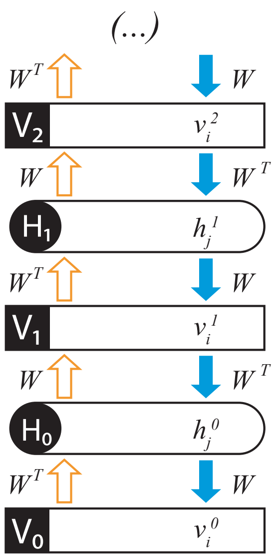


Figure 30 – Infinite logistic belief net model

Source: Adapted from Hinton, Osindero and Teh (2006)

Initiating on layer H_0, we compute the derivative for the generative weight w_{ij}^{00} from a unit "j" in layer H_0 to a unit "i" in the V_0 layer. Under these conditions, in a logistic belief network, the maximum likelihood learning rule for a single date vector v^0 is

$$\frac{\partial\, log\ p(v^0)}{\partial w_{ij}^{00}} = \langle h_j^0(v_i^0 - \hat{v}_i^0)\rangle \tag{4.37}$$

where $\langle ... \rangle$ denotes an average over the sampled states and $\hat{v}_i^0$ represents the probability that unit "i" is activated if the visible vector was stochastically reconstructed from the sampled hidden layers. Computing the posterior distribution over the second hidden layer V_1 from the sampled binary states of the first hidden layer H_0 is characterized by being the same data reconstruction process. Thus, v_i^1 is a sample of a random variable from *Bernoulli*[4] with probability $\hat{v}_i^0$. The learning rule can be written as

$$\frac{\partial\, log\ p(v^0)}{\partial w_{ij}^{00}} = \langle h_j^0(v_i^0 - v_i^1)\rangle \tag{4.38}$$

It can be noticed in the derivation of the equation 4.37 to 4.38 the dependence of v_i^0 on h_j^0, which does not present any problem since $\hat{v}_i^0$ is an expectation, a probability that is conditioned on h_j^0. Extensively, given that the weights are replicated, the complete derivative for the generative weights between all pairs of layers is

$$\frac{\partial\, log\ p(v^0)}{\partial w_{ij}} = \langle h_j^0(v_i^0 - v_i^1)\rangle + \langle v_i^1(h_j^0 - h_j^1)\rangle + \langle h_j^1(v_i^1 - v_i^2)\rangle + \ldots \tag{4.39}$$

all paired products, excepting the first and the last, cancel each other out. Hinton, Osindero and Teh (2006) stated that the proposed logistic belief network is equivalent to a restricted *Boltzmann* machine — RBM — with the difference that an RBM has only one hidden layer with symmetrical connections with the visible layer. Data sample generation process in an RBM is the same used in an infinite belief network (starts in an infinitely deep layer), both ending when an equilibrium distribution is reached. Paired cancellation leaves only the *Boltzmann* machine learning rule presented on 4.40

$$\frac{\partial\, log\ p(v^0)}{\partial w_{ij}} = \langle v_i^0 h_j^0\rangle - \langle v_i^\infty h_j^\infty\rangle \tag{4.40}$$

Previously, Geoffrey Hinton (2002) described that maximizing log probability of the data is the same as minimizing the Kullback and Leibler (1951) divergence between two probability populations, described as $D_{KL}(P^0 \| P_\theta^\infty)$ for P^0 data distribution and equilibrium distribution P_θ^∞ defined by the model. He calls this procedure *contrastive divergence learning*, detailing its functioning of *Gibbs* sampling applied on a *Markov* chain as shown in figure 31.

[4] In probability theory, a process from *Bernoulli* is a finite or infinite sequence of binary random variables that take on a value of 1 for true or 0 for failure

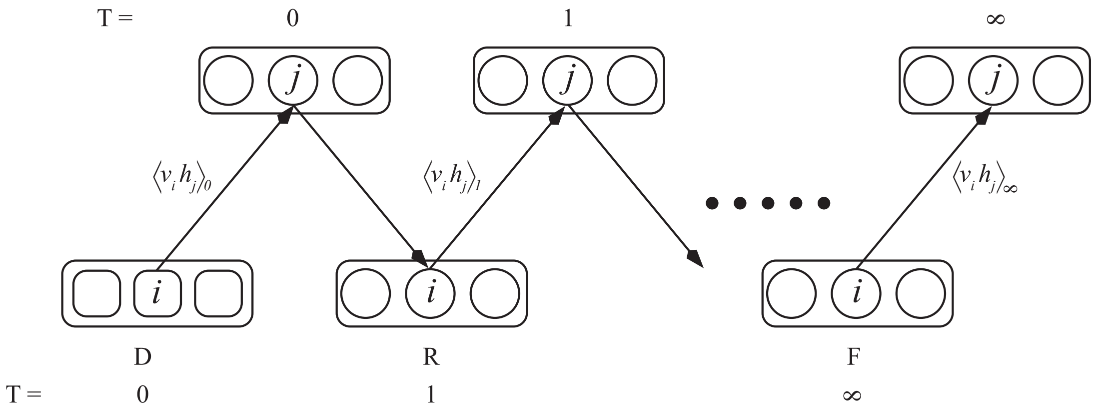


Figure 31 – *Markov* chain using alternate *Gibbs* sampling

Source: Adapted from Hinton (2002) and Hinton, Osindero and Teh (2006)

The goal is to perform complete rounds of *Gibbs* sampling *n* times before proceeding to the subsequent correlation *n+1*. In a practical way, it starts with the subtraction $D_{KL}(P^0 \| P_\theta^\infty) - D_{KL}(P^1 \| P_\theta^\infty)$, followed by the reducing the difference between P^0 and P_θ^1 and then updating the parameters to reduce the *Markov's* chain tendency of moving away from the initial distribution which would end on reconstruction "*R*" in time T = 1 of the original data set "*D*" at T = 0.

This procedure is equivalent to ignoring the derivatives that come from the higher layers of the infinite network. The sum of the derivatives of the ignored layers corresponds to the derivative of the logarithmic probability of the distribution posterior to the layer V_n which, in turn, corresponds to the derivative of the *Kullback-Leibler* divergence between the posterior distribution in the layer V_n , P_θ^n, as well as the equilibrium distribution defined by the model. In this way, *constrative divergence learning* minimizes the difference of two *Kullback-Leibler* divergences (HINTON, 2002; HINTON; OSINDERO; TEH, 2006):

$$D_{KL}(P^0 \| P_\theta^\infty) - D_{KL}(P_\theta^n \| P_\theta^\infty) \tag{4.41}$$

Despite the advancements obtained through *constrative divergence learning*, it was noticed that the process is not efficient on deep multilayer networks with different weights at each layer, due to computational complexity and time needed to obtain minimum balance for a vector with "stuck" data, even in obvious applications.

4.4.1.4 Hinton, Osindero and Teh (2006) algorithm

Neural network complexity is given according to the problem it faces. When large analysis capacity is necessary, demanding deeper layers structure, learning becomes a problem. One of the most efficient ways to deal with this issue is to divide the complex model into simpler models, and them executing in a sequence. Two previous techniques that divides the problem can be analyzed.

Freund (1995) uses as basis an algorithm originally proposed by Schapire (1990) that references *hypothesis boosting* identified by Kearns and Valiant (1994). The technique describes a sequence of models that complement each other through errors identified at each step of its execution.

Friedman and Stuetzle (1981), on their method for searching projections, seek interpretation of high-dimensional data through lower dimension projections.

The idea behind the algorithm proposed by Hinton, Osindero and Teh (2006) is to allow each sequence model to receive a different representation of the data set. Each instance performs a nonlinear transformation of its input vectors, producing output vectors that will be used as input to the next instance.

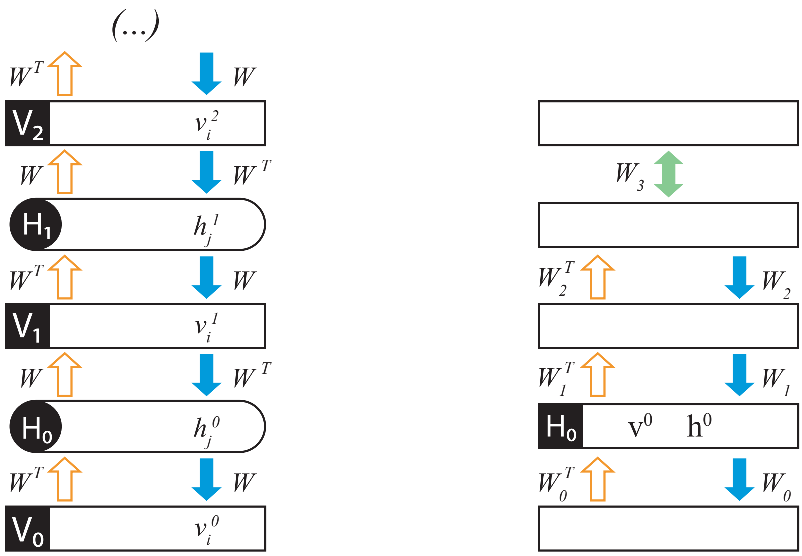


Figure 32 – Comparing a logistic belief network and the hibrid model proposed by Hinton, Osindero and Teh (2006)

Source: Adapted from Hinton (2002) and Hinton, Osindero and Teh (2006)

The right side of figure 32 shows Hinton, Osindero and Teh (2006) hybrid architecture. The top two layers have undirected connections (green arrows), which are equivalent to an infinity sequence of hidden layers with "tied" weights, behaving like an associative memory. The layers below have generative directed connections in a top-down direction (in blue) that can be used to map a particular state of the associative memory. It is also observed the presence of directed bottom-up recognition connections (in orange), which are used to infer a factorial representation in one layer from the binary activities in the lower layer. All layers have the same number of analysis units and do not have links between units of the same layer.

Due to the characteristics described, it is assumed that it is possible to find sensitive values (although not optimal) for the weight $\mathbf{W}_0$ assuming that all parameters comprised between the upper layers will be used to build a complementary prior to $\mathbf{W}_0$. Thus, learning W_0 becomes something similar to learning a RBM. Although being a difficult task, good approximations can be obtained through contrastive divergence. Once $\mathbf{W}_0$ is learned, the data can be mapped through $\mathbf{W}_0^T$ to create a high-level representation on the first hidden layer. Ordinarily, this representation obtained by the RBM will not be a perfect model of the original data. In this regard, the proposed algorithm acts as it improves the generative data model as follows:

i. Learn $\mathbf{W}_0$, assuming all the weight matrices are tied;

ii. Freeze $\mathbf{W}_0$ and committing to use $\mathbf{W}_0^T$ to infer factorial approximate posterior distributions over the states of the variables in the first hidden layer, even if subsequent changes in higher-level weights mean that this inference method is no longer correct;

iii. Keeping all the higher-weight matrices tied to each other, but untied from $\mathbf{W}_0$, learn an RBM model of the higher-level "data" that was produced by using $\mathbf{W}_0^T$ to transform the original data.

If the upper layer weight matrices are changed by this algorithm, generative model improvement is guaranteed. This conclusion was explained on Neal and Hinton (1998), which defined that the negative logarithmic probability of a data vector "$\mathbf{v}^0$" in a multilayer generative model is limited to the result of the subtraction between the amount of energy expected in the approximate distribution $Q(\mathbf{h}^0|\mathbf{v}^0)$ and the entropy of that probability distribution.

For a directed model, the "energy" of the configuration $[\mathbf{v}^0, \mathbf{h}^0]$ is given by

$$E(\mathbf{v}^0, \mathbf{h}^)) = -[\log p(\mathbf{h}^0) + \log p(\mathbf{h}^0|\mathbf{v}^0)], \tag{4.42}$$

so the bound is

$$\log p(\mathbf{v}^0) \geq \sum_{todos\ \mathbf{h}^0} Q(\mathbf{h}^0|\mathbf{v}^0)[\log p(\mathbf{h}^0) + \log p(\mathbf{h}^0|\mathbf{v}^0)] - \sum_{todos\ \mathbf{h}^0} Q(\mathbf{h}^0|\mathbf{v}^0) \log Q(\mathbf{h}^0|\mathbf{v}^0) \tag{4.43}$$

where:

- "$\mathbf{h}^0$" is a binary configuration of the units in the first hidden layer;
- "$p(\mathbf{h}^0)$" is the prior probability of "$\mathbf{h}^0$" under the current model (which is defined by the weights above "H_0");

- "$Q(\bullet|\mathbf{v}^0)$" is any probability distribution over the binary configurations in the first hidden layer;

- the bound becomes an equality if and only if "$Q(\bullet|\mathbf{v}^0)$" is the true posterior distribution.

When all of the weight matrices are tied together, the factorial distribution over "H_0", produced by applying $\mathbf{W}_0^T$ to a data vector is the true posterior distribution, so at ii. of the model, the value $\log p(\mathbf{v}^0)$ is equal to the bound, that freezes "$Q(\bullet|\mathbf{v}^0)$" e "$p(\mathbf{h}^0)$". and with these terms fixed, the derivative of the bound is the same as the derivative of

$$\sum_{todos\ \mathbf{h}^0} Q(\mathbf{h}^0|\mathbf{v}^0) \log p(\mathbf{h}^0) \tag{4.44}$$

which makes it possible to conclude that maximizing the bound with respect to the weights of the higher layers is the same as maximizing the log probability of a data set in which "$\mathbf{h}^0$" has a probabilistic incidence of $Q(\mathbf{h}^0|\mathbf{v}^0)$. The proposed process is based on layer-by-layer learning, as shown in figure 33.

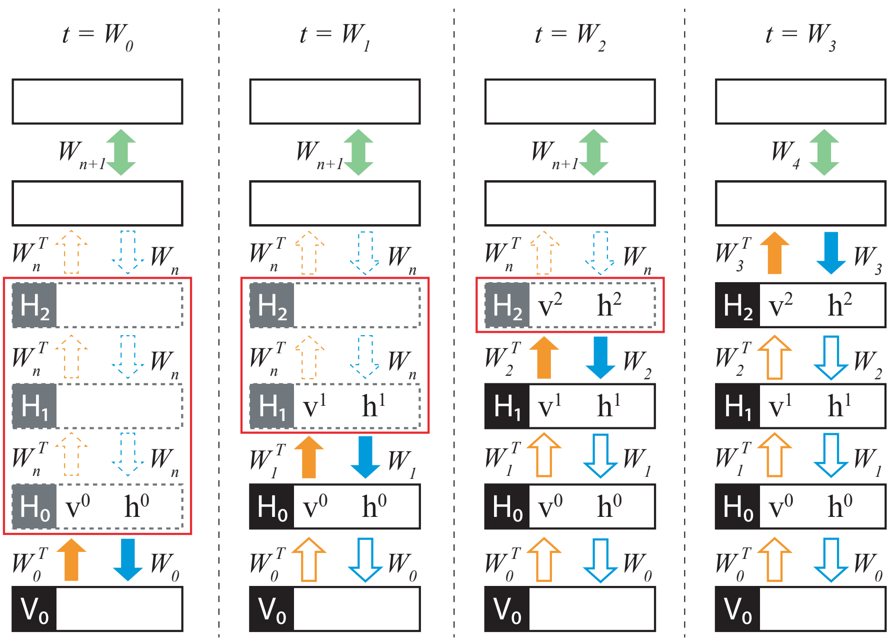


Figure 33 – Exemplified Hinton, Osindero and Teh (2006) learning process

Source: Produced by the author

It can be noticed that the "layer tying" process forms a complementary prior whose posterior probability distribution is exactly factorial. As learning time advances from $t = W_0$ to $t = W_3$, the set of "tied" weight matrices decreases as the level of the layer being treated (learned) gets deeper. Thus, as higher level weights are learned, the complementary prior ones obtained on lower levels are no longer applicable as factorial distributions and also affecting the generative weights values inferred. According to the authors, the generative model produced suffers from limitations: it was originally designed to recognize images in which non-binary values can be treated as probabilities, which does not apply to natural images. However, it can be considered as a milestone in dealing with multilayered learning.

4.4.2 Deep Neural Networks Models

After the advances obtained by Hinton, Osindero and Teh (2006) it was possible to address two major problems: the *Society of Mind* cited by Minsky and Papert (1988), referring to the ideal quantity of analysis layers to be inserted on a neural network; and the *data dimensionality* cited by Bellman (1954) and Arel, Rose and Karnowski (2010). The latter, more clearly, was addressed in Hinton and Salakhutdinov (2006), which uses the RBM to reduce the dimensionality of images.

Following the analyzes carried out by Arel, Rose and Karnowski (2010), is added to these problems the need to consider a temporal component as a fundamental matter when dealing with real manifestations. Ordinarily, a sequence of patterns can be meaningful to an observer. At the other hand, presenting only isolated fragments of the same sequence can make interpreting too complex: meaning is usually inferred by observing events with short temporal difference, for both identifying distortions of the same object or distinguishing distinct objects (MIYASHITA, 1988; EDELMAN; WEINSHALL, 1991; FÖLDIÁK, 1991; MIYASHITA, 1993; STRYKER, 1991; SINHA; POGGIO, 1996; WALLIS; ROLLS, 1997; WALLIS; BADDELEY, 1997; WALLIS; BÜLTHOFF, 1997; WALLIS, 1998; STONE, 1998).

4.4.2.1 Convolutional Neural Networks

A convolutional network is a multilayer perceptron specifically developed to perform two-dimensional shapes recognition with a high degree of distortion invariance. Its architectural design, according to LeCun et al. (1998), adopts three guidelines for dealing with the problem: local receptive fields, shared weights (or weights replication) and spatial or temporal sub-sampling (LECUN et al., 1998, p.6.).

The combination of techniques allows better feature extraction of analyzed instances, which leads to better pattern recognition by the network. Originally, scientific basis for this set of praxis comes from the discoveries made by Hubel and Wiesel (1962) about the existence of locally-sensitive (does not communicate with neighbor units) and orientation-selective (respond to a certain pattern of movement) neurons in cats visual cortex. Despite the initial application on image recognition, the aforementioned architectural design is also applied in sound recognition.

In a simplified way, combining the definitions presented in LeCun, Bengio et al. (1995) with LeCun et al. (1998), also referenced in Haykin (2009) and exemplified through figure 34, the techniques have the following attributes that contribute to pattern recognition:

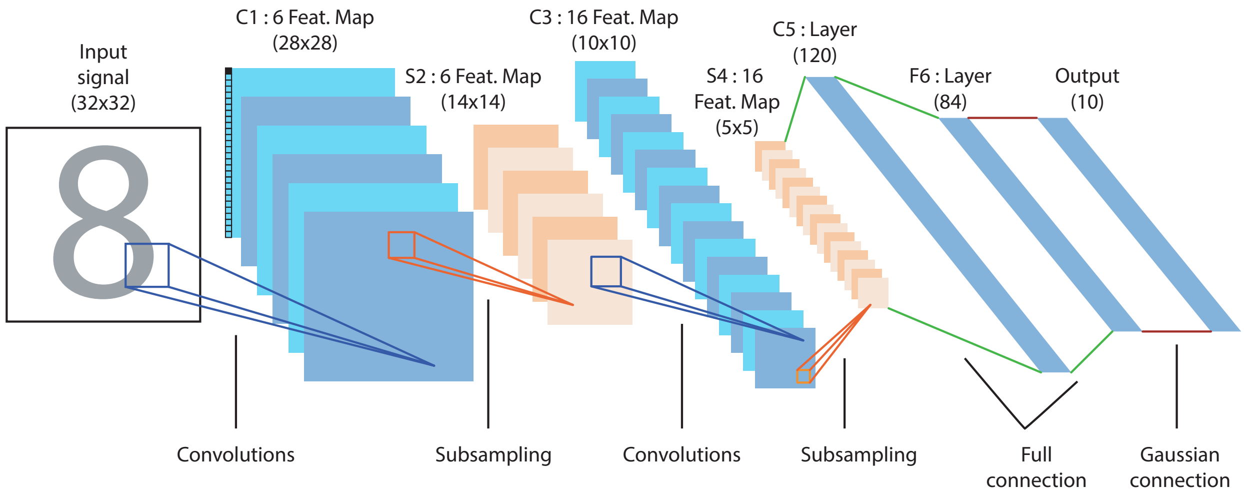


Figure 34 – Architectural model of LeNet-5 presented in LeCun et al. (1989).

Source: Adapted from LeCun et al. (1998)

a. Local receptive fields or feature extraction: each analysis unit (neuron) receives only input signals from a small group of neurons from the previous layer, which are located close to the referenced unit. This approach makes it possible to identify features such as oriented edges, endpoints, corners or other attributes in other signals such as speech spectrograms. These identified features are then combined by the subsequent layer. In figure 34, the input signal to be analyzed is a 32x32 *pixels* image.

b. Weight sharing or feature mapping: eventual distortions in input signals can cause relevant features position to shift. Furthermore, such features can be repeated in several parts of the same instance (an image, a sound or other manifestation). This "knowledge" can be represented in a network by constraining a set of neurons, whose receptive fields are located in different parts of the instance, to share the same set of synaptic weights. These groupings are called *feature mapping*. Units of the same map look for the same attributes in different parts of the image and learning takes place in all layers, simultaneously, in order to obtain a more assertive set of activation functions in each map.

Map analysis takes place in a convolutional and simultaneous way. In figure 34, shows that the 6 (six) attribute maps obtained in the first convolutional layer "C1" comes from a reduction in image complexity, going from 32^2 *pixels* to 28^2 *pixels*. Figure 35 presents a simplified representation of the convolution layer "C1". The origin of each attribute map comes from the overlapping of the receptive fields of each unit, as well as in the sequencing of the outputs of each unit. This superposition followed by the product of the units is equivalent to a mathematical operation of **convolution**.

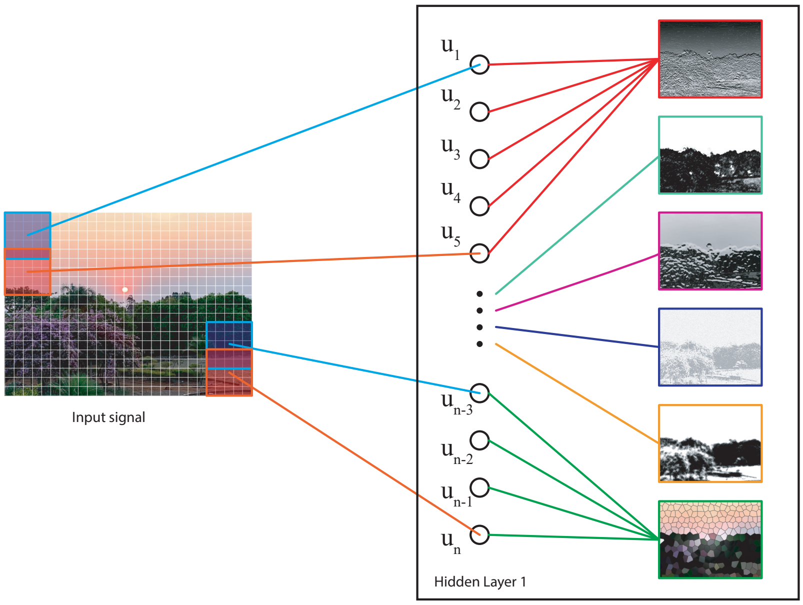


Figure 35 – Convolutional layer example

Source: Produced by the author em Junho de 2021

c. **Temporal or spatial subsampling:** once features are detected, their exact location becomes less important. Only its approximate position related to other features is relevant. After each convolutional stage of the network, follows a computational that performs local averaging and subsampling, which leads to a reduction in the resolution of the attribute maps. Figure 36 demonstrates the sequencing of convolutions and subsamplings operations.

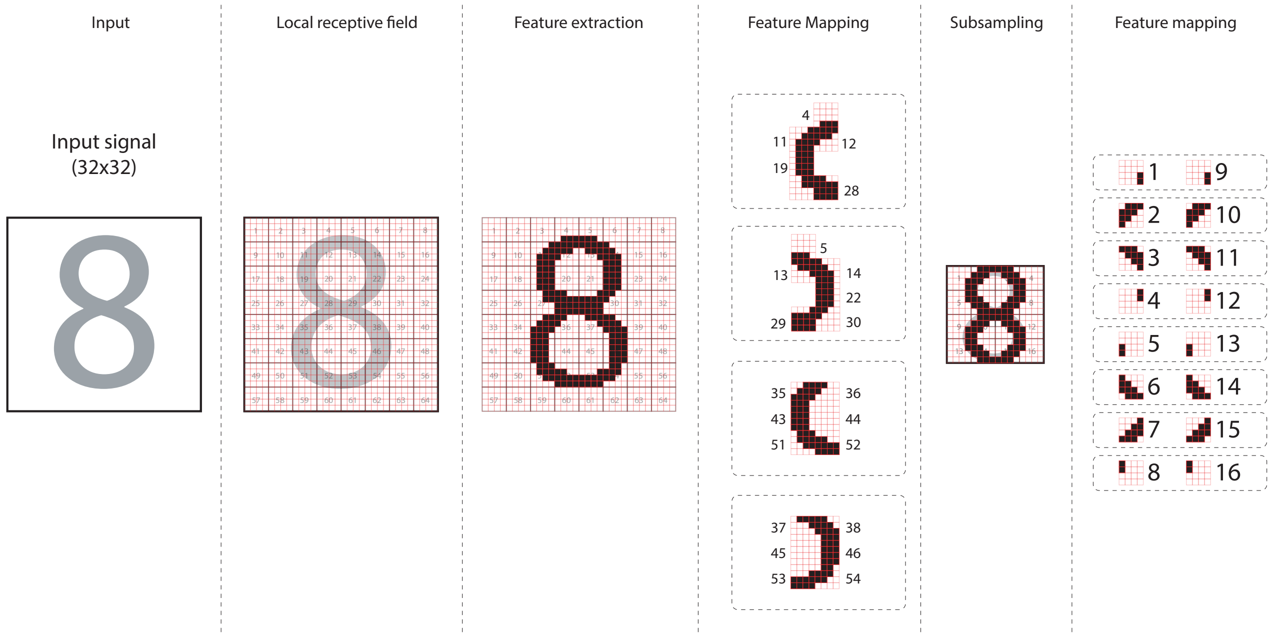


Figure 36 – Simplified example of sequencing convolutional and subsampling operations

Source: Produced by the author in june, 2021

During convolutional rounds followed by subsampling, a phenom described as "bipyramidal" by Haykin (2009) can be observed, that is, with each operation performed, the number of feature maps increases, while the spatial resolution (receptive field size) decreases. This reduction of problem size is also noticed in the number of free parameters along the network when considering weight sharing property. The author points out two advantages obtained in its implementation, when compared to fully connected multilayer networks:

1. Better generalization: as the quantity of free paramenters diminishes, categories classifying criteria also diminishes, making learning machine capacity reduced, but improving its generalization ability.

2. Parallelization ability: another noteworthy point of weight sharing is that as different analysis units have the same weight, processing can be taken in different units, simultaneously, presenting the same result.

4.4.2.2 Recurrent Neural Networks

As stated by Jordan (1986), an aspect that cannot be removed in a large part of human behavior is the serially ordered characteristic of its actions, in an unfolding of events that follows one another in time. For the author, temporal sequencing is closely linked to parallelism of initiatives. He distinguishes, in this manner, two kinds of parallelism: when the actions in a sequence overlap during execution, characterizing a parallelism along the execution time; and when two actions must be executed in parallel, given the nature of the task or some other implicit restriction, characterizing parallelism along the execution space.

Initial approaches to temporal sequencing of actions in human behavior were based on reflexes chains, where the results of each element of the sequence provide parameters for activating the next ones, forming an associative chain (LASHLEY, 1951, p.114). In this sense, action ordering would take place through direct connections between control elements that represent them and, consequently, a sequence performance would be measured by the path taken through the network of these control elements.

Lashley (1951) points out that this practice limits ordering possibilities applicable to a set of actions, since there is no mechanism that can indicate which connection should be activated if an action has two or more possible results.

Wickelgren (1969) proposes a reinterpretation of the associationist approach, adding a context analysis of action sequencing applications, which, according to the author, makes it possible to face Lashley (1951) issues. It begins with the concept of control elements on a network, defined by a local representation of actions in which an action is represented by a single unit (a control element), and activation of that unit causes the action to be executed (JORDAN, 1986, p.6). Therefore, considering that neuron B is activated, the sequence [$_A$,B,$_C$] would be totally different from [$_C$,B,$_A$] and presented as two possible elements in the same network. Three major shortcomings can be listed in this attempt. First, it requires a large number of elements and still cannot deal with pronunciation of words with repeated subsequences of sounds of length two or more (as an example, b*ar*ny*ar*d). Second, effects of context are generally extended to more than four of five phonemes forward in an utterance, enhancing the effects of the first issue. Lastly, as the theory only treats phonemes as tokens, it cast aside the concept of types. This means that the semantic context is not taken in account.

Jordan (1986) also addresses a different approach described by Fowler (1980) and Rumelhart and Norman (1982). Both articles make a stand that actions are not taken solely but in parallel, that is, several control elements can influence behavior at the same time. It enables treatment of context sensitivity, partially addresses temporal ordering of actions, but has difficulty dealing with sequences in which there are repeated occurrences of actions. In order to address serial order issues completely, Jordan (1986) proposes a simple distinction between networks given its overall connectivity:

> An important distinction can be made between networks based on their overall connectivity. If a network has one or more cycles, that is, if it is possible to follow a path from a unit back to itself, then the network is referred to as recurrent. A nonrecurrent network has no cycles (JORDAN, 1986, p.5).

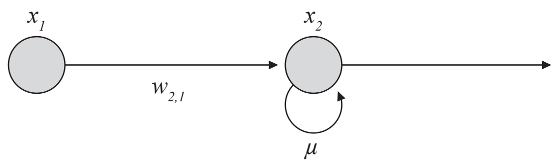


Figure 37 – Jordan (1986) simple recurrent network model

Source: Adapted from Jordan (1986)

On figure 37 is presented a simple model of recurrent network, where μ is the value of the recurrent weight. Mathematically, the activation of unit x_2 at time t would be obtain trough

$$\begin{aligned} x_2(t) &= \mu\, x_2(t-1) + w_{2,1}\, x_1(t) \\ &= \mu^t\, x_2(0) + \sum_{\tau=0}^{t=1} \mu^\tau\, w_{2,1}\, x_1(t-\tau) \end{aligned} \tag{4.45}$$

where $x_1(t)$ is assumed to be constant over time, that is, the input value is always the same. Considering the equation applied to a simple recurrent network, the trajectory would reach a constant state if μ has value less then one, and would go to infinity if μ reaches larger values.

From this simple representation Jordan (1986) constructed a *Theory of Serial Order*, considering one major constraint: the input vector $\boldsymbol{p}$ cannot be modified during processing. The choice for the letter $\boldsymbol{p}$ means exactly that what is ***planned*** cannot be modified — is the goal to be achieved and serves primarily to designated the particular sequence which is to be performed. It also leads to assume that temporal order of input signals are not considered in this proposition. As in general it is desirable that the system to be able to produce different sequences, different vectors $\boldsymbol{p}$ would lead to different sequence of actions. Consequently, ***plans*** can be arbitrary patterns of activation serving as keys to particular sequences, excluding the interpretation of then as systematic scripts to be followed.

To reproduce temporal context, each actions has to be taken considering other actions nearby in time. These temporal "neighbors" define the context in which the system is inserted at the time and serve as guidance for deciding which action should be taken. Jordan (1986) defines that the system makes this decision based on a representation of the context in form of a state vector, considering two functions. First, a function $\boldsymbol{f}$ representing the output action $\boldsymbol{x_n}$ at time $\boldsymbol{n}$

$$\boldsymbol{x_n} = \boldsymbol{f}(s_n, \boldsymbol{p}) \tag{4.46}$$

then, a function $\boldsymbol{g}$ which determines the next state $\boldsymbol{s_n} + 1$

$$\boldsymbol{s_n} + 1 = \boldsymbol{g}(s_n, \boldsymbol{p}) \tag{4.47}$$

both depending on the current state vector s_n. A model of the network proposed is presented at figure 38. As the author proposed, three pools of processing units can be identified. *Plan units* and *State units* serve as *Input units*. As the output function $\boldsymbol{f}$ is generally nonlinear, there is a need for hidden units between input units and output units. Recurrent connections implement the next-state function $\boldsymbol{g}$ departing from the state units to themselves and from output units to state units, allowing the current state to depend on both previous (as there is recurrent connections from the output) and current state (state units self-connections).

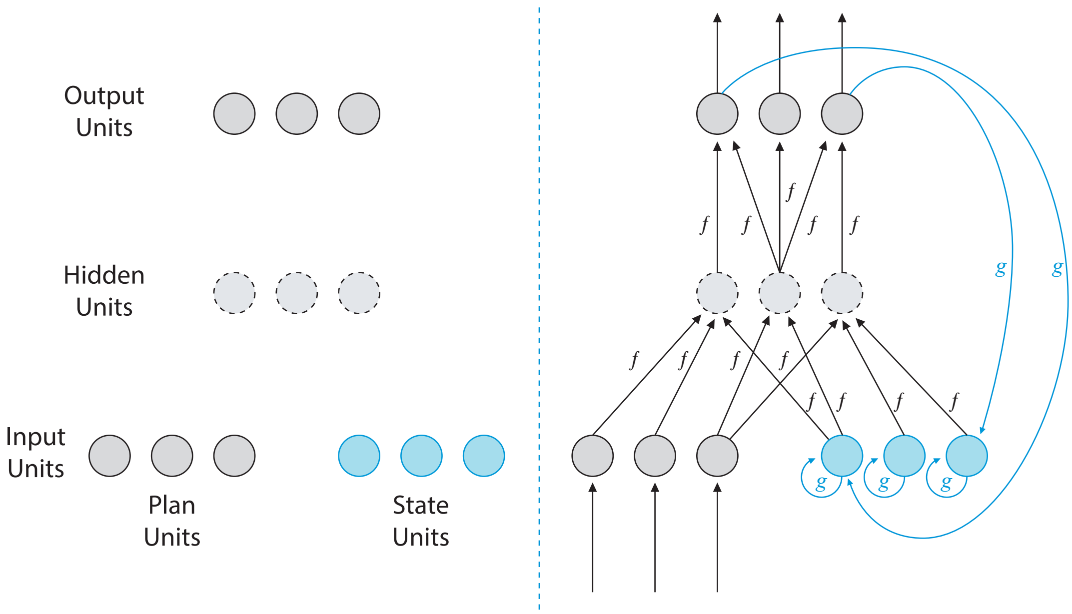


Figure 38 – Jordan (1986) simple recurrent network model. On left only the analysis units distribution and classification. On the right, a connection scheme (not all connections are shown)

Source: Adapted from Jordan (1986)

The proposed network does not present any explicit representation of temporal order and no explicit representation of action sequences. As there is only one set of output units for the network, only one output vector is presented at a time. These output vectors are produced in a dynamic manner (as the input signal is processed), not prepared in advanced, in a static buffer and serially executed. Learning only occurs on the $\boldsymbol{f}$ function, as $\boldsymbol{g}$ function is fixed in order to maintain a continuity property on the network.

The author conclude that state is the central concept on his theory, as time is represented implicitly by the configuration of the state vector that, in turn, is influenced by the configuration of all states related in time, keeping the sequential character of these relationships. Although the theory seemed promising, dual-task (parallel processing) and state similarity still challenging issues.

Elman (1990) proposed a modified version of Jordan (1986) network, generating the concept of *Internal Representation of Time*. On Jordan (1986) work, state units are the ones responsible for storing the configuration of the last and current context also being visible units (can be seen from outside the network). Instead, Elman (1990) network modifies the recurrent phase of the network, replacing the concept of state units with the concept of context units.

The main difference resides on the fact that context units are hidden (not visible from outside the network) and receive inputs from other hidden units. A comparative view between Jordan (1986) and Elman (1990) networks is shown on figure 39.

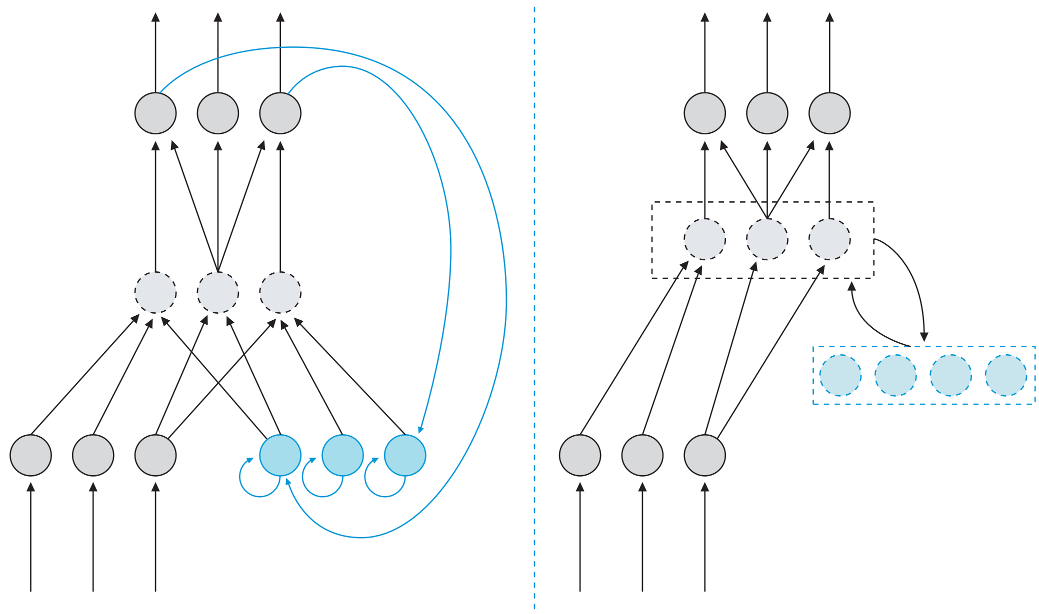


Figure 39 – Comparative graphic between Jordan (1986) and Elman (1990) networks

Source: Produced by the author in February 2022

After Elman (1990) proposition, it is noteworthy Schmidhuber, Hochreiter et al. (1997) method called ***Long Short-Term Memory — LSTM***, addressing a performance issue identified in several works, while dealing with the need to store information for a extended time interval. They observed that error signals back-propagation tend to assume exponential values (blowing up or vanishing), ether leading to oscillating weights on the network, taking prohibitive amount of time to learn or failing to implement the intended goal.

The main idea resides on reducing exponential deviations with a constant error flow with special, self-connected units called *memory cells*. These units are protected from irrelevant inputs sent over long period of time through input and output gates. These gates open or closes access to the central unit, which is self-connected and with fixed weight of value 1,0. Figure 40 demonstrate the proposal.

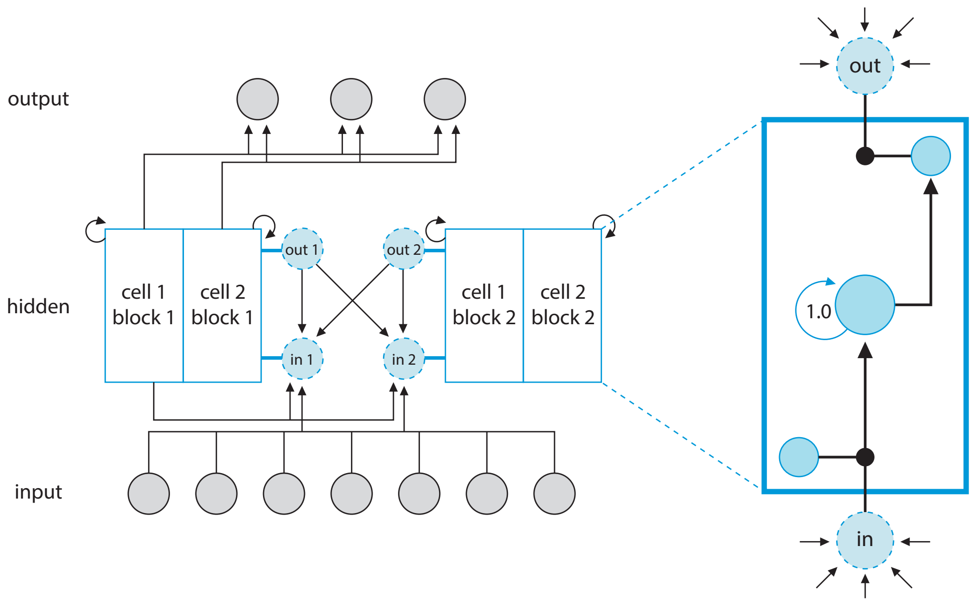


Figure 40 – A simplified model of a LSTM from Schmidhuber, Hochreiter et al. (1997)

Source: Produced by the author in February, 2022

4.5 Natural Language Processing — NLP: theory and praxis

For 16th century physician and psychologist *Juan Huarte*, the essential property of human intelligence resides in the mind's ability to "*engender within itself, by its own power, the principles upon which knowledge rests*" (CHOMSKY et al., 2006, p.viii).

Chomsky et al. (2006) mention that in the case of language, such *principles* are those of the internalized language (*I-language*) that a person acquires. Linguistics, in turn, seeks to discover true theories of *I-languages* (grammars) and, at a deeper level, the theory for the genetic basis for language acquisition (universal grammar).

Complementarily, Manning and Schutze (1999) defines the general objective of a linguistic science as to provide the ability to explain and characterize a multitude of linguistic manifestations that surround us in different ways: conversations, writing and other means. In three problems this issue materializes:

P.1 the cognitive side of how humans acquire, produce and understand language;

P.2 the understanding of relations between utterances and the world; and

P.3 the comprehension of linguistic structures by which language communicates.

For the authors, last question was commonly addressed by assuming the existence of rules that structure linguistic expressions. During the 20th century, approaches became too formalized and rigorous as linguists searched for detailed grammars capable of distinguishing well-formed propositions from poorly-formed ones. They conclude that, over time, such intent presents clear empirical problems: people tend to distort the proposed rules so that their communicative objectives are met.

4.5.1 NLP: Epistemological approaches

For Manning and Schutze (1999), in general, two approaches presented as epistemological basis for theories and models about language and its relations.

The **Rationalist** approach dominated studies in linguistics, psychology, artificial intelligence and natural language processing between the 1960s and the mid-1980s. This approach is characterized by the belief that a significant part of knowledge in the human mind does not derive from senses, but is fixed in advance, probably through genetic inheritance. Chomsky (1986) argues for the existence of an innate faculty of language, as a result of the *problem of poor stimuli*. He suggests that it is difficult to conceive that children can learn something as complex as natural language from limited variety and interpretability of the stimuli they receive over the years. In terms of artificial intelligence, such assumptions underlie attempts to design systems by hand coding a robust body of knowledge and early logical mechanisms in order to duplicate a working model of the human brain.

The **Empiricist** approach also assumes prior existence of cognitive abilities of the brain, but as a detailed set of specific principles and procedures for the various components of language and other domains of the mind. In this way, a child's brain would only be endowed with general operations of association, pattern recognition and generalization, which can be applied to the vast sensory stimuli available to promote natural language learning. It is mentioned that this approach was dominant during the 1920s to the 1960s, reappearing at the end of the 1990s. Applied to NLP, learning the complicated and extensive structure of language would take place through specifying an appropriate general model of language, proceeding then to parameters adjustments through statistical techniques, pattern recognition and machine learning applied to a large number of instances of language usage.

Harris (1951) is known as the most relevant Empiricist work, which describes a series of methods in structural linguistics presenting the following characteristics:

(Char.1) treat utterances which occur in a single language community at a single time. These procedures determine what may be regarded as identical in various parts of various utterances, and provide a method for identifying all the utterances as relatively few stated arrangements of relatively few stated elements;

(Char.2) also do not constitute a necessary laboratory schedule in the sense that each procedure should be completed before the next is entered upon. In practice, linguists take unnumbered short cuts and intuitive or heuristic guesses, and keep many problems about a particular language before them at the same time;

(Char.3) do not eliminate non-uniqueness in linguistic descriptions. It is possible for different linguists, working on the same material, to set up different phonemic and morphemic elements, to break phonemes into simultaneous components or not to do so, to equate two sequences of morphemes as being mutually substitutable or not to do so;

(Char.4) are consistent, but are not the only possible ones of arranging linguistic description.

Although **Rationalists** and **Empiricists** present similarities, they observe different objects. *Chomskian* (or generative) linguists seek to describe the linguistic module of the human mind (*I-language*) which stimuli like texts (*E-language*) are merely indirect evidence amenable to supplementation through intuitions of native speakers of the language. In turn, Empiricist approaches focus on describing the *E-language* as they occur. Another point of divergence lies in two notions proposed by Chomsky (1965):

> We thus make a fundamental distinction between **competence** (the speaker-hearer's knowledge of his language) and **performance** (the actual use of language in concrete situations). (CHOMSKY, 1965, p.4.).

Rationalists claim it is possible to isolate linguistic competence and describe it in isolation. Empiricists reject that hypothesis and seek to describe the actual use of language. For Manning and Schutze (1999), such difference comes from the interest in computational methods by empiricist techniques. Chronologically, between the 1970's and 1990's, sciences of the mind were largely discussed, which increased the number of attempts to conceive systems simulating intelligent behavior, that would address issues treated until today, even though, at the time, on much smaller scales (pejoratively designated as "toy-problems"). From the end of the 1990s, greater emphasis on engineering practical solutions that manipulate real texts can be noticed, as well as on objective comparative analysis of efficiency between the methods used. Such practices receive new terminology such as *Language Technology* or *Language Engineering* in detriment of NLP.

Additionally, when Chomskynian currents recognize the existence of conflicts between language principles, they resort to categorical principles, under which a given sentence or proposition is unsatisfactory or not. On the other hand, Statistical NLP departs from Shannon's ideas, where the objective resides in assigning probabilistic value for linguistic events, so that it is possible to say whether certain sentences or propositions are "usual" or "unusual". On a practical manner, while Chomskynian theories tend to focus on categorical judgments about rare types of sentences, NLP Statistics seek to describe associations and preferences that occur when completely using a certain language.

4.5.2 NLP: Scientific basis

In quick reference to Van Gigch and Moigne (1989) world view, epistemological questions give grounding to define objects of the scientific problem to be faced. Therefore, following the epistemological grounding defined in section 4.5.1, two main scientific currents will be treated: Chomskynian Generative for rationalist basis and Statistical NLP for empiricist basis.

4.5.2.1 Rationalist NLP: Chomskynian currents

As said on section 4.5.1, rationalist or Chomskynian currents seek to describe language in its totality, as a common I-language that derives pairs of [sound/sign, meaning]. On Chomsky, Gallego and Ott (2019) a list of basic operations and constraints is presented as being fundamental to any computational mechanism that seeks this goal. These definitions are based on the *Universal Grammar* — UG — thesis, and enriched with recent psychological experimentation and neurolinguistics developments.

For the authors, a traditional characterization of language defines it as "sound with meaning." Parting from that, an I-language would be a system that connects sound/sign and meaning in an orderly manner. Two non-negotiable empirical properties are considered:

(EP 1.) **Discrete infinity:** there would be no longest "sentence", meaning that there is no limit for the quantity of sign/sounds to form a sentence. The notion of "sentence" is replaced by the term *hierarchically structured set of objects*.

(EP 2.) **Displacement:** there is no universal structural order of terms that would restrict meaning formation applied to all possible I-languages grammars.

The first operation defined is **MERGE**, which is applied to two objects X and Y, yelding a new one K = [X,Y]. It differs from concatenation as it does not impose order (MERGE [X,Y] is the same as MERGE [Y,X]), presenting itself as the computationally simplest operation. It also can be applied recursively, sufficing the basic properties of discrete infinity and displacement. Two types of MERGE are distinguished:

(MER 1.) **External Merge — EM:** objects X and Y are distinct, that is, taken directly from the lexicon or independently assembled.

(MER 2.) **Internal Merge — IM:** on K = [X,Y], if Y is a term of X, then in K there will be two incidences of Y (either a word or a syntactic term). This method can turn Y into a *discontinuous object*, a *chain* that can be understood as a sequence of occurrences of Y in K.

All objects constructed by MERGE are mapped onto a semantic representation <SEM>, accessed by **Conceptual-Interpretive** (C-I) systems; and instructions to the vocal or gestural articulators, a phonetic representation <PHON> accessed by **Sensorimotor** (SM) systems. Each pair [PHON,SEM] correspond to a pair [sound/sign, meaning] derived by the I-language.

For being the simplest operation (exclude the concept of objects order) it cast aside of range languages which ruling and operations are defined in linear terms (e.g., "reverse the order of words in the sentence to yield a question"). This structure-dependence is not treated by UG, which defines only MERGE as an operator.

The main problem of its typically *ad hoc* application is the exclusion of features contained on syntactic objects, possibly leading an interface system to any assigned interpretation of expressions. For example:

Pesquisa e desenvolvimento duvida se foram rejeitados. (4.48a)

Research and development doubt if rejected. (4.48b)

(Há) Duvida se **pesquisa e desenvolvimento** foram rejeitados. (4.49a)

(There is) Doubt if **research and development** *(were)* rejected. (4.49b)

Considering only the MERGE operator, (4.48a) and (4.49a) (and their direct translation from Brazilian Portuguese to English (4.48b) and (4.49b)) would have the same semantic value, as their are only an rearrangement of terms [A,B,C,D,E,F,G] into [D,E,A,B,C,F,G].

A second operation called **AGREE** would come to relate features of syntactic objects. The asymmetric nature of the operation would relate unvalued unitary features to those contained on a certain goal within the set of objects analyzed. Considering the example above (4.49b), an AGREE operator would identify if it would be semantically correct the syntax IS/ARE in the case, that is, if the MERGE [research, development] is a singular subject (e.g., meaning an area of a company) or a plural one (e.g., steps of a process). Revisiting the example through this view would lead to:

Há dúvida se **pesquisa e desenvolvimento** *[é rejeitada/são rejeitados]*. (4.50a)

There is doubt if **research and development** *[is/are]* rejected. (4.50b)

Other question faced when considering only MERGE is the need for the objects created to be mapped to pairs [PHON,SEM] of the I-language. The authors propose an operation **TRANSFER** that hands constructed objects over to the mapping components, in order to access them through **C-I** and **SM** systems. The semantic component appears to be simpler to map, as the hierarchical structure of terms leads the meaning intended (e.g., in a simple example, "a subject does something"). Phonetic mapping is the main issue, due to influences of stress and prosodic contour, "flattening" of the hierarchical structure and other distortions related to the manner messages can be transmitted from case to case.

Another key point to be mentioned concerns syntatic derivation of displaced terms. Ideally, TRANSFER should map objects in a way that they cannot be modified by any further computation. This would lead to structure elimination generating another problem: there are cases where a certain term of the sentence is not presented in its original position, being displaced, partially or totally. Considering the example

$$[_{\alpha} \text{ o parecer técnico } [_{\beta} \text{ que aprova o projeto}]] \tag{4.51a}$$

$$[_{\alpha} \text{ the techical report } [_{\beta} \text{ that aproves the project}]] \tag{4.51b}$$

suppose that after TRANSFER of β is done, α is raised to a higher level of importance in the sentence as in 4.52a and 4.52b

$$[_{\alpha} \text{ o parecer técnico } [_{\beta} \text{ que aprova o projeto}]][_{\alpha} \text{ foi finalizado] pelo comitê.} \tag{4.52a}$$

$$[_{\alpha} \text{ the techical report } [_{\beta} \text{ that aproves the project}]][_{\alpha} \text{ was finalized] by the comitee.} \tag{4.52b}$$

The authors make an argument that on these cases, there is no structure loss, as the TRANSFER operation simply renders β accessible for syntactic purposes but inaccessible to subsequent manipulation.

As dictated by the authors, an syntactic object W is constructed through a derivational process of multiple **MERGE** and **AGREE** operations. This object is then subjected to **TRANSFER** to representational interfaces, mapping W onto <SEM> and <PHON>, accessed by **C-I** and **SM** systems, respectively. The main problem, as stated before, is <PHON>, as **C-I** system imposes a requirement of *Full Interpretation*: all terms of a syntactic object must be interpreted, that is, partial interpretation of an object (leading to two separated syntactic objects) can't be done. For instance, 4.53a and 4.53b can't be interpreted at **C-I** as either "Quem John viu?" ("Who did John see?") or "John viu Mary" ("John saw Mary"), ignoring the other terms.

$$[\text{quem, [John, [viu, Mary]]]} \tag{4.53a}$$

$$[\text{who, [John, [see, Mary]]]} \tag{4.53b}$$

The authors conclude that the approach based on **MERGE** can be considered as a progress, but the majority of aspects of I-language remains untreated. Furthermore, the insertion of this operation on the matter raised more questions concerning its conceptual and empirical applicability.

4.5.2.2 Empiricist NLP: Statistical NLP

At the other face of epistemological view of NLP, empiricist begin with two basic questions proposed by Manning and Schutze (1999):

(Question 1.) What kind of things people say, covering all aspects of the structure of language.

(Question 2.) What these things say/ask/request about the world, entering the fields of semantics, pragmatics and discourse, that is, how to connect propositions to the world.

The first issue is the core of *Corpus Linguistics*, defined by Kennedy (2014) as the study of the structural elements and patterns that make up a linguistic system, as well as the mapping of their use. A *corpus* is defined as a systematically planned and structured compilation of texts. The author distinguishes it from the definition of *text collection* or *text database*, which is characterized by a repository of texts, commonly collected opportunistically, unstructured and in large-volume (KENNEDY, 2014, p.3-4).

Due to the described nature of *Corpus Linguistics*, there is a close relationship between it and the use of computational machines, given the tendency to errors while manually operating large volumes of texts, which still remains a restrictive and slow practice. Furthermore, according to Manning and Schutze (1999), *corpora* patterns can be extensively interpreted into a deeper understanding of language manifestations, indirectly covering the fields addressed in the second question.

At the other hand, generative/rationalist linguistics abstracts away from any attempt to answer the first question, focusing on describing a competence grammar that is said to underlie the language (the I-language) (MANNING; SCHUTZE, 1999, p.8). Instead (and in extremely reduced extent, an attempt to approach **(Question 1.)**), it is suggested that there is a set of sentences — grammatical sentences — which are licensed by the competence grammar, leaving other strings of words as ungrammatical, leading to a concept of grammaticality:

> This concept of grammaticality is meant to be judged purely on whether a sentence is structurally well-formed, and not according to whether it is the kind of thing that people would say or whether it is semantically anomalous. (MANNING; SCHUTZE, 1999, p.8)

Even though this binary classification of sentences seems to bring some gains, it becomes extremely limited when considering real use of language. Firstly, is highly improbable that all sentences used can be classified as grammatical or ungrammatical. Secondly, a statistical study on real use of different sentences and sentences types can reveal nuances of communication. Two factors can be easily described on these matters:

i. **Conventionality:** a convention can be defined as a certain mode of expressing something, despite the fact that other ways are, in principle, possible. As Wittgenstein (1968) stated, the meaning of a word comes not only from its semantic value but also from the context of use.

ii. **Evolvability:** meaning of words and syntax of a language can change over time. A hypothesis is that the frequency use of a word in different contexts can gradually modify its original category resembling words from another category.

Both phenoms may only be observed if the general vision of language is not focused on categories but on the statistical and probabilistic use of it. These mutations of syntax and semantics are generally sudden and gradual. The details of this graduality are only revealed by examining frequency of use and measuring variances on strength of relationship between terms.

Another strong argument stated by the authors is that we live in a world filled with uncertainty and incomplete information. Our sensory receptors are constantly processing or discarding inputs so that the very nature of cognitive processes can be resumed through probabilistic (or at least quantitative) frameworks, therefore, dealing with uncertainty and incompleteness.

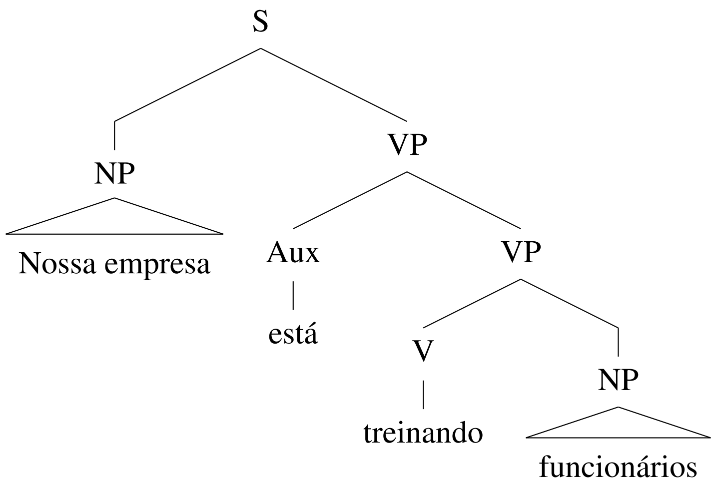
A Chomsky (1965) argument against statistical approaches is that even if notions like "likely to be produced" and "probable" sometimes gives a certain feeling of objectivity, they would produce an utterly useless notion, since ungrammatical sentences could be substituted by grammatical ones and the probability of both these sentences would be indistinguishable from zero. Manning and Schutze (1999) confront the argument stating that early probabilistic models were extremely simplistic, which hinder them to simulate the complexity of human language, therefore, as computational power grows, completeness of analysis would grow too. For the authors, the main issue do not reside in whether the probabilistic value is close to zero or not, but if statistical approaches could deal with meaning. In that case, a definition of meaning is crucial: from a statistical perspective, meaning would be the distribution of context over which words and utterances are used (MANNING; SCHUTZE, 1999, p.16).

The biggest obstacle for any NLP systems resides in language ambiguity, both in semantic and syntax. For the latter, a common procedure is called *Parsing*, which seeks to determine the syntactic structure of a sentence. Consider sentence 4.54a, which leads to three possible parsing (4.55a, 4.55b and 4.55c); and sentence 4.54b which also leads to three possible parsing (4.56a, 4.56b and 4.56c):

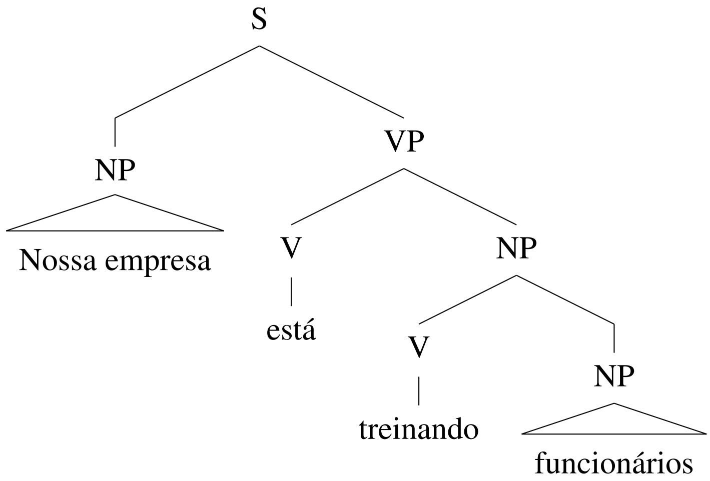
Nossa empresa está treinando funcionários. (4.54a)

Our company is trainning workers. (4.54b)

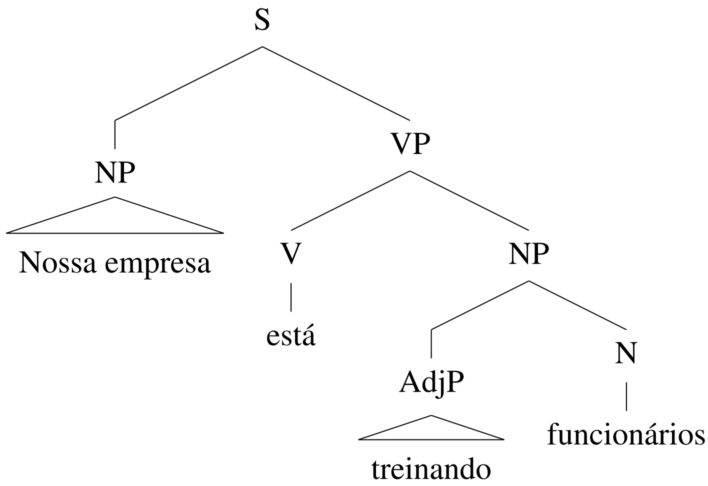
(4.55a)

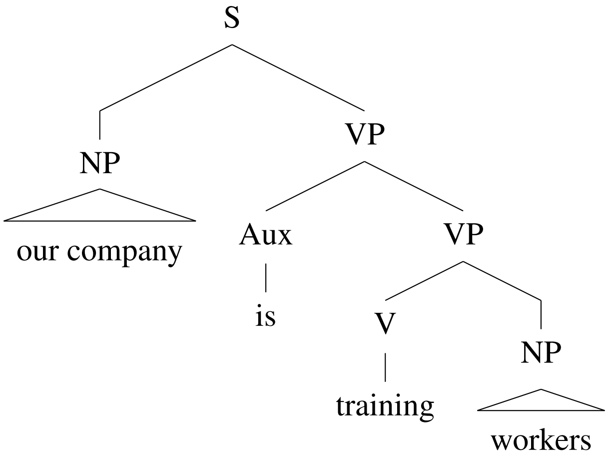


(4.55b)

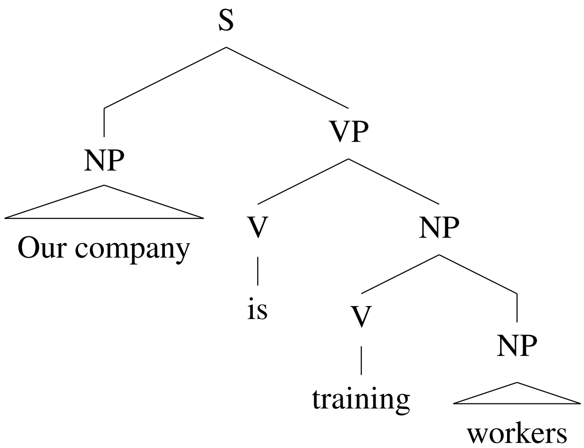


(4.55c)

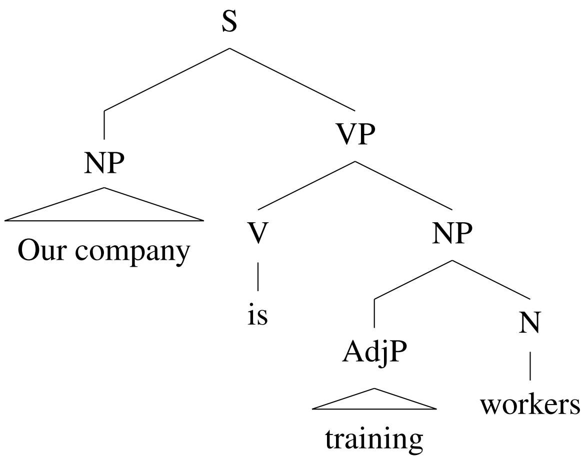




(4.56a)



(4.56b)



(4.56c)

Analysis 4.55a and 4.56a is the one humans perceive, where the phrase *is training* (*está treinando*) is a verb group that means the *act of training* the noun *workers* (*funcionários*).

Another possibility can be noticed in 4.55b and 4.56b, where *is* (*está*) is the verb and *training workers* (*treinando funcionários*) is a gerund, meaning the state of the noun phrase *our company* (*nossa empresa*).

A third possibility can be seen in 4.55c and 4.56c, where *training* (*treinando*) modifies *workers* (*funcionários*), meaning a characteristic of the noun phrase *our company* (*nossa* *empresa*).

From the same sentence, three syntactic structures can be produced using the same grammar. This kind of ambiguity will grow as sentences become larger and grammars get more comprehensive. In that manner, Lakoff (2008) says that hand-coding syntactic constraints and rules to fit all possible structures has proven to be time consuming, do not scale up well and operate poorly when face extensive use of metaphors in language. The observation opens a big gap for statistical NLP to fill: instead of parsing sentences alone, using syntactic categories, would it be more assertive if the analysis focuses on the relationship between words, that is, which words present a tendency to group with another ones given a certain circumstance? First step to address the question is to find resources to find this relations.

4.5.3 NLP: Relevant technological achievements

From epistemology through scientific definition, two paths have been distinguished with fundamentally different basis. Rationalists focus lies on the I-language and describing its categories. Empiricists, at the other hand, try to describe the E-language as it develops and used by humans. Either paths relies heavily on technological implementation able to capture context relations. For ANNs this is usually done with **encode-decode** operations, which encode data into a certain format (a vector, for example), apply a set of mathematical operations based on pre-defined rules, then decode the result into intelligible language again.

4.5.3.1 FFNN-based models

The first use of Feedforward neural network for language modeling, most called **NNLM**, was proposed by Bengio, Ducharme and Vincent (2000), addressing the *Data Dimensionality Problem* in learning joint probability function of sequence of words. As mapping connections of words within a vocabulary can become an intractable problem, the approach created the concept of ***Word Vector***, with the following characteristics:

[C. 1.] each word is associated with a distributed “feature vector” that create a notion of similarity between words;

[C. 2.] each feature vector represents different aspects of a word;

[C. 3.] each word is associated with a point in the vector space. As close a word is from another means the more relation both have between them.

As the main idea was to compute probability distribution over all the words in the vocabulary presented, it lacked performance. Departing from this idea, Mikolov et al. (2013a) proposed two architectures that have vector representation of word sequence and use of a projection layer as common points. The projection layer is intended to learn both word vector representation and a statistical language model.

The first architecture is named **Continuous Bag-of-Words (CBOW)**. It is based on NNLM with the difference that the projection layer is shared for all words, not only the projection matrix. This way, all words are projected into the same position and their vectors are averaged. The name *bag-of-words* makes reference to the fact that the order in which the words are presented does not make any influence but in this architecture, the context representation of these words is continuously distributed, giving birth to the name CBOW.

In a slightly different path **Continuous Skip-gram** model aims to maximize classification of a word based on another word in the same sentence. A single word is used as input to long-linear classifier with continuous projected layer in order to predict words that comes before and after the input word, within a certain range.

Both architectures need an initial embedding model of language to produce vectors for word representation. The most common used ones are Mikolov et al. (2013a)'s Word2vec and Pennington, Socher and Manning (2014)'s GloVe.

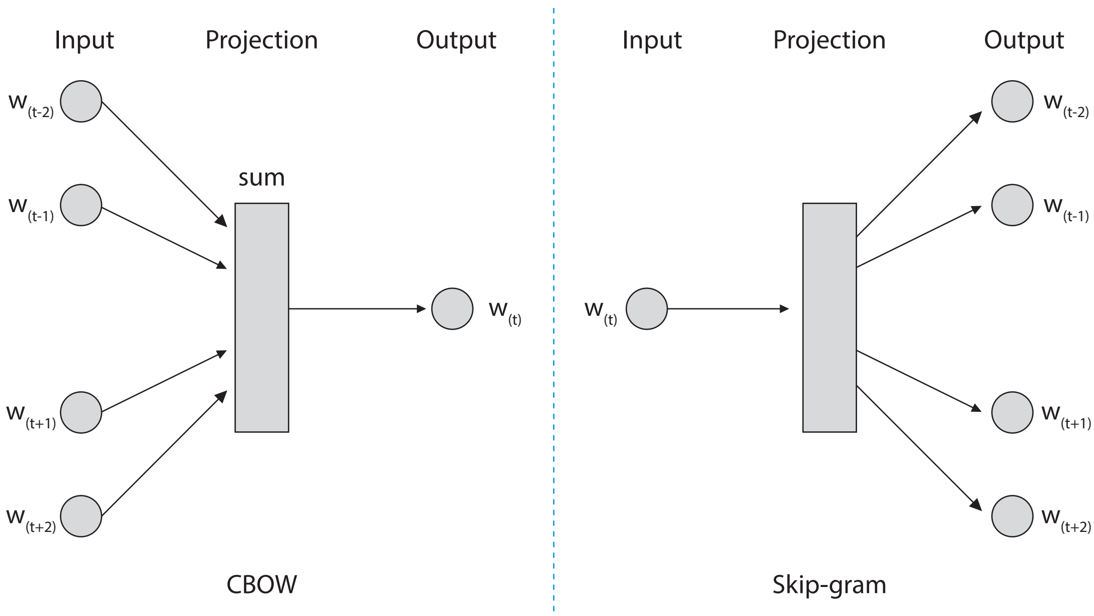


Figure 41 – CBOW and Skip-gram models representation

Source: Adapted from Mikolov et al. (2013a)

Another model developed by Iyyer et al. (2015) also utilizes the concept of bag-of-words as an input. Unlike other BOW models, it is much simpler and implements a dropout regularizer: for each training instance, randomly drop some of the tokens' embeddings before computing the average.

4.5.3.2 RNN-based models

As stated on section 4.4.2.2, RNN models aim temporal analysis of inputs, therefore, they view text as sequence of words. The main goal on utilizing these kind networks on NLP is to capture word dependencies and text structures. For text structure being sometimes long, the most popular architectures are based on the **LSTM** model. Works that utilize it presented improved results by capturing richer information such as tree structures of natural language and long-span word relations.

For Tai, Socher and Manning (2015) there are three classes for distributed representations of phrases and sentences: bag-of-words, sequence models and tree-structured models. Bag-of-words cast aside the order of words making it insufficient to fully capture the semantics of natural language. The authors find tree-structured models linguistic more attractive due to their relation to syntactic interpretation of sentence structure.

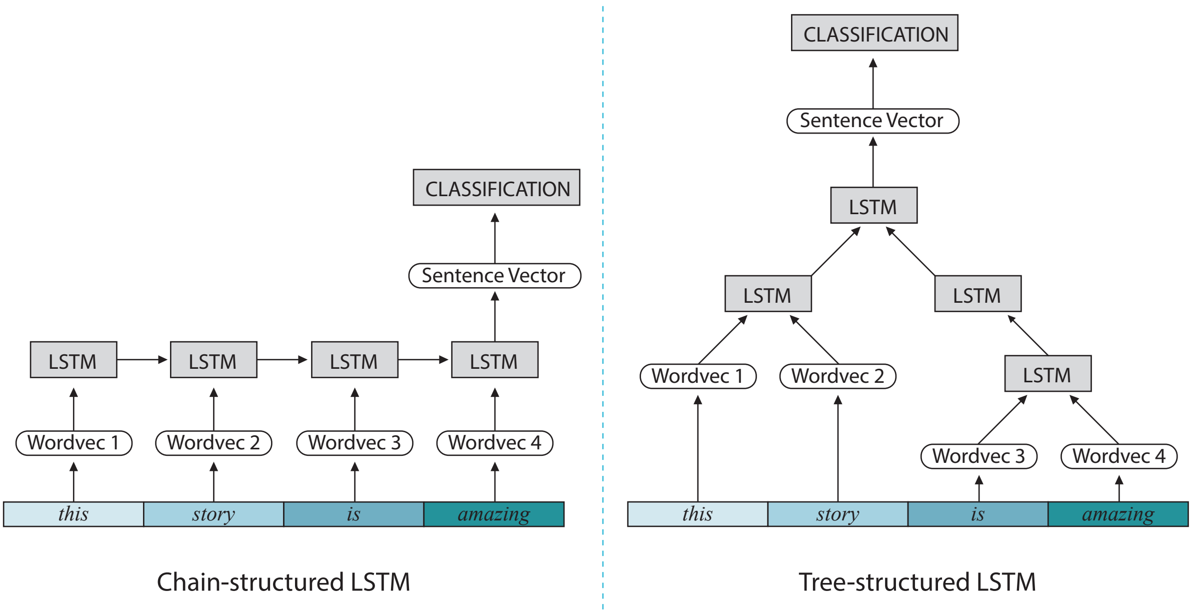


Figure 42 – Chain-structured and Tree-Structured LSTM graphic model

Source: Produced by the author in February, 2022

Addressing long text modeling (such as sentences and documents), Liu et al. (2015) developed the **Multi-Timescale LSTM** neural network. The idea is to capture valuable information with different timescales, that is, either shorter or longer period of time. This is done by separating the **LSTM** units into groups, which are activated at different time span. The first group g_1 is the fastest one and can be activated every time step, working as a standard **LSTM**. The last group g_k is the slowest one.

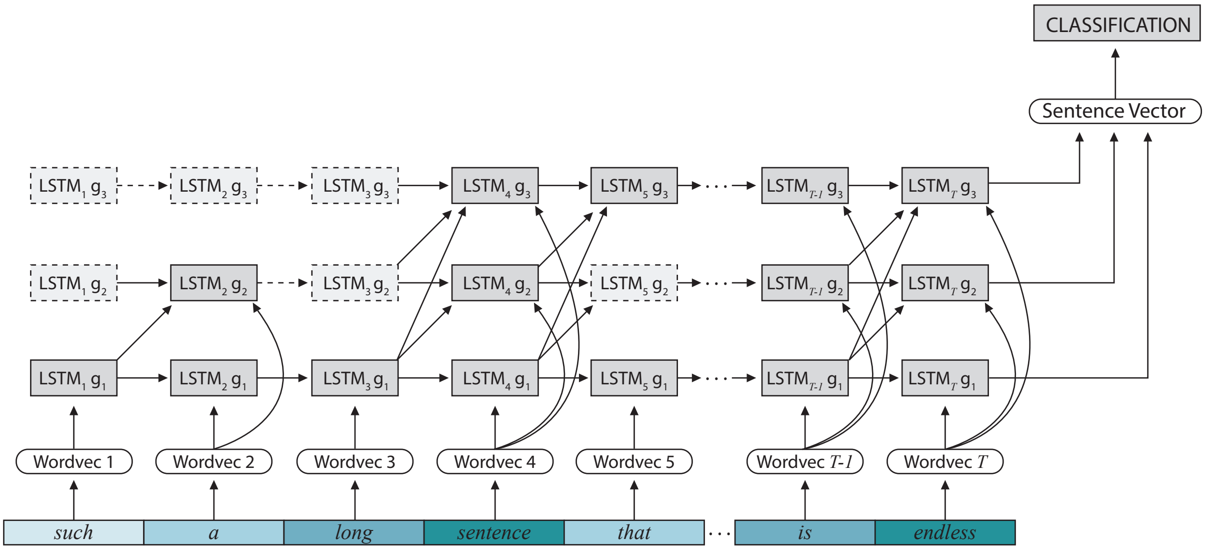


Figure 43 – MT-LSTM graphic model

Source: Produced by the author in February, 2022

4.5.3.3 CNN-based models

According to LeCun et al. (1998), as RRNs are trained to recognize patterns considering time, CNNs learn to recognize patterns across space. While RNNs tend to work better with tasks where comprehension of long-range semantics is needed, CNNs perform better when detecting local and position-invariant patterns is the main goal. These patterns can either a particular feeling about something (like saying/writing “I like”) or a concept or topic inside a sentence (like trying to find if the term “basic healthcare attention” is present on a sentence).

Considering the particular task of text classification, one among pioneers projects stands Kalchbrenner, Grefenstette and Blunsom (2014) **Dynamic CNN**. As stated on section 4.4.2.1, convolutional operations are based on feature maps. On DCNN, these maps are obtained through a process of alternating between word embeddings from a sentence organized into layers with dynamic k-max-pooling layers. These maps are capable of capturing short and long-range relations of words and phrases. The pooling parameter k can be dynamically chosen depending on the sentence size and level of convolution hierarchy.

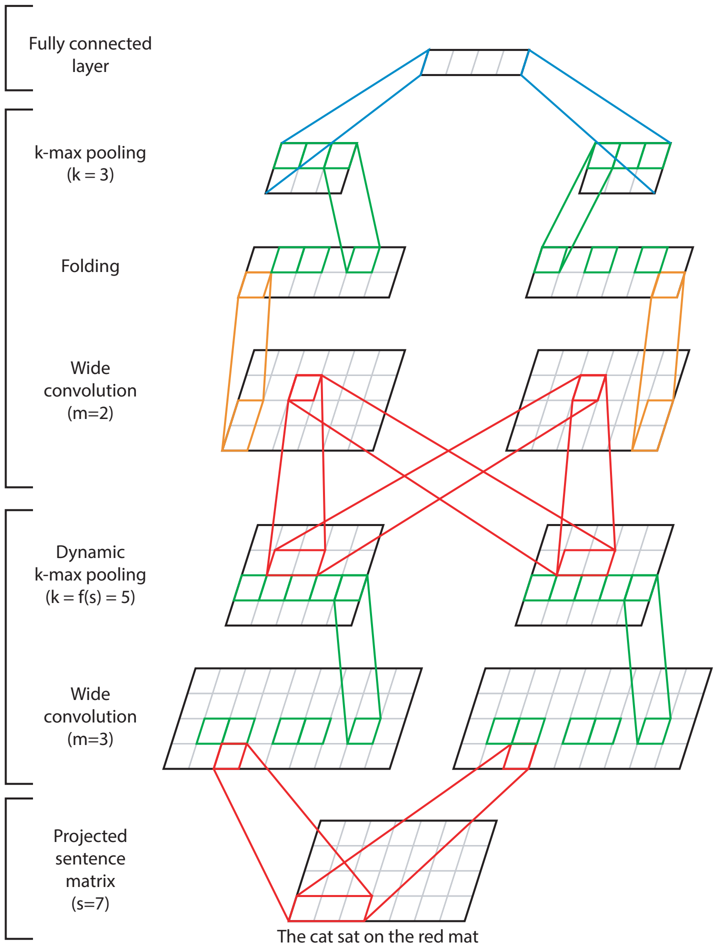


Figure 44 – Kalchbrenner, Grefenstette and Blunsom (2014) DCNN model

Source: Kalchbrenner, Grefenstette and Blunsom (2014)

There is recent interest on investigating performance impact of word embedding on Deep CNN architectures as a counterpoint to the dominance of **LSTM**s architectures and shallow CNNs. Departing from Conneau et al. (2016) whom presented a Very Deep CNN (VDCNN), which operates directly at the character level and uses only small convolutions and pooling operations. The authors claim that *ConvNets* (a short for Convolutional Neural Networks) are largely used and namely adapted for computer vision because of the compositional structure of an image. As texts have similar properties: characters combine to form n-grams, stems, words, phrase, sentences, they believe that a challenge in NLP that could be addressed with VDCNNs is to develop deep architectures which are able to learn hierarchical representations of whole sentences, jointly with the task.

At the other hand, Le, Cerisara and Denis (2018) show that deep models indeed give better performances than shallow networks when the text input is represented as a sequence of characters. However, a simple shallow-and-wide network outperforms deep models such as Huang et al. (2017) DenseNet when dealing with word inputs.

Zhang and Wallace (2015) and Guo et al. (2019) study impact of word embeddings on text classification. As RNNs, **CBOW** and **Continuous Skip-gram** use pre-trained language model as Mikolov et al. (2013a)'s Word2vec and Pennington, Socher and Manning (2014)'s GloVe.

4.5.3.4 Capsule Networks

Considering how CNNs operate through alternate operations of feature extraction, pooling and convolution. Although the technique indeed reduces computational complexity, it is natural to think that some information can be lost during these processes. As stated before by LeCun et al. (1989), CNNs recognize patterns across space, therefore, considering spatial relationship, convolution operations are likely to be mis-classify entities based on their orientation or proportion.

Addressing this issue Hinton, Krizhevsky and Wang (2011) developed a new approach called ***Capsule Networks*** — *CapsNets*. To avoid feature and/or information loss during pooling, groups of neurons are separated into ***capsules*** that perform internal computations aiming to recognize a specific type of entity (an object or part of an object) within a limited domain, and then encapsulated the results into vectors. The length of the resulting vector represents the probability that the entity is present on the domain and the orientation of the vector represents the attributes of the entity. Unlike max-pooling, capsules do not discard information during feature extraction: they are passed through a process of ***routing*** from capsule to capsule, from lower layers to the uppers one:

> If a capsule can learn to output the pose of its visual entity in a vector that is linearly related to the "natural" representations of pose used in computer graphics, there is a simple and highly selective test for whether the visual entities represented by two active capsules, $\boldsymbol{A}$ and $\boldsymbol{B}$, have the right spatial relationship to activate a higher-level capsule, $\boldsymbol{C}$. Suppose that the pose outputs of capsule A are represented by a matrix, $\boldsymbol{T_A}$, that specifies the coordinate transform between the canonical visual entity of $\boldsymbol{A}$ and the actual instantiation of that entity found by capsule $\boldsymbol{A}$. If we multiply $\boldsymbol{T_A}$ by the part-whole coordinate transform $\boldsymbol{T_{AC}}$ that relates the canonical visual entity of $\boldsymbol{A}$ to the canonical visual entity of $\boldsymbol{C}$, we get a prediction for $\boldsymbol{T_C}$. Similarly, we can use $\boldsymbol{T_B}$ and $\boldsymbol{T_{BC}}$ to get another prediction. If these predictions are a good match, the instantiations found by capsules $\boldsymbol{A}$ and $\boldsymbol{B}$ are in the right spatial relationship to activate capsule $\boldsymbol{C}$ and the average of the predictions tells us how the larger visual entity represented by $\boldsymbol{C}$ is transformed relative to the canonical visual entity of $\boldsymbol{C}$. If, for example, $\boldsymbol{A}$ represents a mouth and $\boldsymbol{B}$ represents a nose, they can each make a prediction for the pose of the face. If these predictions agree, the mouth and nose must be in the right spatial relationship to form a face. (HINTON; KRIZHEVSKY; WANG, 2011, p.2)

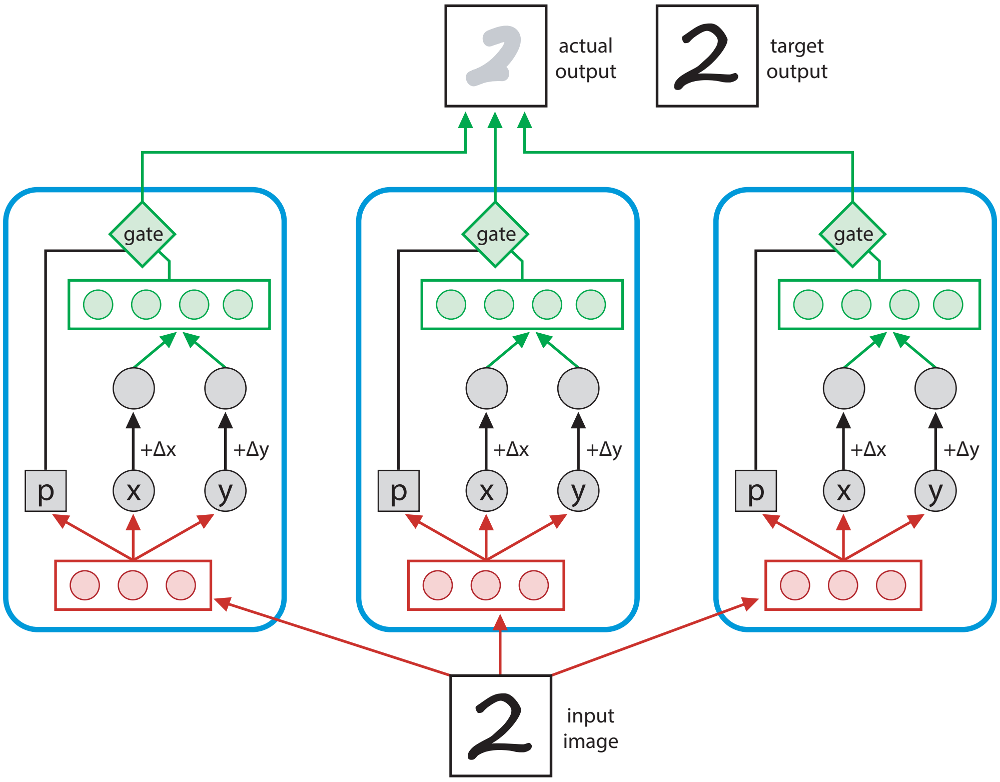


Figure 45 – Hinton, Krizhevsky and Wang (2011) CapsNet model. The network is composed of three capsules that interacts only at the final layer, cooperating to produce the desired shifted image. Each capsule is composed by three recognition units and 4 generation units.

Source: Hinton, Krizhevsky and Wang (2011)

Figure 45 presents the initial model proposed by the authors. The task in frame is a transforming auto-encoder that models translations. The network is deterministic (always present the same result given certain conditions) and once learning is achieved, it takes as inputs an image and the desired shifts Δx and Δy to output the desired shifted image. Each capsule has a hidden layer of recognition units that outputs three numbers: x, y and p that the will be sent to higher levels of the network. p is the probability that that x and y will be present in the input image.

For text classification tasks, the most common routing procedure is dynamic (ZHAO et al., 2018; REN; LU, 2018; YANG et al., 2019; ZHAO et al., 2019; ALY; REMUS; BIEMANN, 2019). Recently, Kim et al. (2020) proposed a CapsNet-model with static routing procedure for text classification. The authors observe that objects can be more freely assembled in texts than in images. For example, a document semantics can remain the same even if the order of some sentences is changed, unlike the positions of the eyes and nose on a human face. Thus, they use a static routing schema, which consistently outperforms dynamic routing.

4.5.3.5 Attention mechanisms

Dealing with a multitude of objects and their properties on multimodal environments is, by nature, a complex and voluminous task. As human beings select which real-world manifestation are more important then others, ANN should also do. Models with attention mechanisms intend to make a stand towards this kind of problem. On NLP, one of the first models is Bahdanau, Cho and Bengio (2014), which deals with text translations of long sentences. In brief, the authors verify that fixing sentence representation vectors to a certain length is overcome by allowing the model to automatically search for relevant parts of the sentence, regardless of the distance between them, and then predicting a more suitable result.

According to Minaee et al. (2021), attention in language models can be interpreted as a vector of importance weights. In order to predict a word in a sentence, we estimate using the attention vector how strongly it is correlated with, or “attends to”, other words and take the sum of their values weighted by the attention vector as the approximation of the target (MINAEE et al., 2021, p.10).

Wang et al. (2018) develop a Label-Embedding Attentive Model to improve text classification. The authors follows the idea proposed by Shen et al. (2018) that word embedding presents better results on text classification tasks, and classify the technique as a fundamental building block for neural-based NLP due to their capacity of capturing semantic and syntactic regularities between words using vector arithmetic, cited by Mikolov et al. (2013b) and Pennington, Socher and Manning (2014). This procedure has been extended to compute embeddings that capture the semantics of word sequences (e.g., phrases, sentences, paragraphs and documents). The main idea resides in jointly embedding the word and label in the same latent space, and the text representations are constructed directly using the text-label compatibility through cosine similarity.

On figure 46(a) represents the traditional pipeline for text classification. Analyzing the image, the use of label information occurs only at the last step, while learning $\boldsymbol{f_2}$. All impacts from this knowledge on learning the representation $\boldsymbol{f_0}$ or word sequence $\boldsymbol{f_1}$ are either ignored or presents an indirect effect.

On figure 46(b) not only words $\mathbf{v}$ are embedded on the space $\boldsymbol{f_0}$ but also labels $\mathbf{C}$ that act like drivers to identify which classes influence the refinement of word embeddings. In the figure, there are two potential classes on $\mathbf{C}$. Compatibility between words and labels are leveraged through the operator $\otimes$ resulting on vector $\mathbf{G}$ that derive the attention score β, which improves word embedding $\mathbf{z}$.

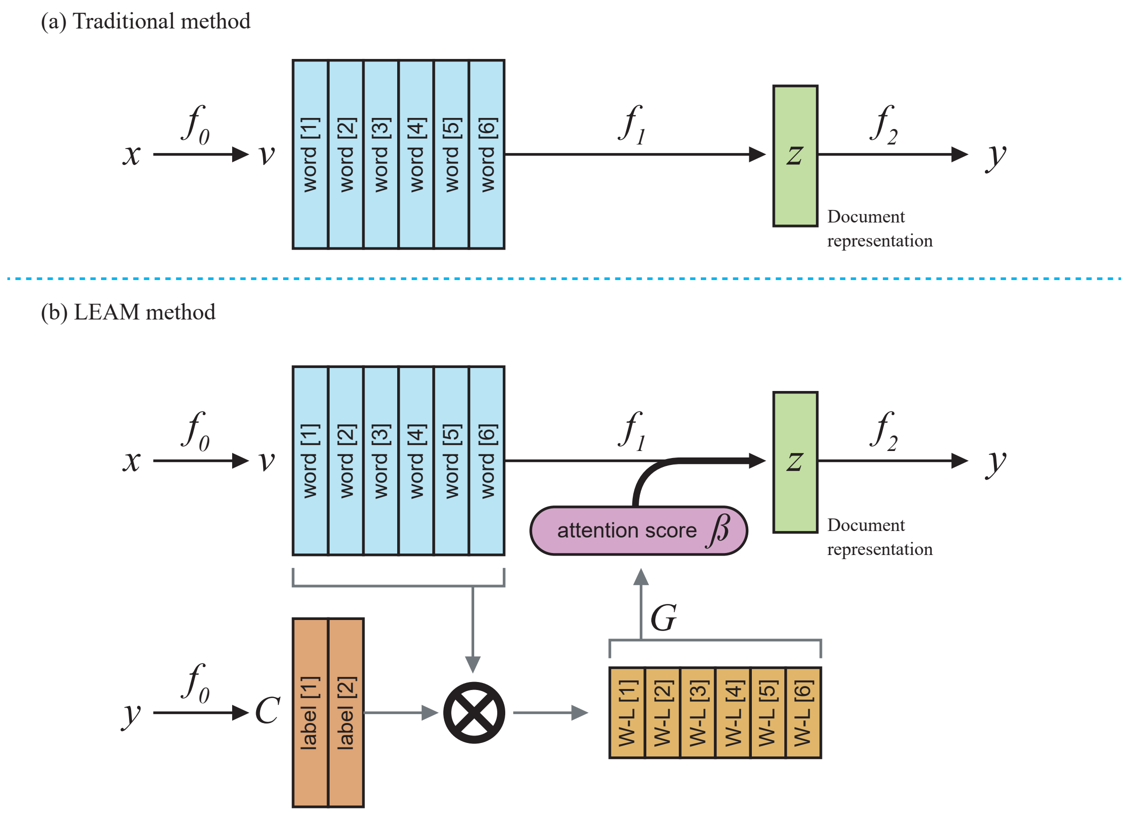


Figure 46 – Wang et al. (2018) LEAM model.

Source: Adapted from Wang et al. (2018)

4.5.3.6 Memory-augmented networks

While Attention mechanisms implement an *internal memory* of the network in which vectors are hidden entries, Memory-Augmented techniques combine neural networks with an *external memory*, which the model can read from and write to.

For text classification methods, Munkhdalai and Yu (2017) developed a model called Neural Semantic Encoder — NSE. The network is equipped with a variable sized encoding memory that evolves over time and maintains the understanding of input sequences through read, compose and write operations. It can also access multiple and shared memories. Figure 47 presents a simplified representation. NSE performs three main operations in every time step. After initializing the memory slots with the corresponding input representations, NSE processes an embedding vector x_t and retrieves a memory slot $m_{r,t}$ that is expected to be semantically associated with the current input word w_t. The compose module implements a composition operation that combines the memory slot with the current input. The write module then transforms the composition output to the encoding memory space and writes the resulting new representation into the slot location of the memory.

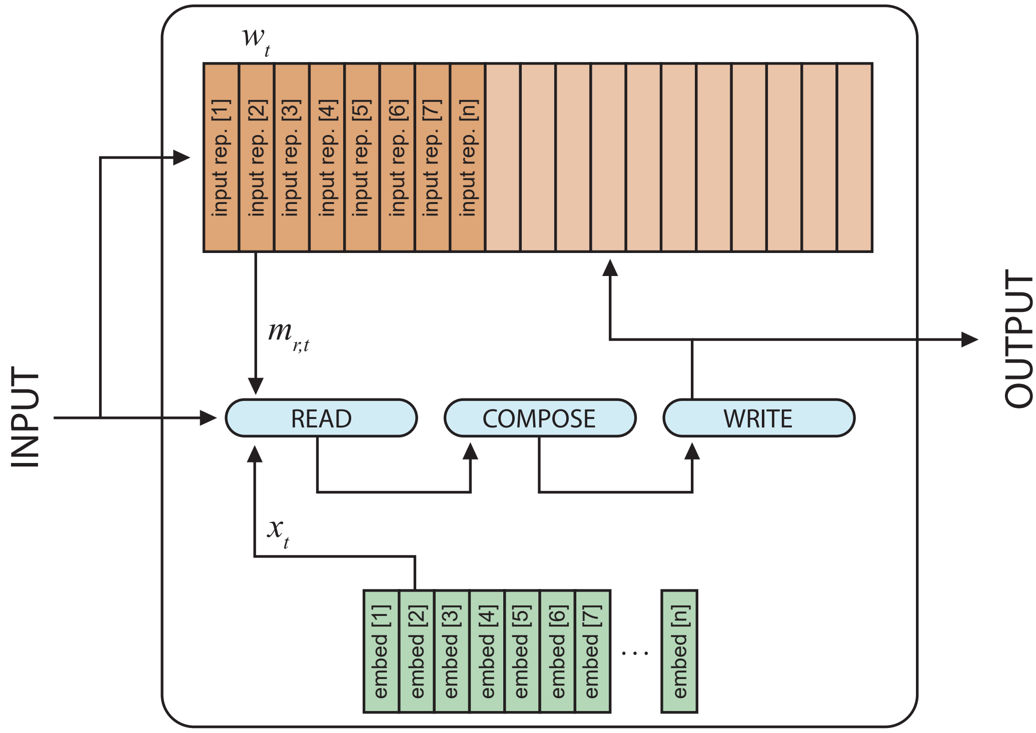


Figure 47 – Munkhdalai and Yu (2017) NSE model.

Source: Adapted from Munkhdalai and Yu (2017)

4.5.3.7 Tranformers and Pre-Trained Language Models — PTMs

A major problem faced by both CNNs and RNNs in text classification tasks is capturing relationship between words in a sentence. Specially as this complexity grows with the increasing length of the sentence. Until 2017, NLP was dominated by CNNs, RNNs or **LSTM**s. As Attention mechanisms presented as the best alternative to connect encode and decode procedures with an attention mechanism, Vaswani et al. (2017) propose a new architecture called ***Transformer***, solely based on attention mechanisms. The technique introduced two major innovations:

1. **Self-attention easing computing:** for every word in a sentence or document, an “attention score” is given, in order to attribute a certain value of influence of one word on another;
2. **Improved parallelization methods:** since Kaiser and Sutskever (2015) introduced the concept of **Neural GPUs**, sequential depth limitation imposed by traditional networks has been overcame, due to the capacity of graphic processors to implement parallel computations. The authors show that Neural GPUs can be trained on short instances of an algorithmic task and successfully generalized to long distances. This feature reduces training time and improve model quality.

Since 2018 there is an increase of implementations of large-scale **Pre-trained Language Models** — PTMs — based on Transformers. These models have deeper architectures then contextualized embedding models based on **CNNs** or **LSTMs**, and are *pre-trained* on much larger amount of text corpora, which leads to better contextualization of words and sentences. Basic training for PTMs is based on Unsupervised learning, as it initially aims to acquire knowledge about the language representation. *Fine-tuning* is done through Supervised learning using task-specific labels.

Recently, Qiu et al. (2020) categorize the most popular PTMs based on a taxonomy from four different perspectives: representation types, model architecture, type of pre-training task and extensions for specific types of scenario.

i. **Representation type**: language representation aims to capture implicit linguistic rules and common sense knowledge hidden in text data such as lexical meanings, syntactic structures, semantic roles, and even pragmatics. Word embeddings can either be non-contextual or contextual.

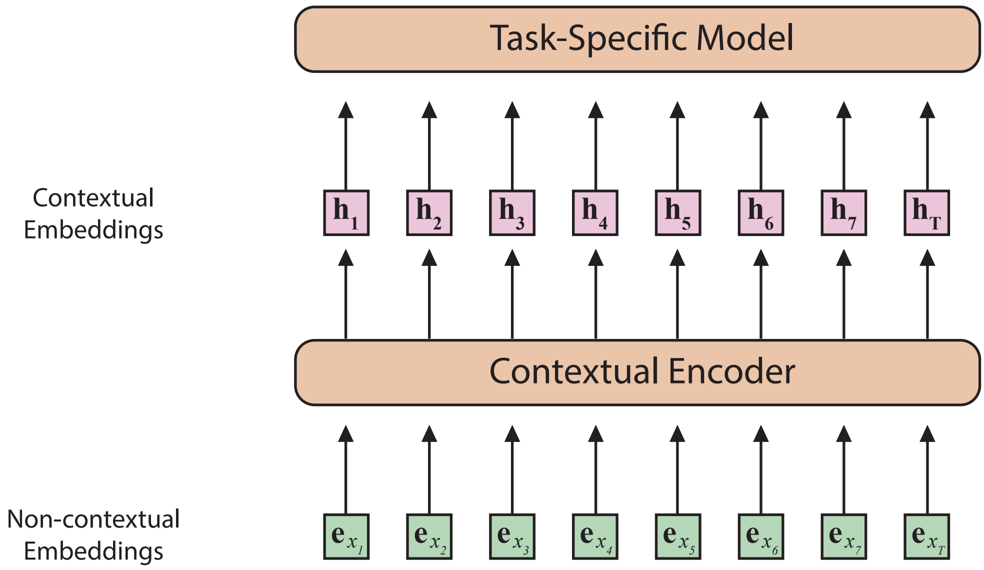


Figure 48 – Qiu et al. (2020) Generic Neural Architecture for NLP.

Source: Adapted from Qiu et al. (2020)

- **Non-contextual embeddings** is the first step on language representation mapping. In brief, the procedure maps each word x in a vocabulary $\mathcal{V}$ to a vector $\mathbf{e}_x \in \mathbb{R}^{D_e}$ with a lookup table $\mathbf{E} \in \mathbb{R}^{D_e \times |\mathbb{V}|}$, where D_e is the dimension of tokens embeddings. The result leads to a static model, unable to deal with polysemous words, with vocabulary-limited range of action, that is, only mapped words will be identified.

- **Contextual embeddings** address the issues of polysemous and context-dependent words. Through a neural encoder $f_{enc}(\cdot)$, the contextual representation h_t of a token x_t depends on the whole text (or sequence of words) $[x_1, x_2, x_3, \cdots, x_T]$, where

$$[h_1, h_2, h_3, \cdots, h_T] = f_{enc}(x_1, x_2, x_3, \cdots, x_T) \tag{4.57}$$

ii. **Model architecture**: analyzing the definition of an encoder, its architecture can directly affect effectiveness of the model. Most neural context encoders can be classified into sequence and non-sequence models.

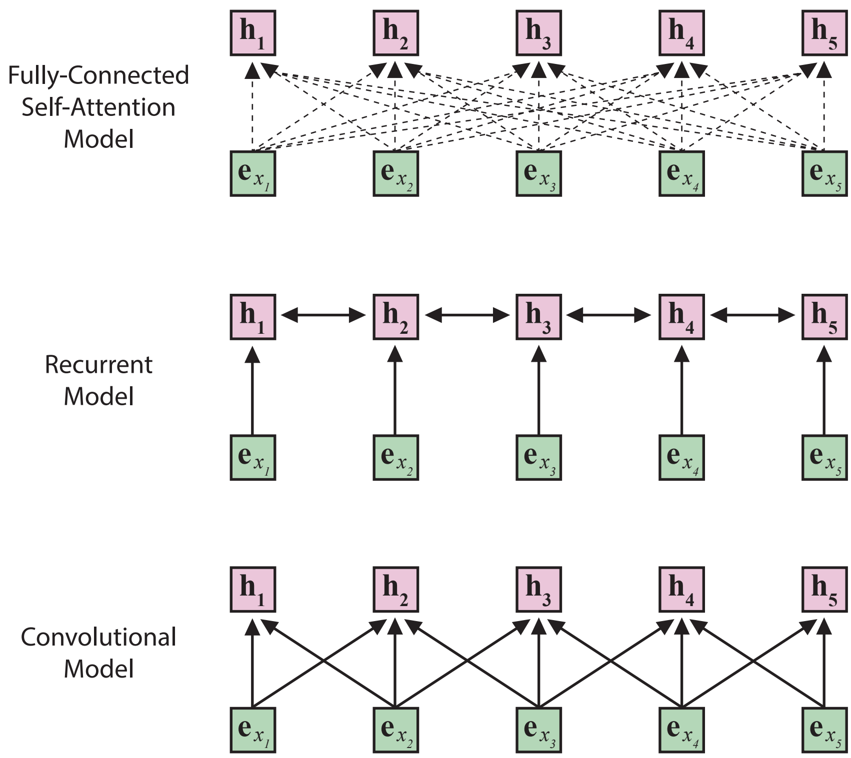


Figure 49 – Qiu et al. (2020) examples for Neural Contextual Encoders architectural models.

Source: Adapted from Qiu et al. (2020)

- **Sequence Models:** usually capture local context of a word in a sequential order. CNN-based models take embeddings of words in the input sentence and use convolution operation to extract the meaning of a word through analyzing its neighbors. RNN-based models capture contextual representation of words with short term memory like LSTM. Both methods are easy to train, but fail to capture long-range interactions between words.
- **Non-Sequence Models:** learn the contextual representation with a pre-defined tree or graph structure between words, such as the syntactic struc-

ture or semantic relation. In practice, a more direct way to obtain these structures is through a Fully-connected Self-Attention Model, which would predetermine every two word relation, letting the model learn the structure by itself. This characteristic makes it a powerful long-range dependencies identifier but requires a large training corpus and is easy to overfit on small data-sets.

iii. **Pre-training task:** as a strategy to avoid the challenge of building large-scale labeled data sets for NLP, specially mapping syntax and semantics, pre-training models works with unlabeled corpora which are relatively easier to construct, leveraging the huge amount of text corpus to learn universal language representations that facilitate other specific tasks. Model initialization becomes easier with pre-training (since common language relationship are already mapped) and it also can be viewed as a regularization tool to avoid overfitting when dealing with small data.

Historically, it is possible to divide PTMs into two generations of development. The ***first-generation PTMs*** aim to learn good word embedddings. Therefore, these models acquire a pairwise ranking of words instead of language modeling, being context-independent vectors. Mikolov et al. (2013a)'s **Word2vec**, Pennington, Socher and Manning (2014)'s **GloVe** and Mikolov et al. (2013b)'s CBOW and Continuous Skip-Gram are examples.

As the majority of NLP tasks are beyond word-level and suffer great influence from the context they are inserted, ***second-generation PTMs*** aim to produce word vectors on a sentence-level or higher, therefore called *contextual word embeddings* since they represent word semantics depending on its context. McCann et al. (2017)'s **CoVe**, Peters et al. (2018)'s **ELMo**, Radford et al. (2018)'s **OpenAI GPT** and Devlin et al. (2018)'s **BERT** are examples of these PTMs.

How the PTM is trained makes great difference on learning universal representation of language. In summary, Qiu et al. (2020) divide all tasks into three categories:

- **Supervised learning (SL):** aims to learn a function that maps an input to an output based on training data consisting of input-output pairs.
- **Unsupervised learning (UL):** aims to find some intrisic knowledge from unlabeled data, such as clusters, densities, latent representations.
- **Self-supervised learning (SSL):** is a blend of of supervised and unsupervised learning. The procedure of learning is the same as in supervised learning, but the labels of training data are generated automatically. The main objective of SSL is to predict any part of the input from other parts of the same input.

iv. **Extensions:** PTMs usually learn universal language representations for general-purpose applications. Data assembled for basic training is composed of a vast variety of contexts: legal, technological, romance, fiction and many others. Therefore, to execute properly (and with higher degree of assertiveness) some specific tasks, model enrichment is not only desirable, but needed.

- **Knowledge-Enriched PTMs:** specific knowledge, like linguistics, semantic, commonsense, factual or domain-specific, can be inserted into PTMs both during or after pre-training. BERT appears as the main base used for enrichment nowadays.
- **Multilingual PTMs:** can either be multilingual or language specific. For multiple languages, PTMs can work at cross-lingual language understanding, acting as translators from different idioms; and cross-lingual language generation, to generate text in different idioms from one input. BERT appears as the main base used for multilingual PTMs nowadays.
- **Multimodal PTMs:** due to the growing success of PTMs on NLP tasks, some researches focused on obtaining a cross-modal version of PTMs. Essentially, majority of these models are designed for a general visual and linguistic feature encoding. And these models are pre-trained on some huge corpus of cross-modal data, such as videos with spoken words or images with captions, incorporating extended pre-training tasks to fully utilize the multi-modal feature. BERT appears as the main base used for multimodal PTMs nowadays.
- **Domain-specific and Task-specific PTMs:** most publicly available PTMs are trained on general domain corpora such as Wikipedia, which limits their applications to specific domains or tasks. Recently, some studies have proposed PTMs trained on specialty corpora, like biomedical texts, scientific texts, clinical texts and sentiment analysis.

4.5.3.8 Named entity recognition

A long lasting objective on NLP is to develop techniques for understanding textual messages, that is, obtaining the semantic value within a sequence of words. Grishman and Sundheim (1996) describe that since 1987 US Military promotes conferences in order to asses and foster researches on automated message analysis of military content. These conferences were called MUC — *Message Understanding Conferences*. In 1993, one of the main goals was to promote *deep understanding*, as countermeasure to the tendency towards relatively shallow understanding techniques which were primarily based on local pattern matching. Three tasks would represent what was called *Semantic Evaluation*:

i. **Coreference**: the system would have to mark coreferential noun phrases (the initial specification envisioned marking set-subset and part-whole relations, in addition to identity relations);

ii. **Word sense disambiguation**: for each open class word (noun, verb, adjective, adverb) in the text, the system would have to determine its sense using the Wordnet[5] classification (its "synset", in Wordnet terminology);

iii. **Predicate-argument structure**: the system would have to create a tree interrelating the constituents of the sentence, using some set of grammatical functional relations.

These practices were distributed into 4 categories: coreference, template element, scenario element and named entity. For Grishman and Sundheim (1996), the name entity task involves identifying the names of all the people, organizations and geographic locations in a task. This idea of named entity has evolved from handcrafted rules, lexicons and ontologies to feature-engineering and machine learning. Yadav and Bethard (2019) presented a review on Named Entity Recognition — NER — systems, dividing them into four major groups.

1. **Knowledge-based systems:** do not require annotated training data as they rely on lexicon resources and domain specific knowledge. These work well when the lexicon is exhaustive making precision generally high for knowledge-based NER systems because of the lexicons, but recall is often low due to domain and language-specific rules and incomplete dictionaries. Another drawback of knowledge based NER systems is the need of domain experts for constructing and maintaining the knowledge resources;

2. **Unsupervised and bootstrapped systems:** systems which require training data in order to extract named entities. The data in question does not contains labels of expected outputs, but can include some features like orthography, context of entities, words contained within named entities, gazetteers, person, organizations among others. Techniques like ***Inverse Document Frequency (IDF)*** combined with shallow syntactic knowledge can be used to obtain potential named entities. Papineni (2001) explains that IDF is a popular measure of a word's importance, being defined by Jones (1973) as the logarithm of the ratio of number of documents in a collection to the number of documents containing the given word. Common words presents low IDF values (empirically meaning low relevance) in contrast to high IDF values associated with rare words (therefore, empirically meaning high-relevance).

[5] https://wordnet.princeton.edu/

3. **Feature-engineered supervised systems:** learn to make predictions by training on example inputs and their expected outputs, and can be used to replace human curated rules.

4.6 Multimodal Information Architecture: Kuroki Jr. (2018) proposal

Objective reality is Multimodal. Our experience of it is based on multiple modes. It is not conceivable to separate, atomically, all stimuli that takes place on meaning construction. It would be difficult to understand a mode called language: modes writing and speaking seems more accurate. Nonetheless it would be awkward to ask a normal person (non-color-blind) to see only the form of a bird, ignoring the colorfulness on the experience. Kress and Van Leeuwen (2001) addressed the issue on their book Multimodal Discourse. Aiming to assemble guidelines for writing in musical, imagery or signal language, they realize that a meta-theory for multimedia (as several technological implementations) based on communicative practice would be necessary. The authors recognize that any semiotic grammatical regulation (as governance for the use of signs) will always be tested by the repository of circumstantial associations that is the human knowledge. They conclude that no form of communication is privileged: when giving meaning to a context, all stimuli placed at the disposal of the interpreter can be used.

At the other hand, Wilson and Sperber (2002) proposed Relevance Theory, which states that utterances raise expectations of relevance not because speakers are expected to obey a Cooperative Principle: the search for relevance is a basic feature of human cognition. What makes it possible for the hearer to recognize the speaker informative intention is that utterances encode logical forms (conceptual representations, however fragmentary or incomplete) which the speaker has manifestly chosen to provide as input to the hearer's inferential comprehension process. As a result, verbal communication can achieve a degree of explicitness not available in non-verbal communication

Assuming the coexistence of Multimodal reality and Relevance Theory, as well as setting ways towards the construction of an answer to the initial question: is it possible to determine the ideal amount "order" that an informational environment can "absorb"?

4.6.1 Epistemological foundations of Multimodal Information Architecture

Information Architecture has been treated as a discipline that has foundations on the Internet explosion. Several authors use Rosenfeld and Morville (2006) definition, which addresses methods for web sites mapping and designing. This technicist view assigns a marginal role to information organization. On a slightly different path, Resmini and Rosati (2012) define a new concept called *Pervasive Information Architecture*, where information is distributed through cross-channel means. These means are still bounded to technological implementations: the same information needs to be distributed through mobile applications, printed versions and physical spaces as well.

The sense of order that MIA aims passes through all these technological implementations, but with more fundamental objective: is there a more rational way to manage how meaning and knowledge are developed even when reality is composed of several modes of signification? To achieve this goal, MIA needs to address meaning constructing and modelling, not only technological implementations.

Therefore, Kuroki Jr. (2018) assumes the three-level methodological concourse proposed by Van Gigch and Moigne (1989) as world-view for MIA. For his work, only epistemological and theoretical levels where addressed and, among indications of future works, a direct reference to Deep Learning applications was made.

MIA's definition is conceived through the epistemological conjunction of two terms: information and architecture. Afterwards, the result is applied to multimodal realities.

4.6.1.1 A review on the definition of Architecture

Pollio (1960), cited by some authors as the *Father of Architecture*, initiates his discussions about the definition of the activities performed by an architect, stating that in all matters, but particularly in architecture, there are two points: what is signified, and that which gives it its significance. The author proposes six pillars for producing an architectural design: *Order, Arrangement, Eurythmy, Symmetry, Propriety* and *Economy*.

Order gives due measure to the parts of a work considered separately, and symmetrical agreement to the proportions of the whole. Arrangement includes putting things in their proper places and the elegance of effect; *Eurythmy* is beauty and fitness in the adjustments of parts and Symmetry is an agreement between the members of the work. It is possible to presume all of them on Order. For this work we will consider this agglutination.

Propriety is that perfection of style which comes when a work is constructed on approved principles. It arises from prescription (the solution denotes clear link to the purpose that gave rise to it), from usage (historically consolidated standards) or from nature (natural conditions restrictions). The definition denotes functional, cultural and environmental constraints imposed on the object, are external to it and refer to a context of construction of the architectural project.

Economy also denotes restrictions, however, about means of production as well as limitations on expanding the object. The use of appropriate materials for each situation imposed by Propriety restrictions, with rational use of resources and physical space available for construction.

Philosophically, to Abbagnano (2015), *Order* is defined as any relation between two or more objects expressed by a rule. In some sense, the author makes connection between this definition and *Economy*, for which he states as being the Order or regularity of any social totality, from a house to all human existence and quotes that *William of Ockham* was the first to express a principle of Economy through the expressions *entities should not be multiplied without necessity* and *in vain accomplished by several instruments when fewer where demanded*.

Both ways, Architecture can be related to the construction of rules which govern possible relations between objects, subjects and context.

4.6.1.2 A review on the definition of Information

Defining the object that an Architecture impose a sense of Order is critical when constructing the concept of MIA. Notoriously polysemic is the term Information. From the need of instructions for a context to ideas or thoughts of a being. The search for a consensual definition is too bold of a task. Since Floridi (2004) defined seventeen open problems on the new discipline of Philosophy of Information, two of them seems to take special part on Information Science:

[P. 1.] The elementary problem: What is Information?

[P. 3.] The UTI challenge: Is a grand unified theory of Information possible?

For the latter, Floridi himself seems to discard the possibility, stating that reductionist strategies are unlikely to succeed. Several surveys have shown no consensus or even convergence on a single, unified definition of Information.

Later, Floridi (2008) presents the convergence on admitting a General Definition of Information (GDI) as a semantic content in terms of data + meaning. GDI has become an operational standard especially in fields that treat data and Information as reified entities (as expressed on “data mining” and “information management”). Examples include Information Science and Information (Systems) Management. Recently, GDI has begun to influence the philosophy of computing and information.

Brier (2015) presents a transdisciplinary concept of Information, which the core should not be based on pure logical or mathematical rationality. It adds interpretation, signification and meaning construction while Information is a basic aspect of reality alongside physical, chemical and molecular biological. It discusses not an “objective” definition but a relativized one in relation to both the sender’s and the receiver’s knowledge. He proposes a *Cybersemiotic* view of Information, combining the cybernetic perspective of information based on *Gregory Bateson’s* work (the difference that makes the difference) with the Semiotic vision of *Charles Peirce*, founded on phenomenology and pure mathematics stating that Information bits are at most pre- or quasi-signs, and, insofar as they are involved with codes, they function only like “keys in a lock”. Information bits in a computer do not depend for their functioning on living systems with final causation to interpret them. They function simply based on formal causation, as interactions depending on differences and patterns. But, when people see Information bits as encoding for language in a word-processing program, then the bits become signs for them. Following in the footsteps of Peirce, whose Semiotics allows us theoretically to distinguish between the Information the sender intended to put in the sign, the (possible) Information in the sign itself and the Information the interpreter gets out of the sign, instead of the idea that it is the same in all three.

What makes distinction between Floridi and Brier is that the latter does not restrict Information as being a product of interpretation of an object: it goes deeper. **Information** is an entity that enables the phenomenon of signification to some cognitive subject, what makes real sense when we analyze this statement on a multimodal perspective as Kress and Van Leeuwen (2001) proposed.

4.6.1.3 A review on Modal Logic

According to Abbagnano (2015), Logic can be defined as a discipline that privileges coherence in a set of statements, which is, if there is any possible situation that makes true all statements of the set. What makes this task particularly complex is the Multimodal nature of reality and the *Cybersemiotic* view of Information. Any stimuli can easily be relevant for a subject (a key for his/her lock) and irrelevant for another.

Modal Logic studies the possible ways of qualifying truths. These "Modalities" of qualification are an axiomatic or linguistic extension of Classical Logic. In this sense, classical connectives have the same meaning in Modal Logic. Notions as possibility (symbolized by a diamond) and necessity (symbolized by a square), therefore, will obey rules and thesis from classical propositional calculus. These ways of qualifying truth come along with two notions very useful for our purpose: **Possible worlds** and **Accessibility Relations**.

Suppose that a set of objects and a definition attributed to each object is presented to a group of three people. Everyone asserts true or false for if he/she agrees with the definition assigned to each object presented and write it down on paper. The three pieces of paper produced are now possible worlds in our model. Not necessarily one world is equal to another, in fact, is very likely that, considering the values asserted, we now have three totally different worlds.

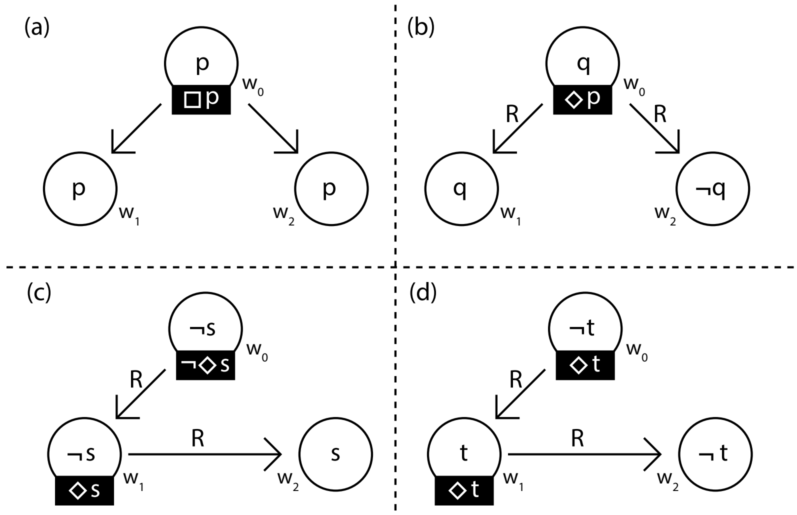


Figure 50 – Accessibility Relations graphic example

Source: Kuroki Jr. (2018)

This situation would be more likely to achieve (three totally different worlds) if we could separate all the individuals so the responses wouldn't be contaminated. In general, people talk to each other (in our example, even "sneak" at someone's answers) before doing things. For that, Modal Logic presents the notion of Accessibility Relation: which worlds can be accessed from one particular world? Through these Accessibility Relations modal notions of necessity and possibility are built. Carnielli and Pizzi (2008) describe **necessity** (represented by the symbol □) as in Rudolf Carnap's theoretical model, which states that necessary propositions are those which are true at all possible worlds, while **possibility** (represented by the symbol ◇) states that in some world the proposition is truth.. Bringing to our example, is the same to say that all three subjects assigned that a certain definition matches the object it is related to. But how does Relations come in to discussion? Analyze another example, with the same three people and the same situation, now in graphical representation on figure 50.

The figure presents four situations where three subjects are represented by their assumptions in w_0, w_1 and w_2 of the objects p, q, s and t. Now, the individuals have access to each other convictions (if the object is indeed related to the concept presented) through the relation R. This changes several things in our representation. Situation (a) shows the possibility where the individual w_0 has access to both other people. Since he or she verifies that p is true to everyone, he or she can assert that necessarily p is true. At the other hand, at situation (b), even though w_0 asserted that q is not true, he or she admits that possibly q is true, because there is a world that makes q true. The model shows us through arrows who can access who by the relation R enriching the model with necessities and possibilities.

Logic is expressed through axioms and propositions. A direct way to explain what an axiom represent is thinking of something so obvious that cannot be negated. Propositions are mathematical ways of expressing any kind of statements. In a simple way of definition, it would be like a mathematical variable. As an example, the proposition [p] could be taken as the "*color of this bird is red*" or "*it sounds like a pigeon*". An axiom can be exemplified as [if p then p], which states for Identity. What modal logic do is to enrich these axiomatic systems with some connectors, generating Logical Modalities. *Knowledge*, *Belief*, *Deontic* (in a sense of morality), *Dynamicity* (in a sense of process execution), *Time*, all of them are Modalities. Carnielli and Pizzi (2008) presented some practical examples of Modalities. For our purpose, a reduced adaptation is showed in figure 51.

Connector	Modality	Syntax
O_i	Deôntic	On world i, is obligatory that
P_i	Deôntic	On world i, is permitted that
F_i	Deôntic	On world i, is forbidden that
$[a]$	Dinamic	Execute process $[a]$
$\boxed{P}$	Temporal	Always have been the case that
$\boxed{F}$	Temporal	Always will be the case that
$\langle P \rangle$	Temporal	It was the case that
$\langle F \rangle$	Temporal	It will be the case that
K_i	Epistemic (Knowledge)	Subject i knows that
B_i	Doxastic (Belief)	Subject i believes that

Axiom	Sintax
$K_i p \supset B_i p$	Subject i knows that p, then, i believes that p.
$O_i p \supset O_j p$	In world i is obligatory that p, then, in world j is also
$\langle P \rangle p \supset \boxed{P} \Diamond p$	It was the case that p, then, always was the case that possibly p

Figure 51 – Modal Logic modalities examples

Source: Kuroki Jr. (2018)

An application of how Modal Logic can enrich Classical Logic is adding the Deontic notion of Obligation [O_i], stating that "*on non-color-blind world is obligatory that if the color of this bird is red, then the color of this bird is red*" by writing [O_i [if p then p]].

Allied to Axioms and Modalities, Portner (2009) describe the notion of Frames, which are the structure of connection between worlds and Relations. In a practical way, Frames are logical ruling that restricts the Relations in a model. Carnielli and Pizzi (2008) describe some Frames which are synthetized in figure 52 below.

Frame	Syntax
Serial	In (w_1, w_2, w_3), w_1 reaches w_2 and w_2 reaches w_3.
Reflexive	In (w_1, w_2, w_3), each world reaches with itself.
Transitive	In (w_1, w_2, w_3), if w_1 reaches w_2 and w_2 reaches w_3, then w_1 reaches w_3
Symmetric	In (w_1, w_2), if w_1 reaches w_2 then w_2 reaches w_1
Euclidean	In (w_1, w_2, w_3), if w_1 reaches w_2 and w_1 reaches w_3, then w_2 reaches w_3

Figure 52 – Frames and their sintax

Source: Kuroki Jr. (2018)

4.6.2 Constructing MIA: adequations and properties

The epistemological base formulated indicated that *Order, Rule, Relation, Worlds* and *Economy* are key concepts to the idea of Architecture. Modal logic brings some syntactic plasticity when formalizing concepts in technological implementations. At the other hand, Information seems to be a problem with no clear solution with considerably amount of theories and technological uses. As the objective intended was not the definition of these concepts but to construct a definition of MIA, for the scientific level of analysis, the assumption of some adequations of terms were proposed (and, sometimes, premises so that these adequations can be understood) in order to ground the development of properties of the concepts of Architecture and Information.

4.6.2.1 Architeture: adequations and properties

For Kuroki Jr. (2018), an architecture must deal with *Rules* and *Relations* to achieve *Order* considering an *Economic* way of dealing with it. Four adequations were proposed:

[ADQ.1] – *Relation* is any form of connection between instances within a world or worlds among each other;

[ADQ.2] – *Rule* is a relational context which restricts the possible Relations of instances within a world or worlds among each other;

[ADQ.3] – *Economy* is a dynamic grouping of worlds that an instance within a world or a world itself requires so that a Rule or Relation be enabled;

[ADQ.4] – *World* is a Mode, as in Kress and Van Leeuwen (2001), which enables meaning to be expressed.

All four adequations and their relations with Architecture can be represented through figure 53, showing also the nature of each relationship.

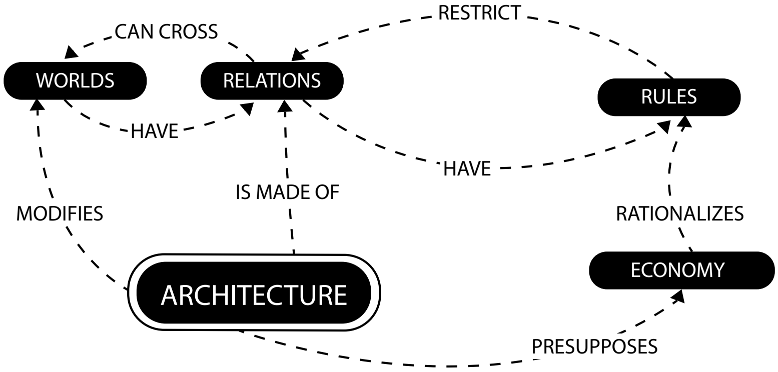


Figure 53 – Concepts related to Architecture

Source: Adapted from Kuroki Jr. (2018)

In order to make a practical example of these adequations, the author describes a simple context of geometrical figures. As reduced and simplistic the model appear to be, several Modes (as in Kress (2009), a socially shaped and culturally given resource for making meaning.) can be listed even before visualizing the example itself: form, size, color, direction among others. Figure 54 reproduces the given situation.

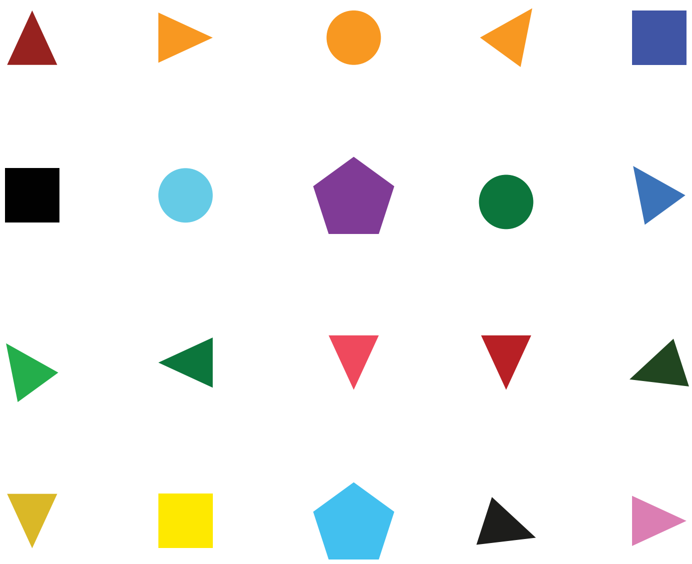


Figure 54 – Multimodal model of a simple reality

Source: Adapted from Kuroki Jr. (2018)

First established task was to identify possible worlds. Even though the model presented seems simple, every characteristic of every object could be a possible world: form, color, shape, volume or any other. The same goes for Rules and Relations. For these questions, three premises are now presented.

[PRM.1] – ***Possible world*** is any distinction of instances of a model, taken individually or by group;

[PRM.2] – ***Applied Relation*** is any structure of analysis of instances of a model, based on a possible world;

[PRM.3] – ***Applied Rule*** is any form of restriction of Applied Relations.

Considering that the distinction *shape* would be the dominant Mode for meaning construction and applying all premises and adequations proposed, it would be feasible to distinguish four possible worlds as presented in figure 55 (world of triangles, circles, squares and pentagons), from which is conceived the first property for Architecture.

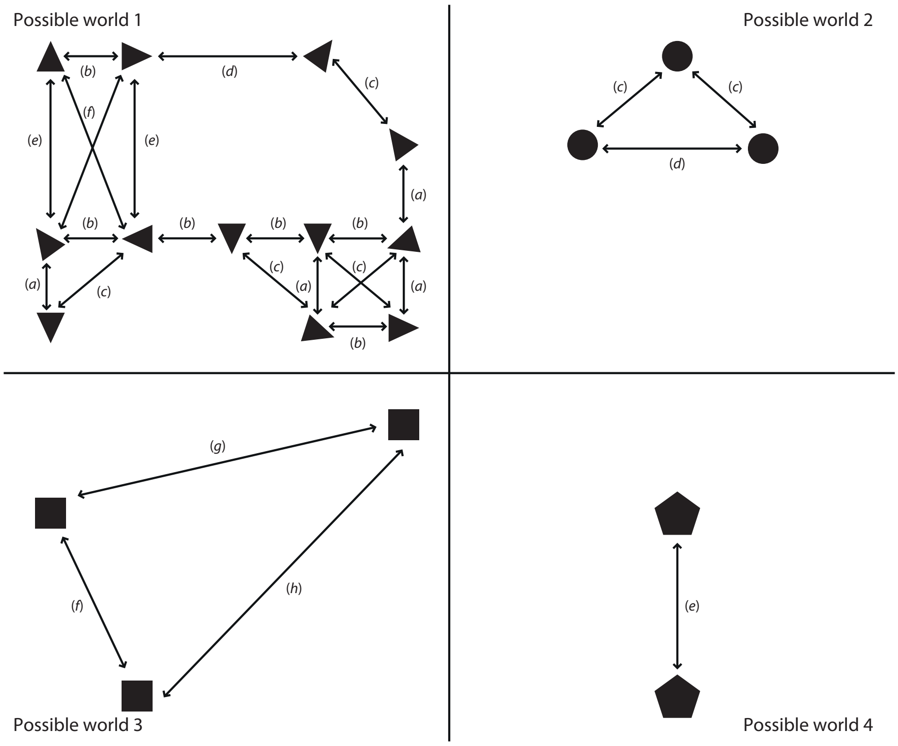


Figure 55 – Distinguished model of figure 54

Source: Adapted from Kuroki Jr. (2018)

[PRP.1] – Architecture is conceived through distinctions.

This definition comes from **[ADQ.4]** along with **[PRM.1]**. According to Kress (2009), meaning activities depend on *Modes* for signification process. These *Modes* present themselves through Multimodal arrangements. The architectural principle of *Order* can only be given by means of distinction: which Modes, or, according to **[PRM.1]**, which *Worlds* to distinguish, in what manner and under which arrangement.

[PRP.2] – Architecture is characterized by assumption and construction of Relational Models.

Figure 55 presented a set of arrows that connects the objects in each possible world. These arrows are instances of *Applied Relations* defined in **[PRM.2]**, as they analyze each instance on a structure of comparison based on distance. The set of *Applied Relations* [a, b, c, d, e, f, g, h] on possible worlds $[W_1, W_2, W_3, W_4]$ can be expressed through Modal Logic. A representation of each world indicates the existence of an *Applied Relation* by assigning the value true or false for it, as presented on figure 56.

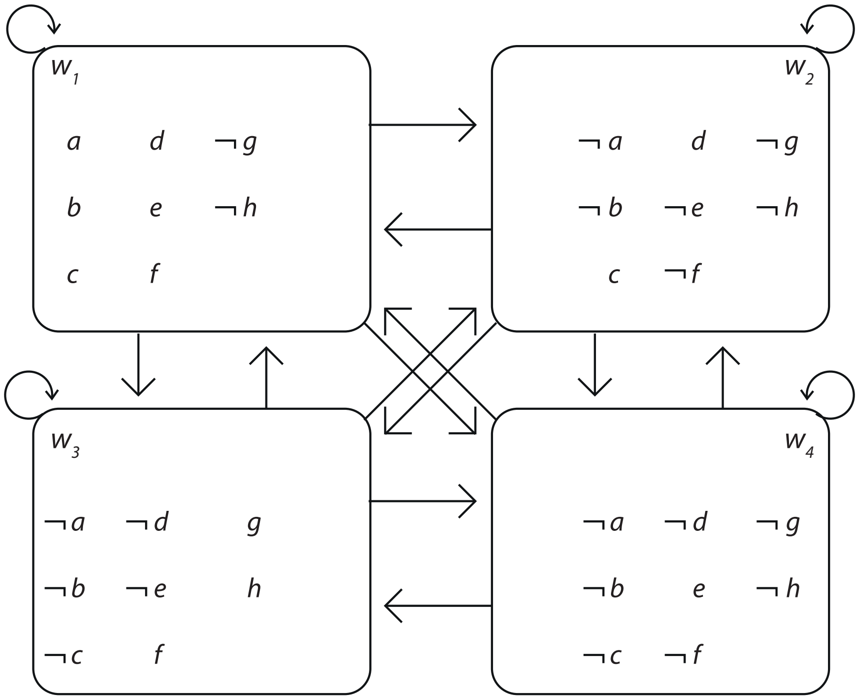


Figure 56 – Logical model of figure 55

Source: Adapted from Kuroki Jr. (2018)

[PRP.3] – Architecture should aim the economy of Relations.

Constructing as many relations as one subject can imagine would be an obvious path. However, as the number of relations gets higher the entropy grows in equal (or, sometimes exponential) ratio, but as Carnielli and Pizzi (2008) exposed, modal systems get stronger (in consistency and completeness) as the number of relations grows. To reach a measure of balance, other property is needed:

[PRP.4] – Architecture manifests through Contextual Rules.

This property is achieved by joining **[ADQ.2]** and **[ADQ.3]**. The fundamental nature of every model is to evolve, to change. As understanding things becomes more natural, some relations may be unnecessary for completeness and consistency of the model. By relevance, certain relation can be discarded but, in a future moment, be necessary again. In this manner, all ruling applied to the model cannot be considered final and absolute: continuous validation of the architecture presented is needed.

4.6.2.2 Information: adequations and properties

Several researches aimed a definition for Information with little success (KUROKI JR., 2018). For the definition of MIA, one idea seems to have no contenders by any position: Information can change things.

[ADQ.5] – Subjects and Objects *correlate* in multiples worlds, at the same time.

This statement comes form an interpretation of Phenomenology, adopted by Brier (2015) in his cybersemiotic view of Information. In a reductionist manner, each subject perceives an object, through a unique phenomenon. He/she never has direct access to the real essence of the object, it is always mediated through some other entity.

[ADQ.6] – Different Subjects can correlate with the same Object, at the same time.

It does not seem conceivable the existence of a situation where a Subject within a group of Subjects, coexisting in objective reality, be hinder of perceiving an Object and make his/her own presumption about it.

[ADQ.7] – Subject-Object atomic correlation phenomena tend to be unique.

Different subjects have their own internal convictions. Each person has his/her own thoughts and opinions. It is highly improbable that two Subjects present the same set of convictions. Joining all three adequations (**[ADQ.5]**, **[ADQ.6]**, **[ADQ.7]**), it is possible to conceive a graphical model for analysis, demonstrated on figure 57.

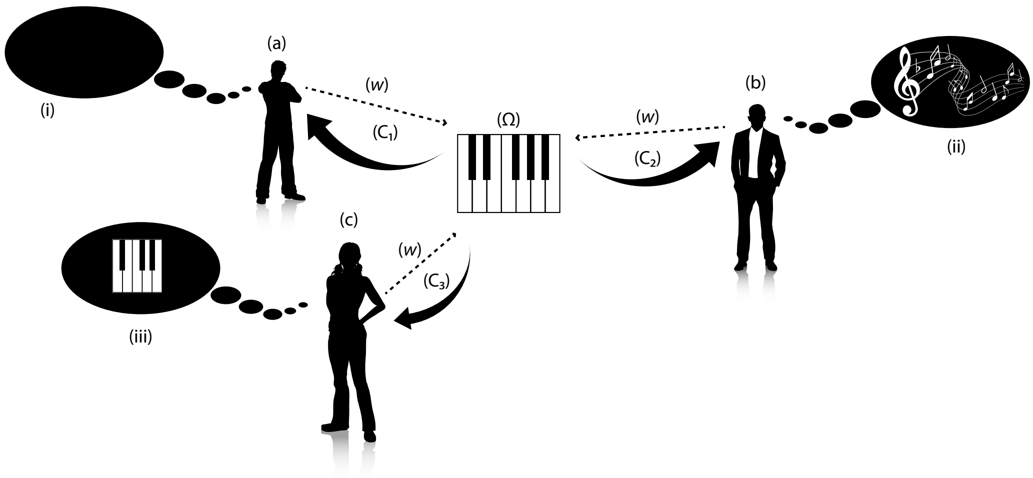


Figure 57 – Model of **[ADQ.5]**, **[ADQ.6]** and **[ADQ.7]**

Source: Kuroki Jr. (2018)

The model presents three Subjects [a, b, c] that realize atomic correlations [C1, C2, C3] with an Object. Each has his/ her own internal convictions, with three different results:

[RST.A] – Subject "a" perceives the Object, however, his internal convictions do not have any record that enable the signification of that Object, or it is irrelevant for him therefore discarding it ("never seen it before, it´s irrelevant").

[RST.B] – Subject “b” perceives the Object and it is compatible with some record in his internal conviction and correlates it with this record, by what makes possible a signification process (“it´s a piano keyboard that produces music”).

[RST.C] – Subject “c” perceives the Object and apprehend the properties presented, but do not correlates to any previous record so it is just stored on her internal conviction (“it´s a set of white and black rectangles”).

From these results, two properties were identified.

[PRP.5] – Information has state change capability.

This property aims to meet the positions of Brier (2015) and Floridi (2008), as it opens the interpretation that an instance of Information necessarily carries a potential charge that can be signified by a Subject. A complementary discussion starts when the phenomena are taken isolated: if the subject does not have any records in his internal convictions that can be matched or conjugated with the stimulus, is it not considered an instance of Information? The simple definition of "state change" is unsatisfactory. A second property is needed.

[PRP.6] – Information has a double potential vector: increase of complexity or reduction of uncertainty.

Based on Wilson and Sperber (2002) comes the interpretation that the search for relevance has fundamental influence on Relations between Subjects and Objects considering a context. On **[RST.A]** the stimulus is not relevant to Subject “a” and considering that there are no other stimuli to as complementation (an “implicature”, as Wilson and Sperber (2002) suggested), Subject “a” discards it. **[RST.B]** and **[RST.C]** explain the double-bias property of Information. If there is no correlation with a previous record by the Subject, but still he apprehends the stimulus received, the complexity of his internal state increases for future correlations. In case of correlation, the stimulus becomes part of the internal convictions in a complementary or supplementary way to previous records which it was joined. This action reduces the uncertainty of approximation of the image (what the Subject has for conviction that the Object means, in our example, a piano keyboard) conceived for the Object itself.

4.6.3 Defining MIA

Seven adequations where constructed which led to six properties applied to the concepts of *Architecture* and *Information*. Multimodality emerged as a key aspect as showed in **[ADQ.5]**. Multiples *worlds* of signification goes along with signification *Modes* described by Kress and Van Leeuwen (2001) and Kress (2009), leading to distinctions of worlds proposed in **[ADQ.1]** and **[ADQ.4]**. The measure of *Order* will be expressed through *economical ruling* as dictated in **[ADQ.2]** and **[ADQ.3]** within a highly complex context where *Subjects* and *Objects* correlate simultaneously, as described in **[ADQ.6]** and **[ADQ.7]**.

4.6.3.1 Architectural contribution of MIA

For Kuroki Jr. (2018) the concept of MIA needed to be constructed aiming technological implementation (following Van Gigch and Moigne (1989)). Each property was obtained from at least one adequation produced so, building the proposal by joining all of properties would automatically attend both. In this sense, the author created a new scenario, combining both architectural considerations (on figure 54) and informational considerations (on figure 57). Attending **[PRP.1]** comes form distinguishing worlds, displayed on figure 58.

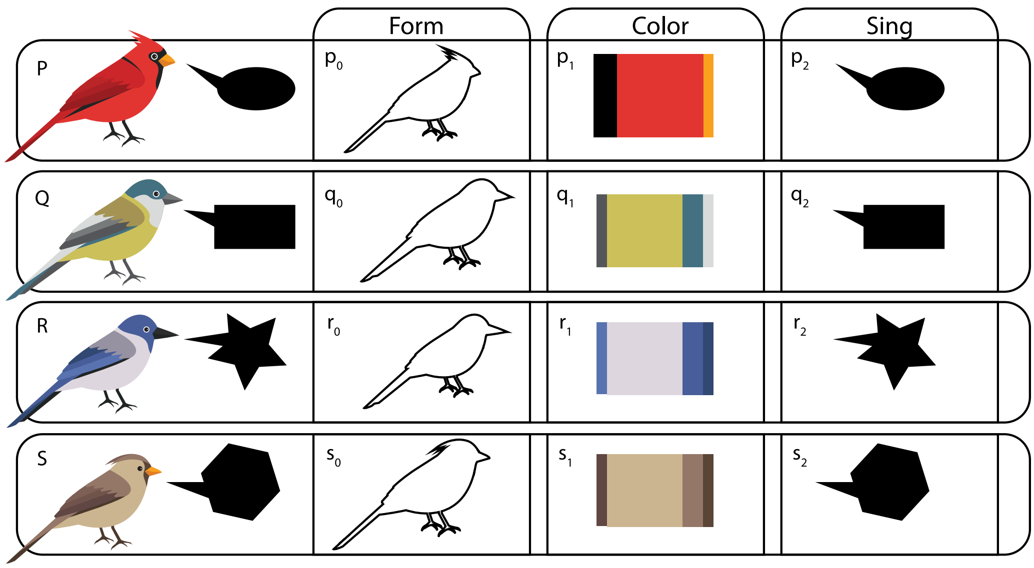


Figure 58 – Real context simulation

Source: Kuroki Jr. (2018)

Four Objects represented the birds identified as [P, Q, R, S]. The model presents three *possible worlds*: *form, color* and *sing*. This distinction of signification *Modes* allows us to conceive a model of relevance. For example, suppose that three individuals took some assumptions about these distinctions, producing a list of propositions stating if the stimulus presented refers to the semantic designation of the Object or not. In practice, it is showing the set of colors displayed on P_1 to each Subject and ask if these are a property of the semantic word for the bird P. This word could be the bird's name, scientific classification or any other socially agreed denomination that could represent the bird. If the Subject thinks it is, assigns "*true*" for P_1, if not, assigns "*false*". In resume, a possible result of this activity is presented on figure 59.

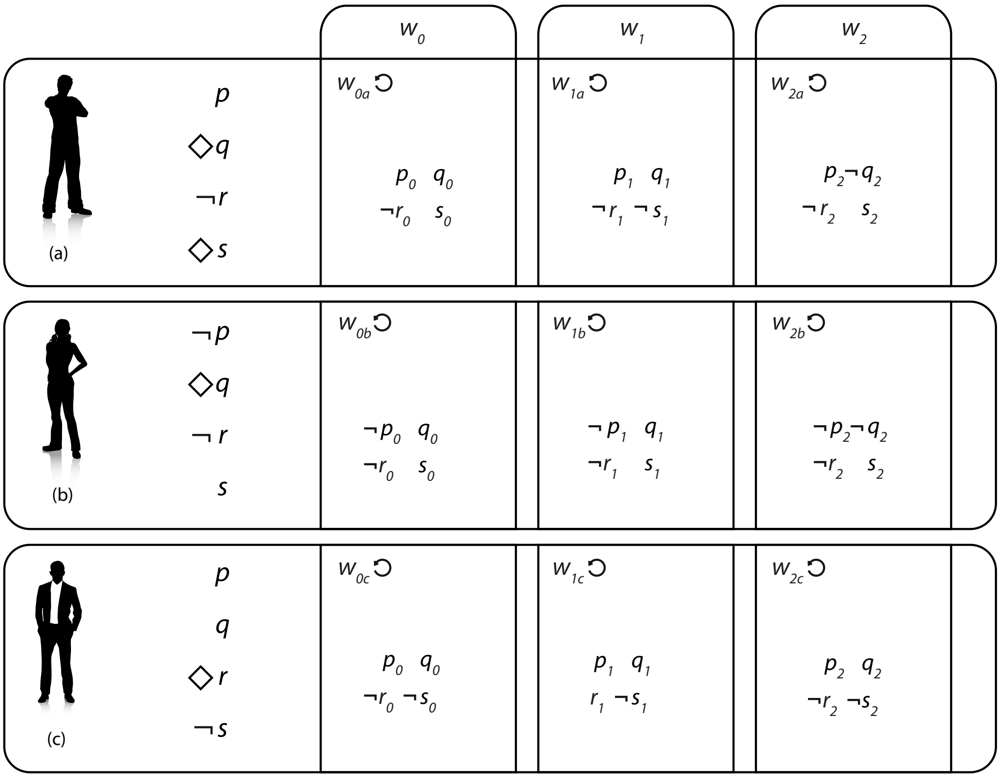


Figure 59 – Real context simulation

Source: Kuroki Jr. (2018)

The values came from correlations that each Subject realized to each stimulus, therefore, relations were established between the entities of our model. This event is closely related to **[PRP.2]**, which says that an Architecture is characterized by relational model assumption and construction. It is so close that it's possible to say that *Relational Models* are the tool for possible worlds distinction, making **[PRP.1]** and **[PRP.2]** complementary.

For each step of the procedure a graphical resume is presented in order to follow each stage of definition comparing to each property analyzed. table 5 shows the first step.

Table 5 – MIA Concept construction. **[PRP.1]** and **[PRP.2]**

[PRP]	Contribution on the definition
1	Distinction and construction of Architectural worlds
2	Through assumption of Relational Models
3	
4	
5	
6	

Source: Adapted from Kuroki Jr. (2018)

[PRP.3] says that an architecture should aim economy of relations. After the brief introduction to Modal Logic on section 4.6.1.3, is reasonable to say that *Euclidean Frames* tend to transgress this property. For instance, comparing a situation where three people talk to each other and considering what everyone has as convictions produces an *Euclidean Frame* of 3 symmetric relations totalizing 6 unitary relations. But when we add another person to this scenario the number of symmetric relations grows to 6, doubling the number of relations to 12.

But, what if these kinds of Frames simply happen? In a practical vision, let's get back to our model. Three people write down their opinion in a piece of paper. But what if, in a certain *Time* and *Space*, they can see each other's opinions? This is a kind of *Accessibility Relation*; therefore, an *Euclidean Frame* is established (as described before). A lot of other possibilities can be analyzed: do all Subjects trust each other? Do they consider each other opinion? What separates these contexts? The answer is simple, but very difficult to implement: *Time* and *Space*. Two people can consider an opinion but do not trust on who emitted that opinion on certain time or on certain circumstance (like talking about knowledge management or talking about politics) but is totally acceptable that these same individuals trust each other and, by that, not only consider that opinion but take it as a possible source of potential knowledge on some matter. This measure of dynamicity makes *Architecting* a constant and unstoppable activity. This is exactly what **[PRP.4]** stated: *Contextual Ruling*. Therefore, in our concept, **[PRP.3]** and **[PRP.4]** will be unified in one phrase as presented on table 6.

Table 6 – MIA Concept construction. **[PRP.3]** and **[PRP.4]**

[PRP]	Contribution on the definition
1	Distinction and construction of Architectural worlds
2	Through assumption of Relational Models
3 and 4	Grouped by Space-Time contexts
5	
6	

Source: Adapted from Kuroki Jr. (2018)

4.6.3.2 Informational contribution of MIA

So far it is defined that the activity here presented is characterized by "*distinction and construction of architectural worlds through assumption of relational models, grouped by space-time contexts*". The definition of the Object in this activity is still missing. **[PRP.5]** says that Information has state change capability. This can easily be exemplified by the development of our model exposed on figure 59 previously displayed. Consider now a new distinction which contains only Subjects (a) and (b). Initially a reflexive-symmetric *Frame* is applied as ruling to the relations between them. Figure 60 shows the result.

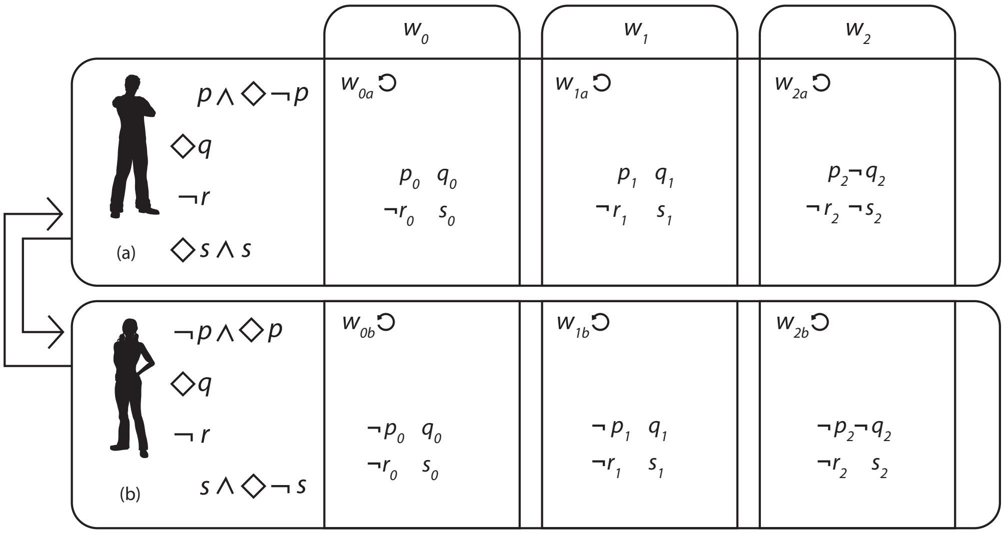


Figure 60 – Reconfiguration of figure 59 after new distinction and ruling applied

Source: Kuroki Jr. (2018)

As observed, the state of the internal convictions of Subjects (a) and (b) has changed. For (a) is now possible that the set mode of form P_0, colors P_1 and sing P_3 are not properties of the semantic word for the bird P; as for (b) now the same set of properties may be indeed related to the semantic word for the bird P. This phenomenon turned the internal convictions of the Subjects to *Information* level (internal conviction of Subject (a) is now *Information* to Subject (b) and vice-versa), therefore, actualizing the definition as showed in table 7.

Table 7 – MIA Concept construction. **[PRP.5]**

[PRP]	Contribution on the definition
1	Distinction and construction of Architectural worlds
2	Through assumption of Relational Models
3 and 4	Grouped by Space-Time contexts
5	Of Information states
6	

Source: Adapted from Kuroki Jr. (2018)

In objective reality it is hard to assume that people trust in each other's opinions. Considering that, symmetry does not seem to be a secure *Frame* to rely on. At the other hand, assume that no information font is secure lead us to complete anarchy, an arbitrary *Frame* for our relations. A reasonable solution for this question was presented through economy, which lead us to the space-time concept present in the definition of MIA so far. As an example, if we substitute the reflexive-symmetrical *Frame* adopted on figure 59 and replace it for a reflexive-serial *Frame*, but still admitting that space-time can change the *Frame*, we could get something like what is showed on figure 61 below.

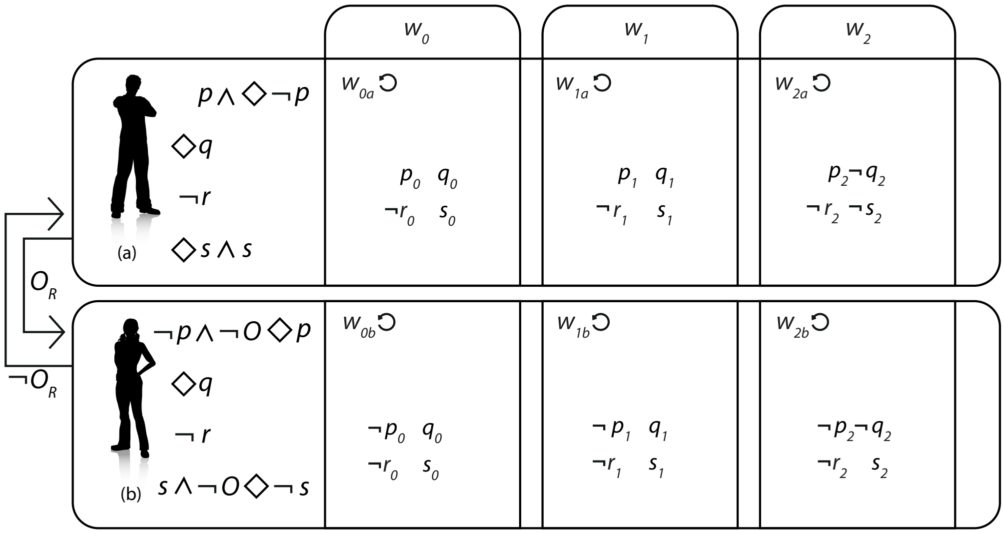


Figure 61 – Reconfiguration of figure 60 after Frame substitution

Source: Kuroki Jr. (2018)

Now Subject (a) *Obligatorily* consider the Information set produced by (b), but the same is not applied to Subject (b): it may not consider Subjects (a) Information. **[PRP.6]** states that *Information* has a double potential vector: increase of complexity or reduction of uncertainty. Both were pictured in figure 61. Increase of complexity for Objects *P* and *S*, reduce of uncertainty on Objects *Q* and *R*. Another facet of this situation is the incidence of *Relevance*. One of the possible reasons for Subject (b) discard Subject's (a) information is that it is irrelevant now but could become relevant in some future moment. Completing the definition, table 8 is presented with the full definition of MIA.

Table 8 – MIA Concept construction. **[PRP.6]**

[PRP]	Contribution on the definition
1	Distinction and construction of Architectural worlds
2	Through assumption of Relational Models
3 and 4	Grouped by Space-Time contexts
5	Of Information states
6	Correlated or not

Source: Adapted from Kuroki Jr. (2018)

4.6.3.3 MIA: full definition

MIA is characterized by the distinction and construction of architectural worlds through assumption Relational Models, grouped by Space-Time contexts of Information states correlated or not.

5 Applying Multimodal Information Architecture on Deep Learning procedures

MIA's definition suggests that, through *Relational Models* and Distinctions of architectural worlds, it is possible to construct arrangements that favor the correlation of *Information* states by Subjects that compose the model. The simulations proposed assume that two subjects change their internal convictions through communication, either at the exact moment of occurrence or later.

As MIA's was intended to aim at technological applications, Kuroki Jr. (2018) listed two preliminary questions that would drive future goals:

[Q.1] – Supposing that a third party can modify the configuration presented to the Subjects, whether including architectural worlds or presenting other convictions generated by other Subjects, how would this process occur?

[Q.2] – Would it be possible to design a sequence of actions to change these settings?

The author concluded that it would be at least plausible to consider the possibility of manipulation of the preconditions for the occurrence of Relations within a model.

For this statement, quotes a Carnielli and Pizzi (2008) logical modality called Dynamic Logic, which is characterized by the construction of propositions from abstract processes, typical of computers. Using a computer as Subject that interfere within a model significantly alters the possibilities of contextual design. Since (TURING, 1950), much is discussed about the ability of machines to construct mental models as men. It is proposed the discussion about the existence of architectural *World-building* forms that allow the modification of architectural contexts. In this sense, from figure 59 showed before, the author inserted a computer (M) assuming the internal convictions of the Subject (c) and, through the set of processes $\langle x; y; w; z \rangle$ it would be able to expose its architectural worlds to Subjects (a) and (b) and, through relations, change the context which they are inserted. Figure 62 show the results.

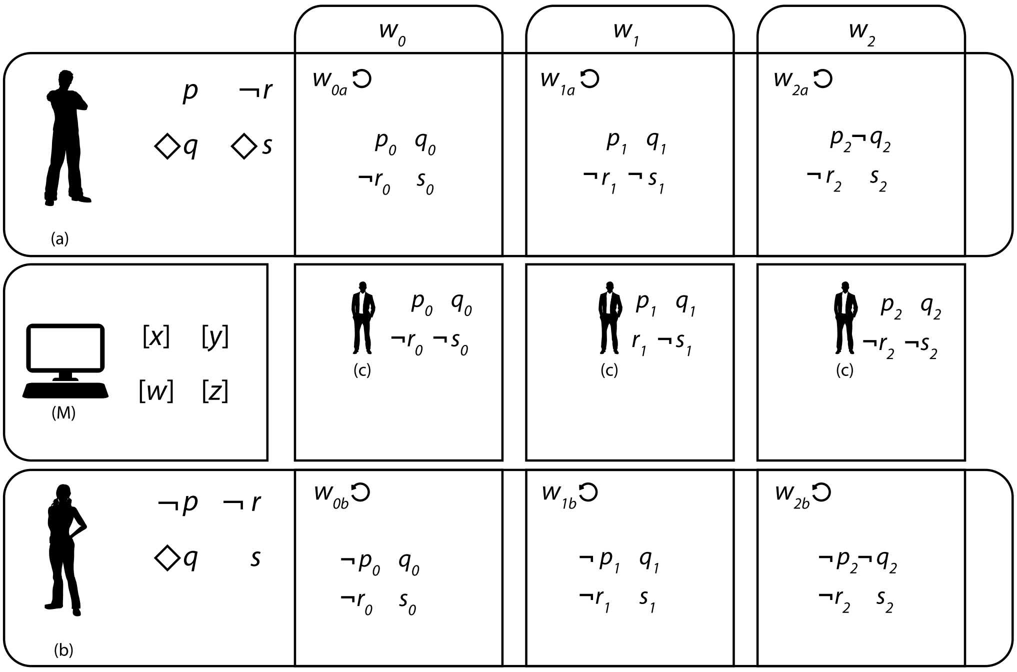


Figure 62 – A computer M acting on the model

Source: Kuroki Jr. (2018)

On the experiment, through the combination of processes ⟨x; y; w; z⟩ it would be possible to separate each Mode (as a syntactic layer) and extract meaning from that. For instance, executing processes *[x]* and *[w]* would separate Subject's (c) propositions for W_0. If presented to Subjects (a) and (b), even though Subject (c) thinks R is possible (because in W_1 it is true for him), for this moment it would be impossible, changing the construction of the architectural model if Subject (c) was indeed acting on it.

Another aspect of this experiment: what if computer (M) had access to all three Subjects assumptions and another Subject (d) was then analyzed by this computer while making the same task of assigning true or false for each set of stimuli? By each layer of analysis, computer (M) could predict Subject (d) next answer by comparing how close he is to either Subject (a), (b) or (c). At the end, assumptions assigned by Subject (d) are recorded and another parameter of comparison is added, so when another Subject start to classify the same stimuli the same process can be done. Here we are discussing only three architectural worlds of meaning but what if others are added? How can we decide if a world is relevant? This problem is a common one in other fields, as Artificial Intelligence.

5.1 Deep Learning and Text Classification open questions

Through sections 4.3, 4.4 and 4.5 a series of considerations on Deep Learning techniques where collected by reviewing Computer Sciences advances through several decades.

LeCun (1993), latter quoted by Haykin (2009), described the need for **maximizing information content**, where Deep Learning algorithms tend to extract more accurate features of the problem when presented with both volume and diversity of examples.

Haykin (2009) also brought the need for **normalizing the inputs** in order to avoid a set of examples with predominantly positive or negative results. This way, it would prevent the algorithm from learning only true cases or false cases.

Duda, Hart and Stork (2006) and Tesauro (1992) present the phenomenon of **Overfitting** which occurs while trying to obtain a close-to-perfection model of classification, sacrificing generalization capabilities.

Duda, Hart and Stork (2006) mentioned the occurrence of **missing features** on the data set (training, test or analyzed ones) which would lead to a misclassification of the input data. Also brought the need for **prior knowledge** on certain domains, as a path to obtain better classification algorithms.

Arel, Rose and Karnowski (2010) take back the initial problem observed by Bellman (1954) called **data dimensionality** which states that in order to avoid exponential growth of variables, a pre-processing stage of the data to be learned is recommended, calling this procedure as *feature extraction*.

Latter on, Wason (2018) describes the growing use of Deep Learning techniques throughout the 2010's, also mentioning that the Deep Learning science is still in its initial stages, making references to challenges found along the different implementations observed, for example:

- **Massive data sets:** Deep learning has found successful application in varied domains like computer vision, natural language processing, robotics etc. However, notably the number of data samples for an efficient learning should be 10X the number of parameters in deepnet;

- **Neural Network Over fitting:** there can be a significant difference in error reported in training data set and error encountered in real data set. This can be a common issue in large networks with multiple parameters thus affecting model efficacy;

- **Brittle Nature:** Deep learning networks are brittle in the sense that a trained network can only perform on the task it is trained for and performs poorly on any new task.

Recently, Minaee et al. (2021) reviewed how text classification problems have been treated with Deep Learning based networks. Resume all strategies into two approaches named *rule-based* and *data-driven* (also addressed as *machine learning based*) methods:

> Rule-based methods classify text into different categories using a set of predefined rules, and require a deep domain knowledge. On the other hand, ma-

> chine learning based approaches learn to classify text based on observations of data. Using pre-labeled examples as training data, a machine learning algorithm learns inherent associations between texts and their labels.(MINAEE et al., 2021, p.2)

The authors state that data-driven models have been widely used, with the most classical implementation adopting a two-step procedure. In the first step, some hand-crafted features are extracted from a certain data set. In the second step, those features are fed to a classifier to make a prediction. This method has several downsides:

- reliance on the handcrafted features requires tedious feature engineering and analysis to obtain good performance;
- strong dependence on domain knowledge for designing features;
- overfitting and overtrainning;
- cannot take full advantage of large amounts of training data due to features (or feature templates) pre-definition.

To explore and address these limitations, neural networks approaches focused on embedding models that can map text into a low-dimensional continuous feature vector, trying to overcome the **data dimensionality** problem identified since Bellman (1954).

A tool that has been used since 2018 is Pre-trained Language Models, which is a set of large-scale Transformer-based algorithms trained in a very deep neural network with large amount of text corpora in order to learn contextual text representations by predicting words conditioned on the context. These PTMs are fine-tuned using task-specific labels, and have created new state of the art in many downstream NLP tasks, including TC (MINAEE et al., 2021). PTMs are than used to enrich text classification analysis intended on regular text classification networks.

Minaee et al. (2021) present five-step tutorial for text classification neural network model choice:

[Step.1] **PTM Selection:** using PTMs leads to significant improvements across all popular text classification tasks, and autoencoding PLMs (e.g., BERT or RoBERTa) often work better than autoregressive PLMs (e.g., OpenAI GPT);

[Step.2] **Domain adaptation:** most PTMs are trained on general-domain text corpora (e.g., Web). If the target domain is dramatically different from general domain, we might consider adapting the PTM using in-domain data by continual pre-training the selected general-domain PTM. For domains with abundant unlabeled text, such as biomedicine, pretraining language models from scratch might also be a good choice.

[Step.3] Task-specific model design: Given input text, the PTM produces a sequence of vectors in the contextual representation. Then, one or more task-specific layers are added on the top to generate the final output for the target task. The choice of the architecture of task-specific layers depends on the nature of the task, e.g., the linguistic structure of text needs to be captured.

[Step.4] Task-specific fine-tuning: depending on the availability of in-domain labels, the task-specific layers can be either trained alone with the PTM fixed or trained together with the PTM. If multiple similar text classifiers need to be built (e.g., news classifiers for different domains), multi-task fine-tuning is a good choice to leverage labeled data of similar domains.

[Step.5] Model compression: PTMs are expensive to serve. They often need to be compressed via e.g., knowledge distillation to meet the latency and capacity constraints in real-world applications.

After reviewing over 150 Deep Learning models for text classification and more than 40 data sets, on their conclusion is mentioned that even though great progress was achieved, some questions still challenging to the field:

- **Absence of data sets for more complex tasks:** although a number of large-scale data sets have been collected for common text classification tasks in recent years, there remains a need for new data sets for more challenging TC tasks such as QA with multi-step reasoning, text classification for multi-lingual documents, and TC for extremely long documents;

- **Commomsense knowledge models:** Incorporating commonsense knowledge into DL models has a potential to significantly improve model performance, pretty much in the same way that humans leverage commonsense knowledge to perform different tasks. For example, a QA system equipped with a commonsense knowledge base could answer questions about the real world. Commonsense knowledge also helps to solve problems in the case of incomplete information. Using widely held beliefs about everyday objects or concepts, AI systems can reason based on "default" assumptions about the unknowns in a similar way people do

- **Memory Efficient Models:** most modern neural language models require a significant amount of memory for training and inference. These models have to be compressed in order to meet the computation and storage constraints of edge applications. This can be done either by building student models using knowledge distillation, or by using model compression techniques.

- **Few-Shot and Zero-Shot Learning:** most DL models are supervised models that require large amounts of domain labels. In practice, it is expensive to collect such labels for each new domain.

5.2 MIA contributions on Text Classification

This thesis seeks to position Multimodal Information Architecture as a domain-establisher tool to text classification Deep Learning methods. It can be noticed that Minaee et al. (2021) **[Step.2]** and **[Step.3]** mention the domain of the problem, but treat it as an agglutination of semantic values, that is, no *architectural world* is identified and properly distinguished based on Kuroki Jr. (2018) view of Kress (2009) definition of semantic *Mode*.

Afterwards, when observing the transition between **[Step.3]** and **[Step.4]** we can also notice a gap when confronting Minaee et al. (2021) 5-step tutorial and Kuroki Jr. (2018) MIA: how can we identified task-specific layers to be trained alone or together with the PTM? It is mentioned that "*if multiple similar text classifiers need to be built (e.g., news classifiers for different domains), multi-task fine-tuning is a good choice to leverage labeled data of similar domains*", but how to do it when on **[Step.2]** no domain-distinction was made, except from treating a group of semantic *Modes* all together?

This gap needs to be treated before **[Step.2]**, therefore, MIA must act as a domain-establisher for later neural network decisions. Initially, five main operations are proposed:

[MIA.1] Identify context entities;

[MIA.2] Identify entities correlations;

[MIA.3] Domain distinctions;

[MIA.4] Proposition of relationship between domains;

[MIA.5] Space-time context-based groupings.

These operations can be grouped in to phases in order to obtain a process for domain establishment.

5.2.1 Identify context entities

Kuroki Jr. (2018) proposal presupposes the existence of Subjects and Objects that correlate with each other, producing distinct information spaces. In an example described by the author, it is possible to observe how these spaces are set in **[ADQ.5]**, **[ADQ.6]** and **[ADQ.7]**, which led to **[PRP.5]** and **[PRP.6]**.

On NLP and PTMs researches, **Context** can be simply viewed as a certain group of text that are put together through categorizing them by linguistic, semantic, factual, commonsense or any other given characteristic. That is not the case for MIA. For a context to be an architectural space, it necessarily has to consider at least one Subject point of view of at least one Object.

Applying this scenario to neural networks trained through supervised learning, it must be considered that it is possible for the same Object (an input signal) to be classified differently by different tutors. Similarly, in unsupervised learning, the occurrence of pattern identification in certain circumstances and non-identification in a different situation, dealing with the same object, cannot be ruled out.

Learning techniques recognize this fact and try to overcome it through volume and repetition: the larger the training sample, the less impact any noise will cause. The problem occurs when there is no properly classified data to undertake effective training as pointed out by Wason (2018) and Minaee et al. (2021). Thus, MIA needs to follow a path that changes the configuration of the sample in order to enhance its relevance. To initiate such an attempt, it is necessary to distinguish context entities that appear in these samples, as defined in **[PRP.1]**.

In Deep Learning networks this practice is done by identifying relevant variables to the problem, following Wilson and Sperber (2002):

> utterances raise expectations of relevance not because speakers are expected to obey a Cooperative Principle and maxims or some other specifically communicative convention, but because the search for relevance is a basic feature of human cognition, which communicators may exploit. (WILSON; SPERBER, 2002, p.251)

There are two points to be verified: problem definition and variable relevance analysis. A problem can contain several contexts, where several entities appear. Each entity has its characteristics that may or may not interfere in the analysis and resolution of the problem. In this sense, entities may or may not be variables. A practical example of this situation:

- **[a]** A group of people observe and try to understand the same object;
- **[b]** To form a uniform body of knowledge, the group institutes common terms to identify fundamental characteristics of this object;
- **[c]** To understand the object in its fundamental nature, they experiment with different techniques and, through the common vocabulary, form a methodological corpus that present the best results to understand this object;

For this particular group, delimiting that the context is the body of knowledge about the referred object, we would have:

[i] Each individual in the group is an entity with the capacity to produce and manipulate information (attributes of an object and any kind of relation), herein called SUBJECT;

[ii] Each common term instituted by the group, with meaning potential and present a set of attributes that can be interpreted in a common way is an entity that can be correlated with, herein called an OBJECT;

[iii] A correlation occurs when, in some manner, a SUBJECT transform an OBJECT by means of definition, comparison, fusion or decomposition, and the product of this operation is accepted on the body of knowledge;

[iv] Groupings of relevant entities become variables when, in some way, they influence one or more domains.

5.2.2 Identify entities correlations

One of Kuroki Jr. (2018) epistemological constraint is about the nature of relations (identified on **[ADQ.1]**). For the author, there is a strong belief in MIA that the relations are objective, that is, they are not some creation that a person construct totally disconnected from reality. At the other hand, defining it as objective does not mean that all relations are real: indeed that are relations made by a subject that aren't real in true world. Therefore, the definition relation is as follows:

> Relation is any form of connection between instances on a world or between worlds. (KUROKI JR., 2018, p.83. Free translation)

As said in section 5.2.1, initially there are four kind of relations that can be defined as follows. All relations are constructed by SUBJECTS towards an OBJECT or a GROUP OF OBJECTS.

[Rel.1] **Definition:** any correlation made by a subject that set the state of a certain "thing" in a world as an OBJECT, therefore, initiating the possibility of gathering other things to be related to it as an attribute. A simple example can be made through collecting words at random from a long text. Depending on what CONTEXT this collecting of words will be put on, some of them can or cannot be defined as an object. Consider the following sentence:

> It is fact that not only the shelf life but also the quality of food is important to consumers led to the concept of preserving foods using preservation methods. Therefore, alternative or novel food processing technologies are being explored and implemented such as Microwave heating, High Pressure Processing (HPP), Ohmic heating, Ozone processing, Atmospheric Pressure Plasma (APP), Ultrasonic (Knorr et al., 2009)

As the CONTEXT of the text is FOOD INNOVATIVE TECHNOLOGIES, the word *preservation* can be classified as an object:

- It can now aggregate other "things" (words) to form sub-contexts;
- It can be used to define gatherings of other words, as an idea of something that actually exists, like a process, a method or any other description of reality;
- It gains a potential variable status, as it can now define a problem or be part of a problem solution.

[Rel.2] **Comparison:** only entities that have been through definition can be compared. Any kind of comparison goes through putting side-by-side object attributes that were previously defined.

[Rel.3] **Fusion:** comes from gathering two objects to form another one. Following the same example of food technology, fusion could come from gathering the object "*high pressure*" (which can be related to a simple cooking technique) the with the object "*processing*" (which can be related to how the food is obtained: processed or natural), originating "*high pressure processing*" (which can now be related to a method of enhancing food quality).

[Rel.4] **Decomposition:** on the opposite side of fusion, decomposition originates two objects from one.

5.2.3 Domain distinctions

Making reference back to **[ADQ.4]**, a *World* is a Mode in Kress and Van Leeuwen (2001) point of view. This definition is the object to what Qiu et al. (2020) defined in Multimodal PTMs. This is only a partial view of *World* to MIA, as it also consider modal logic in its core in the concept of *Possible worlds*.

Combining these definitions with this thesis established view of context, a **DOMAIN** is a group of **ATTRIBUTES** that can be commonly identified by **SUBJECTS** throughout similar **CORRELATIONS** with **OBJECTS**. Thus, subjects and objects can figure on more than one domain that, analyzing through Kuroki Jr. (2018), can be classified as a extended view of *World*:

> *World* is a *Mode* where meaning can be expressed. (KUROKI JR., 2018, p.85 Free translation)

The extension comes from analyzing a simple situation as exposed on figure 63. On the example three subjects knowledge are assembled into two *Possible worlds*. All objects are represented on each model with the correspondent symbol for either necessary ($\Box$) or possibility

(◇) modal logic notation. Therefore, a domain not only is a *Mode* where meaning is expressed, but a particular set of possible worlds that depends on the sample of knowledge in question.

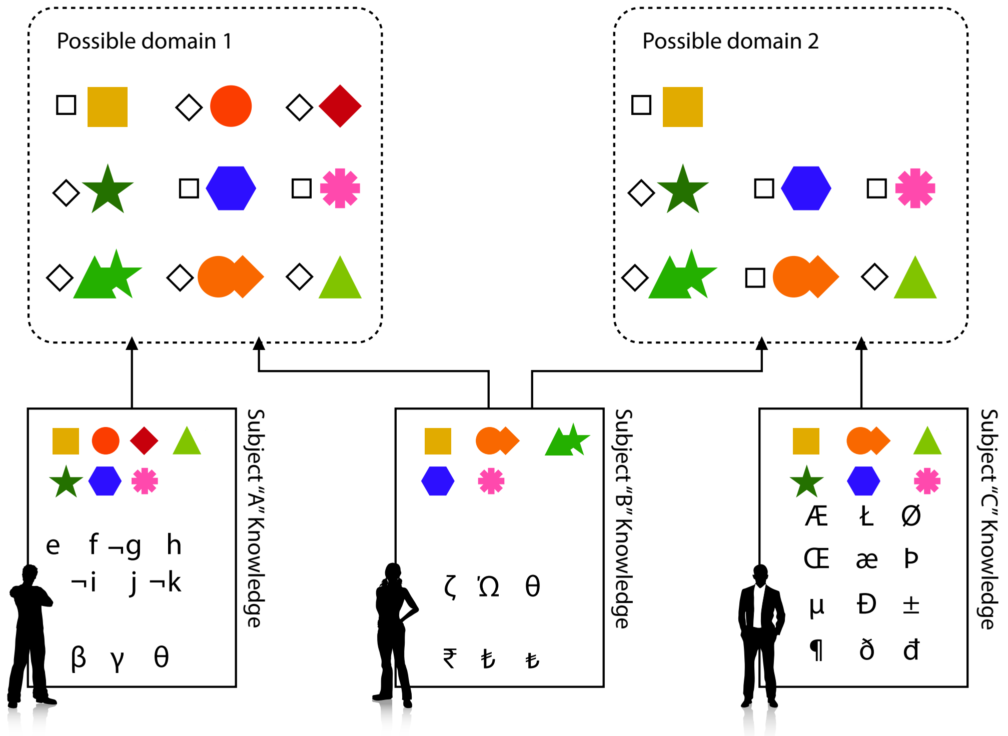


Figure 63 – Domain definition model

Source: Produced by the author in March 2022

Possible Domain 1 was assembled considering only Subjects *A* and *B* knowledge. We can find 9 objects which three of them are necessary (with the modal symbol □). This means that considering an analysis of an input vector with *Possible Domain 1* as source for model learning (the network would be trained with examples that fit *Possible Domain 1* criteria), the output would classify the input as part of *Possible Domain 1* if and only if the three necessary objects are identified in the vector. Comparing *Possible World 1* and *Possible World 2*, criteria for fitting an input vector as part of the domain would rise from 3 to 4.

Another issue to be addressed is the fact that to absolutely isolate knowledge into "boxes" seems not possible. As MIA states on **[PRP.5]** and **[PRP.6]**, information instances can be correlated — either to an real object or another representation. Taking Subject "*A*" from figure 63 and inserting semantic meaning to the objects in the set, it would be possible to have an configuration of knowledge leading to figure 64.

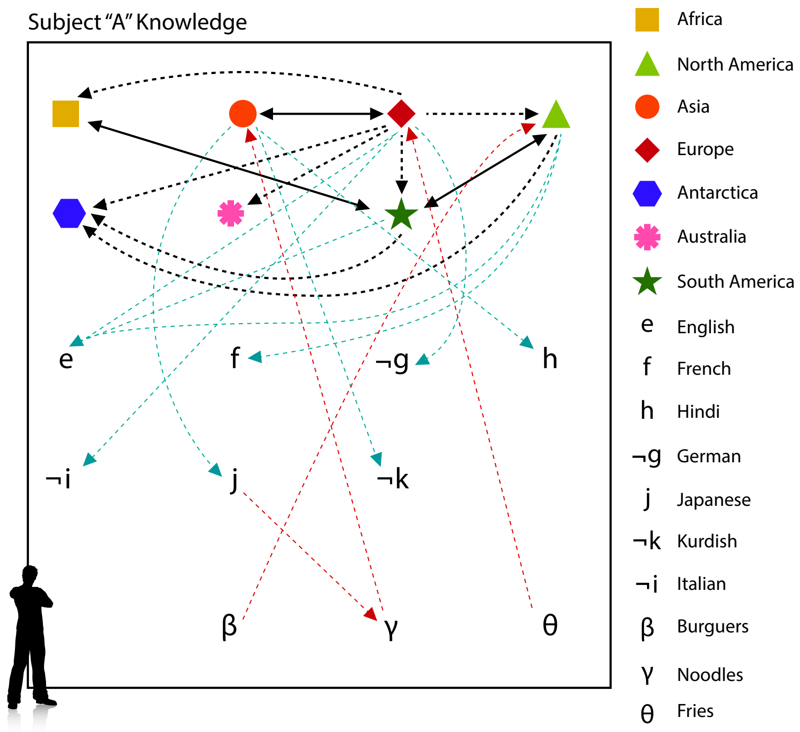


Figure 64 – Instance of Subject's A knowledge from figure 63

Source: Produced by the author in March 2022

Analyzing figure 64, it exemplifies knowledge from Subject "*A*" about geography, idioms and food. These categories could be express through the following attributes on each object:

- Geography: area, population, population density, religion, United Nation states, largest cities, countries.
- Idioms: native to, region, ethnicity, dialects, language family.
- Food: place of origin, region or state, main ingredients.

Possible Domain 1 only considers part of Subject "*A*" knowledge and do not take in account all the other relationships that Subject "*A*" has with idioms and food on geography. Therefore, considering the following sentence:

> Noodles became a part of daily meals since all ingredients where available on the kind of landscape available.

For the network to recognize this sentence at least as a possible instance of geography description for some region or country it would be necessary to acquire knowledge from other areas, in the case, from food. This only illustrate that defining the domain is a major issue to overcome data dimensionality: if informational spaces aren't properly defined, there is no other option other than mapping all knowledge available.

As objects are been defined by subjects, the possibility of distinguishing other worlds from the same set of objects grows proportionally. That comes from Kuroki Jr. (2018) definition quoted on item 5.2.1. Being world a *Mode* where meaning can be expressed, it automatically incorporate what have been described as fundamental conditions for MIA: a group of subjects that share some definitions about reality. As meaning and knowledge can be considered as continuous processes, it is inevitable to see world distinctions as continuous processes as well.

Also, for objective reality being difficult to define in an absolute manner (seen on **[ADQ.7]**), it seems inconceivable to determine whether worlds are distinguished prior to objects and subjects or the inverse order is a more suitable definition. It will depend on multiple variables of the context in which subjects and objects are inserted. On MIA, to achieve a measure of World distinction, that is, separate informational spaces, it is imperative to attend **[PRP.5]** and **[PRP.6]**, which is done through dealing with **[ADQ.5]**, **[ADQ.6]** and **[ADQ.7]**. Therefore, a domain need to be distinguished either from three points of view:

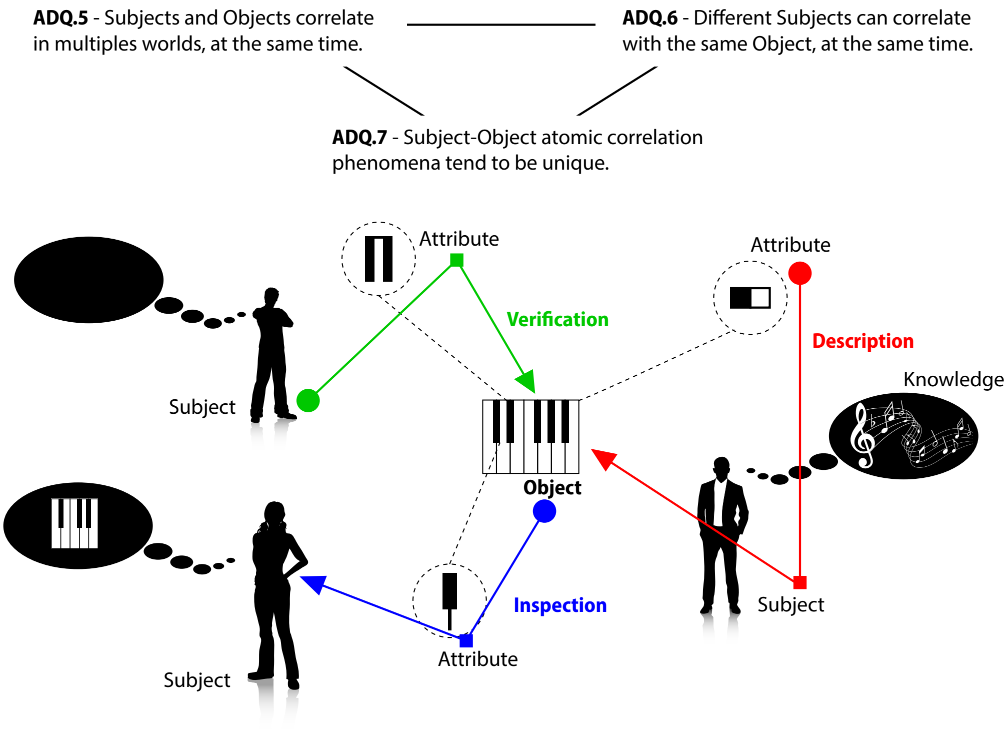


Figure 65 – Instance of Subject's A knowledge from figure 63

Source: Produced by the author in March 2022

[i] **Description:** describe a set of pre-determined attributes, verify acknowledgement by a group of subjects and identify these attributes on certain objects;

[ii] **Inspection:** analyze a set of objects, identify common attributes and verify if these attributes are commonly recognized by a group of subjects

[iii] Verification: inquiry a group of subjects, identify attributes that the group share the same perception and search for objects that meet the criteria.

5.2.4 Propose relationship between domains

Up to this point only the informational part of MIA has been dealt with. Both steps of Identify context entities and Domain distinctions addressed **[PRP.5]** and **[PRP.6]** based on **[ADQ.5]**, **[ADQ.6]** and **[ADQ.7]**. Domains were identified with a set of attributes that can be identified on certain objects by a group of subjects.

For MIA to produce impact on any context it is necessary to operate some changes on the informational space being treated. Otherwise, all products obtained throughout the process can be viewed as a trivial classification, therefore, produce the same results on any text classification task as gathering some text.

The architectural part of MIA needs to address **[PRP.1]**, **[PRP.2]**, **[PRP.3]** and **[PRP.4]** through attending **[ADQ.1]**, **[ADQ.2]** and **[ADQ.3]**. Figure 53 presented how these concepts are connected. Identify entities correlations deals with **[ADQ.1]** only at the object level, not being able to produce an information architecture on implementation level.

On this section, we will address **[PRP.1]**, **[PRP.2]** and **[ADQ.1]**, **[ADQ.2]**.

For Kuroki Jr. (2018), an *Architecture* is composed by *Relations* which have *Rules* that restrict them. So, the gathering of domains only presents the amount of information to be (re)organized — the problem itself, not the solution. To make an impact in the informational space configuration, some operations between domains are proposed in order to either change it or to create a new one. To guide this new configuration, MIA stands on modal operators (to express the relations) and frames (to express rules). Three main relations categories are proposed:

[i] Identity: an identity relation is obtained when all attributes of one domain can be found on another domain. It corresponds to the modal operator of **necessity ($\Box$)**.

[ii] Proximity: a proximity relation is identified when part of the attributes of one domain can be found on another domain. It corresponds to the modal operator of **possibility ($\Diamond$)**.

[iii] Incidental: incidental relations are not always perceivable and, into some extend, present a random character. The simplest way to explain is defining them as a second-order relation. Let $R_{i,p}$ be a relation of identity or proximity, if $R_{i,p}[A,B]$ and $R_{i,p}[C,A]$, an incidental relation would be possible if $R_{i,p}[C,B]$ occur embedded in $R_{i,p}[A \cup R_{i,p}[C,A],B]$.

As for Rules, figure 52 presented 5 types of frames: reflexive, serial, symmetric, transitive and euclidean. Each of them presents characteristics that needs to be identified in order to define each relation influence on the new (or renewed) informational space:

- **[i] Reflexive:** a reflexive frame is identified when the relation proposed is applicable to a domain from itself.
- **[ii] Serial:** a serial frame is identified when the relation proposed is applicable from a domain to another domain.
- **[iii] Symmetric:** a symmetric frame is identified when the relation proposed is mutually applicable between two domains.
- **[iv] Transitive:** a transitive frame is identified when, considering three domains [A, B, C], if A has the proposed relation with B and B has the same relation with C, then A has the proposed relation with C.
- **[v] Euclidean:** an euclidean frame is identified when the relation proposed is reflexive, symmetric and transitive.

To materialize Relations and Rules and their influences on informational domains, lets consider the following situation. Departing from knowledge of Subjects A and B, it would be possible to distinct 4 domains: geography, idioms, food and geology.

Aiming on a new information configuration for the domain geography, both subjects were asked to map two relations within the aimed domain — "*border*" and "*colonize*". In order to comprehend how the aimed domain relate to the others for each subject, suppose both of them are asked to express the connection between certain instances of these domains through a relation named "*remind*".

The semantic nature of the last relation is broad, in order to eliminate any influence of the relation name on the analysis. in certain aspects, it can be consider as a meta-class of epistemic-doxastic logics, which combines knowledge and belief, and deontic logics which deals with obligation and permission, both cited by Carnielli and Pizzi (2008). This assumption leads to Portner (2009) definition that all epistemic frames are reflexive frames, since if some one knows that p is true in w, then p is true in w.

Them, the nature of this relation, for each subject, comes to the following analysis: does every instance of *food* reminds one instance of *geography*? If indeed it does, the nature of the frame is NECESSARY. If not, it is POSSIBLE.

Figure 66 simulate the scenario with the following results:

- **[a]** On both subjects A and B point of view, a serial frame **is necessary** departing from the *food* domain to the *geography* domain;

[b] On subject *A* point of view, a serial frame **is possible** departing from the *geography* domain to the *language* domain;

[c] on subject *B* point of view, a symmetric frame **is necessary** departing from the *geology* domain to the *geography* domain;

[d] on subject *A* point of view, a serial frame **is possible** departing from the *language* domain to the *food* domain.

[e] it would be possible to extend, from subject *A* knowledge, that a transitive frame **could be applied** considering the domains [*geography*, *idioms*, *food*].

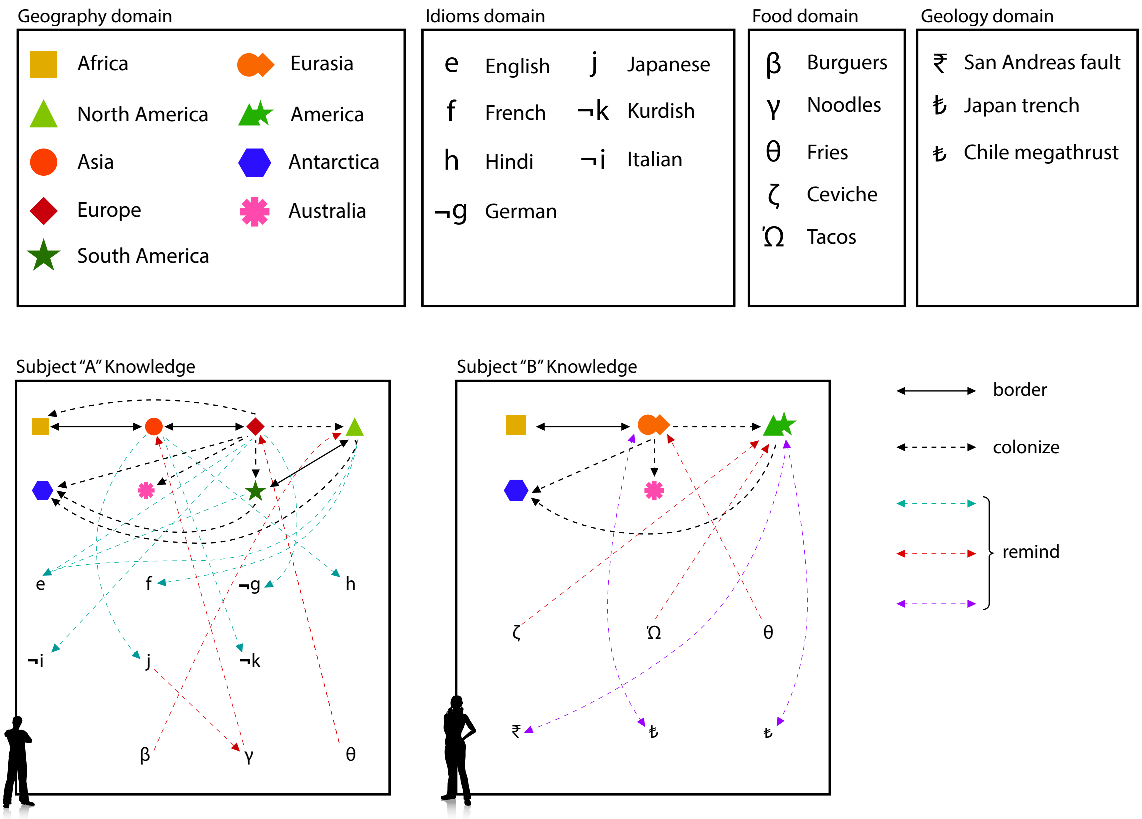


Figure 66 – Separating domains from figure 63

Source: Produced by the author in March 2022

As Portner (2009) described, and Kuroki Jr. (2018) also adopted, all relations on any frame can be seen as *accessibility relations* $R(w, v)$, meaning that the value of v is accessible from w considering the relation R. For applied MIA purposes, when a domain reaches another means that the origin domain is an "*extension*" of the destination domain. This notion of "*extension*" differs from the traditional software engineering view. An extension object (the origin domain) does not inherit all characteristics from the extended object (the destination domain): it adds characteristics to the extended domain.

Bringing this definition into our example, *R(food, geography)* means that *geographic* aspects can be accessed from *food* instances. For example, inquiring Subject *A* about *noodles*, besides from all characteristics that *noodles* has as *food* (place of origin, region or state, main ingredients), he/she would also *remind* characteristics from *Asia* (area, population, population density, religion, United Nations states, largest cities, countries). A graphical demonstration should be as figure 67:

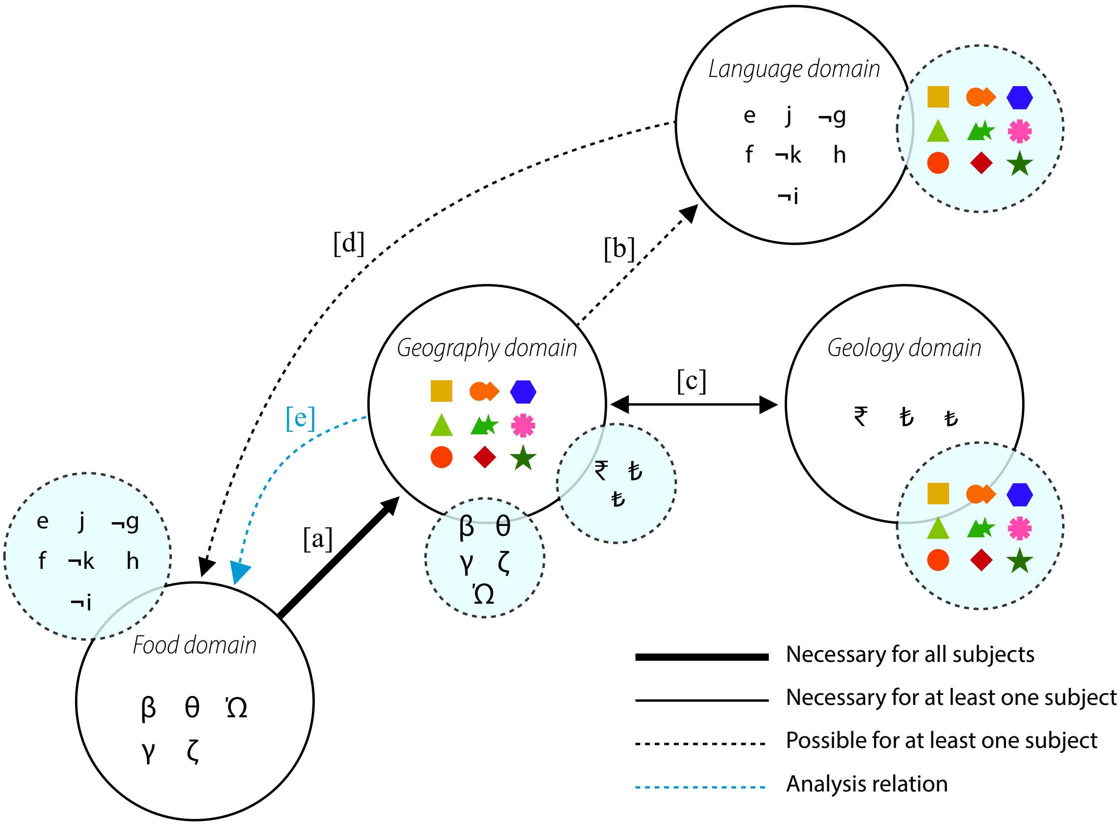


Figure 67 – Separating domains from figure 66

Source: Produced by the author in March 2022

Considering **[ADQ.5]**, multiple *Subjects* can correlate with multiple *Objects* within multiple *Worlds*. Therefore, a domain analysis necessarily needs to take in account all relations assumed by all subjects that can act on the domain. Applying this view to the results obtained above, considering the informational space given by the knowledge of subjects *A* and *B*:

1. The *food domain* **necessarily** reminds the *geography domain*, since all subjects on the model assume a serial frame between these domains;

2. The *geography domain* **possibly** reminds the *language domain*, since only subject *A* assume a serial frame between these domains;

3. The *geology domains* and *geography domain* **possibly** remind each other, since only subject *B* assume a reflexive frame between these domains;

4. The *language domain* **possibly** reminds *food domain*, since only subject *A* assume a serial frame between these domains;

5. It would be **plausible** to assume that the *geography domain* can **possibly** remind the *food domain*, if and only if, the *language domain* be considered, through an transitive frame.

At last, classifying each relation within the model becomes possible. As mentioned earlier, there are three main classes of relations within a model — identity, proximity and incidental. From these definitions, it is possible to interpret that an identity relation only occurs when all subjects presents the same relation, regarding to domains involved and totality of instances. All relations that cannot meet these constraints are considered to be proximity relations. As the initial goal was to produce a new geography domain, the following results are possible, with a graphical model exhibit on figure 68:

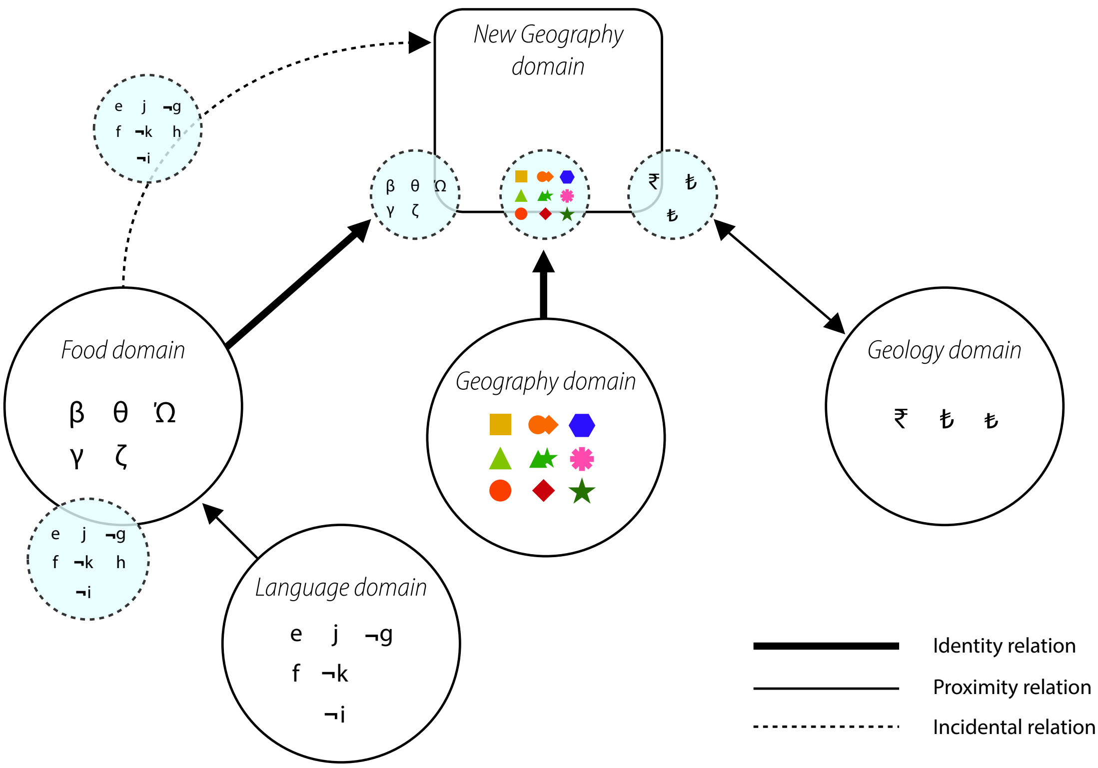


Figure 68 – New domain configuration

Source: Produced by the author in March 2022

i. The *new geography domain* presents an identity relation with the *geography domain*, since epistemic frames are reflexive;

ii. The *new geography domain* presents an identity relation with the *food domain*, since all subjects recognize a serial frame departing from the food domain.

iii. The *new geography domain* presents a proximity relation with the *geology domain*, since subject *B* recognize a symmetric frame between geography and geology domains;

iv. The *new geography domain* could inherit a proximity relation with *language domain*, if and only if, an incidental relation be considered between *geography*, *food* and *language* domains.

5.2.5 Space-time context-based groupings

Up to this point, as *Rules* and *Relations* were identified, the most obvious path is to apply them all. This is not the goal intended while applying MIA, since **[ADQ.3]** was not addressed yet, therefore, **[PRP.3]** and **[PRP.4]** are still missing in our model.

For MIA, *Economy* is what enables *Rules* and *Relations*. Without some measure of economy, any domain configuration will tend to completely reproduce objective reality, which seems implausible.

According to Kuroki Jr. (2018), space-time contexts can be identified through deontic frames. For Portner (2009), deontic frames are related to the concepts of obligation and permission. It differs from epistemic frames (which deals with knowledge) on the nature of frames that can be applied in each case:

a. Epistemic frames are reflexive frames, since it is a property of knowledge that if someone knows p in a world w, then p is true in w. On the authors own words, it would be similar to assuming that if John knows that it's raining right now, then it is indeed raining right now. The axioms applied would be:

$$\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q) \tag{5.1a}$$

$$\Box p \rightarrow p \tag{5.1b}$$

b. Deontic frames cannot assume reflexivity, only seriality. The author gives a simple example of common moral precepts. "No murder" can be assumed to be an universal precept in every conceivable world but nevertheless there is murder. Instead, we can say that deontically, if there is a set of rules, they need to be satisfied all together, as in a serial frame, where if there a set of rules applied on a world w, a relation between w and w' if and only if all rules applied on w would be applied on w'. The axioms applied would be:

$$\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q) \tag{5.2a}$$

$$\Box p \rightarrow \Diamond p \tag{5.2b}$$

The main difference resides on the fact that necessity on epistemic frames implies truth (since is a matter of knowledge) and on deontic frames it only implies possibility (since is a matter of obligation, which may be infringed). Getting back to the example of this chapter, the *new geography* domain would have the following characteristics:

i. The *new geography domain* presents an identity relation with the *geography domain*, since the latter is the basis for all subjects to form the new domain, configuring an serial frame;

ii. The *new geography domain* presents an identity relation with the *food domain*, since all subjects recognize a serial frame departing from the food domain.

All possible relations cannot be considered on the first analysis, since possibility is considered only in **Euclidean** frames. The axioms applied, in this case, would be:

$$\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q) \tag{5.3a}$$

$$\Box p \rightarrow p \tag{5.3b}$$

$$\Diamond p \rightarrow \Box \Diamond p \tag{5.3c}$$

Euclidean frames only occur when all worlds are connected by symmetric and serial frames between then and they are all reflexives considering themselves. On our example, it would be similar to selecting a group of subjects that have knowledge about the four domains (*food*, *geography*, *language* and *geology*) and all these domains where connected through symmetric and transitive relations. Figure 69 demonstrate it visually.

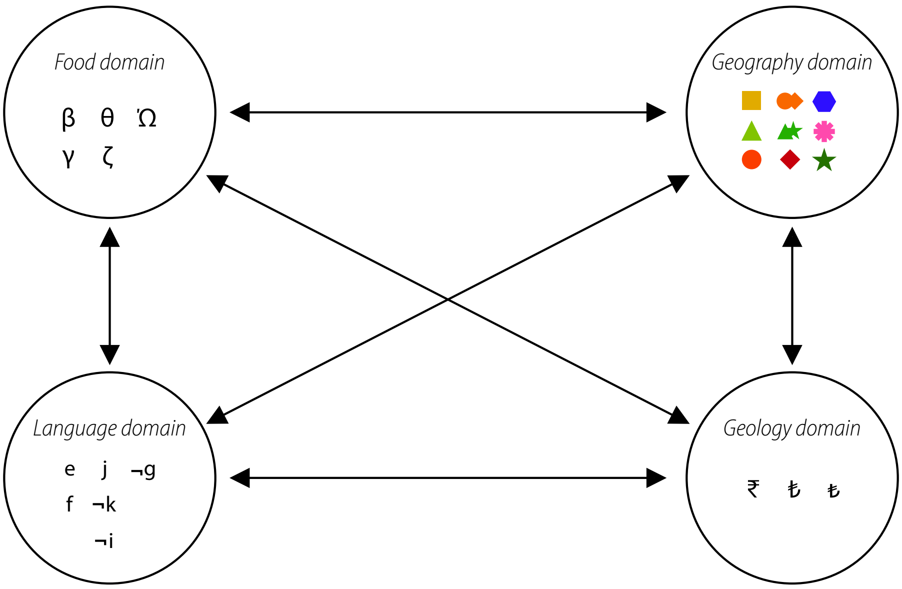


Figure 69 – Example of knowledge configuration that would lead to an Euclidean frame

Source: Produced by the author in March 2022

These definitions addresses the spatial side of MIA: how broad the model relations are and their applicability considering a certain context. At a glance, temporal constraints seems simpler to implement, since MIA temporal dynamicity comes from cross-sections of a longitudinal series of events, that is, MIA accepts that time is a limitation for its models, therefore, admits that any analysis will eventually become obsolete. To diminish impact of this time constraint, a cyclic procedure is proposed as shown in figure 70. On clockwise order, each round of phases 1 to 5 represent a full MIA procedure, that needs to be retaken for a model to be considered valid.

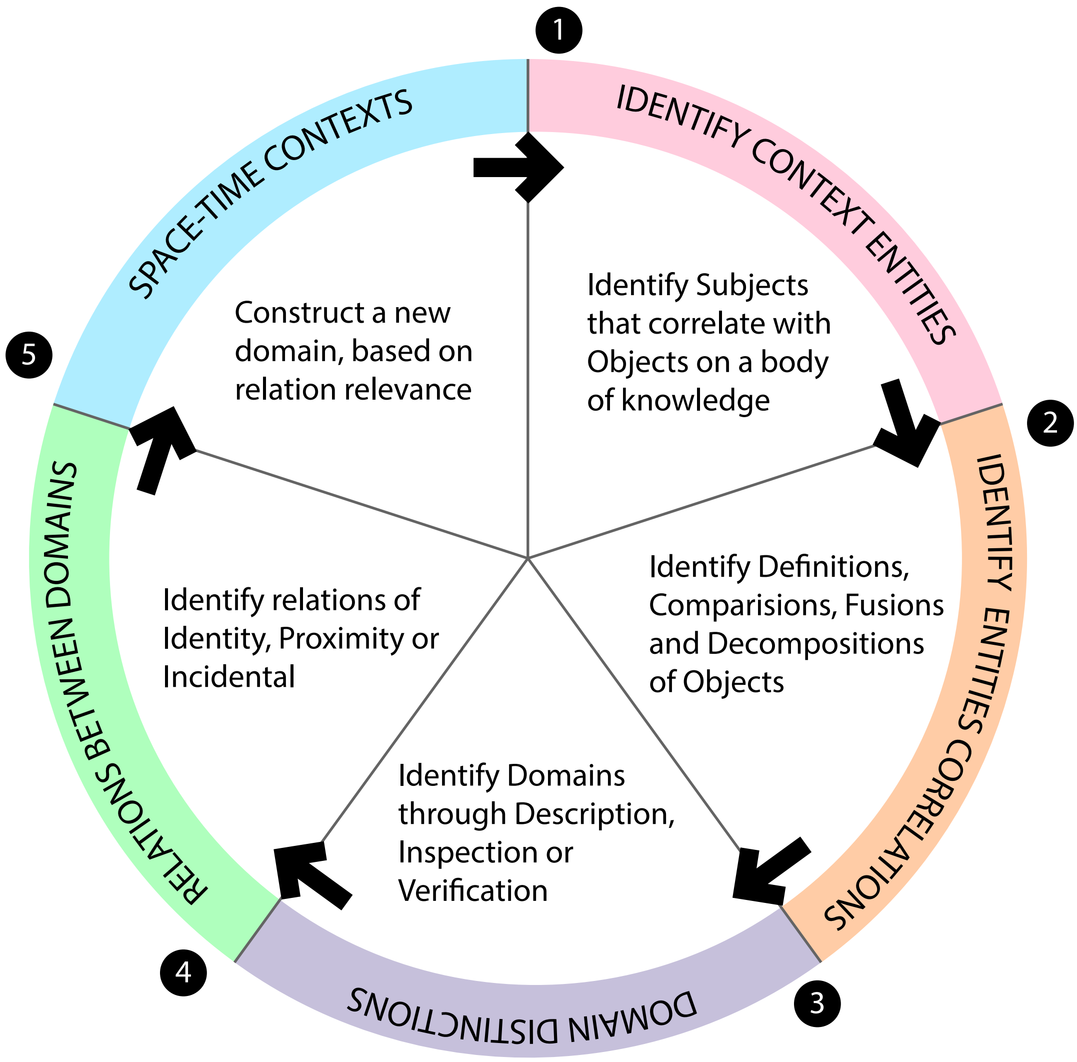


Figure 70 – A cyclic model of MIA procedure implementation

Source: Produced by the author in March 2022

6 Implementing MIA on a NLP problem

Following the methodological path proposed on chapter 3, a technological implementation of chapter 5 has to be presented in order to fulfill Van Gigch and Moigne (1989)'s three levels.

As for regular basis NLP problems, re-addressing Minaee et al. (2021) open questions after 150 Deep Learning models analyzed, only two of them can be treated through MIA: absence of data sets for more complex tasks and commomsense knowledge models. The other two are more related to computer science and ordinary labeling activities.

6.1 Describing the problem

The problem selected for this thesis is a text classification task. Brazil's scientific research, development and innovation — RD&I — policies are based on multiple directives. Between them, there is a particular legislation that deals with projects of RD&I that Brazilian companies undertake with their own resources. Any company can plead for this financial aid, declaring all expenses on their corporate income tax. Part of RD&I expenses are refunded as long as the Brazilian Ministry for Science, Technology and Innovations — MCTI — considers the project as an RD&I-valid project.

Any company can submit their projects for evaluation through an on-line system. Annually, almost 2.500 companies submits more than 10.000 projects. Both qualitative and quantitative information are required. Table 9 presents only the qualitative part required.

Table 9 – Information required for each project

Field description	Expected information	Length (characters)
RD&I activity name	Name of the project	250
Project Description	Resume of the project. Similar to an abstract	4.000
Research objective	Project classification into basic research, applied research or experimental development	2

(Continues...)

Table 9 – ... Continuation

Field description	Expected information	Length (characters)
Project area	Project classification into the following areas: agro-industry, food, consumer goods, cellulose, construction, electronics, pharmaceuticals, finance, mechanics, metallurgy, mining, furniture, paper, petrochemical, chemical, insurance, software, telecommunications, textile, Transport, others	250
Keywords	Words that express what is proposed on the project	250
Technological barrier to be overcome	A specific problem, difficulty, limitation or restriction of technical nature imposed on the development, understanding and implementation of new technologies or new knowledge. All activities carried out to overcome the problem must be of RD&I nature, always presenting results, even if it is an indication that the premise adopted and tested to overcome the barrier should no longer be followed.	4.000

(Continues...)

Table 9 – Conclusion

Field description	Expected information	Length (characters)
Innovative element of the project	The new element must represent scientific or technological progress. Scientific or technological progress is understood as the acquisition of knowledge regarding the understanding of new phenomena (Basic Directed Research); the acquisition of new knowledge, with a view to the development or improvement of products, processes and systems (Applied research); as well as the proof or demonstration of the technical or functional feasibility of new products, processes, systems and services or, an evident improvement of those already produced or established (Experimental Development).	4.000
Methodology or methods	The activities performed, the process used, as well as the skills that were required to implement the project.	4.000
Expected results	What is expected as economical and innovative achievements	250
Complementary information	Any complementary information regarding the previous fields	4.000

Source: Adapted from the system configuration - https://forms.mctic.gov.br/

The MCTI classifies all projects into 16 (sixteen) knowledge areas, adding a general one (Others) if the company considers that no classification suits their intentions:

i. Agroindustry

ii. Chemical

iii. Consumer goods

iv. Construction

v. Electronics

vi. Information and Communication Technologies

vii. Food

viii. Furniture

ix. Mechanics and Transports

x. Metallurgy

xi. Mining

xii. Paper and Cellulose

xiii. Pharmaceutical

xiv. Petrochemical

xv. Textile

xvi. Telecommunications

xvii. Others

The main issue is: what is RD&I, considering that both spatial and temporal variables are always modifying the commonsense of what is or what is not RD&I? Project expenses does not elucidate if the main goal is just a technical problem or indeed addresses and RD&I: it's the project technological barrier, innovative element and methodology that unveils this attribute. Therefore, reading and analyzing all these texts has proven to be a high cost and high complexity task — it requires gathering subjects with specific knowledge on each of the seventeen categories to classify projects into recommended (as being a RD&I activity, considering a particular knowledge area) or not recommended (not being a RD&I activity).

Even though the amount of text to be analyzed can be considered voluminous, the number of labeled instances do not grows proportionally. Considering the past 10 years only 23.738 activities where analyzed. Within this data set it can be noticed a lack of balance between approved (represented on table 10 with values *1*) and non-approved (represented on table 10 with values *0*) activities: 65 percent of them are approved against 35 percent non-approved. If we take the years of 2014 and 2015 separately, the difference grows drastically: 57 percent approval rate on 2014 against 77 percent approval rate on 2015.

Table 10 – Problem domain labeling statistics

Knowledge area	2014				2015			
	0	%	1	%	0	%	1	%
Agroindustry	57	57%	43	43%	62	98%	1	2%
Chemical/Petrochemical	980	63%	576	37%	733	74%	260	26%
Consumer goods	786	64%	440	36%	88	10%	800	90%
Construction	35	23%	116	77%	44	63%	26	37%
Electronics	386	40%	580	60%	99	13%	665	87%
IT & Comms.	275	29%	689	71%	162	18%	750	82%
Food	529	51%	514	49%	240	26%	671	74%
Furniture	71	61%	46	39%	301	84%	57	16%
Mechanics and Transports	1126	44%	1439	56%	615	36%	1101	64%
Metallurgy	335	42%	455	58%	46	8%	523	92%
Mining	21	7%	296	93%	3	3%	104	97%
Paper and Cellulose	48	24%	149	76%	17	10%	159	90%
Pharmaceutical	399	44%	500	56%	9	2%	557	98%
Textile	35	74%	12	26%	1	10%	9	90%
Telecommunications	16	18%	71	82%	12	23%	40	77%
Others	1256	46%	1451	54%	427	22%	1511	78%

Source: Produced by the author in May, 2022

It is also noteworthy a remark about the proportion of activities submitted per knowledge area. Even though projects have been submitted to all areas, three of them — *Others*, *Mechanics and Transports* and *Chemical/Petrochemical* — correspond to 50,09% of activities in 2014 and 46,04% in 2015.

Table 11 – Knowledge area distribution per year

2014		2015	
Knowledge area	%	Knowledge area	%
Others	24,25	Others	19,20
Mechanics and Transports	23,27	Mechanics and Transports	17,00
Chemical/Petrochemical	14,11	Chemical/Petrochemical	9,84
Consumer Goods	11,12	IT & Comms	9,04
Food	9,46	Food	9,03
Electronics	8,76	Consumer Goods	8,80

(Continues...)

Table 11 – Conclusion

2014		2015	
Knowledge area	%	Knowledge area	%
IT & Comms	8,74	Electronics	7,57
Pharmaceutical	8,15	Metallurgy	5,64
Metallurgy	7,17	Pharmaceutical	5,61
Mining	2,88	Furniture	3,55
Paper and Cellulose	1,79	Paper and Cellulose	1,74
Construction	1,37	Mining	1,06
Furniture	1,06	Construction	0,69
Agroindustry	0,91	Agroindustry	0,62
Telecommunications	0,79	Telecommunications	0,52
Textile	0,43	Textile	0,10

Source: Produced by the author in May, 2022

Starting from the same data sets provided, the objectives to be achieved through MIA-based data pre-processing will be:

a. To find domain grouping configurations that increase the accuracy of the NLP algorithm without technical-computational interventions (based on source code changes or any technological procedure for data enrichment);

b. Identify domains that present data with higher or lower learning extraction potential.

6.2 Model selection

Following the design proposed on section 5.2, first decision to be made concerns PTM selection according to section 5.1. According to the review of this section, PTMs are divided into two generations: *first-generation PTMs* and *second-generation PTMs*. As technological development takes a critical part on NLP tasks, only second-generation PTMs are considered. The main choices on second-generation PTMs are CoVe, ELMo, OpenAI GPT and BERT.

Recently, Souza, Nogueira and Lotufo (2020) developed a BERT adaptation for brazilian portuguese named BERTimbau. The corpora used was Filho et al. (2018)'s brWaC, which was based on Bernardini, Baroni and Evert (2006)'s WaC — Web-As-Corpus — methodology. The language scope basis is comparable to other WaC corpus as demonstrated on table 12, particularly with CETENFolha, which is another brazilian portuguese corpora:

Table 12 – Filho et al. (2018) brWaC size comparison with other corpora

Corpus	#Documents	#Tokens	#Types
frWaC	2.20mi	1.02bi	3.9mi
ukWaC	2.69mi	1.91bi	3.8mi
brWaC	**3.530mi**	**2.68bi**	**5.8mi**
CETENFolha	340k	33mi	357k

Source: Filho et al. (2018)

As cited on PTMs extensions, initial language models usually are assembled with various semantic domains. This domain independence is also noticed on brWaC: between the 100 biggest contributors of the initiative, almost 22 domains where involved with 70.348 documents. The whole corpus includes 3.53 million documents. Top 100 contributors distribution is shown on table 13

Table 13 – Filho et al. (2018) brWaC top 100 contributors annotated categories

Category	# of Contributors
News / Weather / Information	20
Arts & Entertainment	10
Education	8
Sports	8
Hobbies & Interests	7
Technology & Computing	7
Health & Fitness	6
Law, Government & Politics	4
Business	3
Style & Fashion	3
Uncategorized	3
Home & Garden	2
Non-Standard Content	2
Careers	1
Illegal Content	1
Personal Finance	1
Real Estate	1
Shopping	1
Travel	1
Video & Computer Games	1
Web Search	1

(Continues...)

Table 13 – Conclusion

Category	# of Contributors
World Football / Soccer	1

Source: Filho et al. (2018)

Quality control was also addressed through filters of size (smaller than 256 characters or bigger then 1MB), non-target content (HTML codes, headers, footers and advertisements), density of stop-words (prepositions, articles and other high density connectors) and content duplicity. Only 5,6% of the original seeds were selected.

6.3 Pre-conditioned simulation

Following the methodological path of the adopted Research Method, to enable a pre-conditioned test, a *raw-data set* of texts and their classification must be acquired, based on database entries coming from projects descriptions as previously described on Table 9. Not all fields in the database are relevant to the research goal, therefore, on the *raw-data set* formation only descriptive information about projects relevance for RD&I where considered, adding the target variable named *specialist advice*, which has a boolean value. A 5-column data set was produced, with the configuration shown on Table 14:

Table 14 – Raw-data set configuration

Knowledge area	Barrier	Element	Method	Specialist advice
Assigned knowledge area	Declared techno-logical barrier	Declared innovative element	Declared Method	Approved/Non-approved

Source: Produced by the author in August, 2022

6.3.1 Model instancing and data pre-processing

In order to isolate effects of MIA modeling on the problem, a simple out-of-the-box algorithm was used, as cited on Model selection. The base code selected was a *Transformer*-based BERT distribution and Kaggle[1] was used as development environment, in order to accelerate experiment progress and facilitate code version control.

```
# Import BERT/neuralmind
from transformers import BertForSequenceClassification, BertTokenizer,
   pipeline
```

Code Listing 6.1 – Importing basic BERT model

[1] https://www.kaggle.com/

Data pre-processing also was done in the simplest way, with highly known libraries. Other libraries are used for data import, progress bar and data export.

```
# Import auxiliary libraries
import numpy as np
import pandas as pd
import glob
import os
import gc
import torch
from torch.utils.data import Dataset, DataLoader
from sklearn import preprocessing
from tqdm import tqdm
```

Code Listing 6.2 – Importing data pre-processing libraries

As possible, in order to fasten result outputs, GPUs were used, if available.

```
# Configuração da CPU/GPU
device = torch.device("cuda:0" if (torch.cuda.is_available()) else "cpu")
print(torch.__version__)
print("Conferindo a unidade de processamento:", device)

#Additional Info when using cuda
if device.type == 'cuda':
    torch.cuda.set_device(0)
    print(torch.cuda.get_device_name(0))
    print('Memory Usage:')
    print('Allocated:', round(torch.cuda.memory_allocated(0)/1024**3,1), 'GB')
    print('Cached:   ', round(torch.cuda.memory_reserved(0)/1024**3,1), 'GB')
```

Code Listing 6.3 – GPU/CPU setup

Data coming from 2014 and 2015 were treated to meet Table 14 criteria, isolating textual data and the approval value. To get a single input textual variable, columns *Barreira* (Barrier), *Elemento* (Element), *Método* (Method) were concatenated into a single string named *Mérito* (Merit).

```
# data loading from CSV
data2014 = pd.read_csv('../input/entitydomainanalysis/LB-2014-Labels.tsv',
                       sep='\t',
                       engine='python',
                       encoding='latin-1')

data2015 = pd.read_csv('../input/entitydomainanalysis/LB-2015-Labels.tsv',
                       sep='\t',
                       engine='python',
                       encoding='latin-1')
```

```
columns = ['ELEMENTO TECNOLOGICAMENTE NOVO OU INOVADOR', 'BARREIRA OU DESAFIO TENOLóGICO SUPERáVEL', 'METODOLOGIA / MéTODOS UTILIZADOS' ]

data2014['MERITO'] = data2014[columns].astype(str).sum(axis=1)
data2015['MERITO'] = data2015[columns].astype(str).sum(axis=1)

data2014.drop(['METODOLOGIA / MéTODOS UTILIZADOS',
           'PB/PA/DE',
           'CLASSIFICAçãO DE ATIVIDADE ECONÔMICA DA EMPRESA',
           'BARREIRA OU DESAFIO TENOLóGICO SUPERáVEL',
           'DESCRIçãO',
           'ELEMENTO TECNOLOGICAMENTE NOVO OU INOVADOR',
           'ID',
           'NOME DA ATIVIDADE',
           'DATA DE INíCIO / PREVISãO DE TéRMINO',
           'VALOR TOTAL DA ATIVIDADE'], axis = 1, inplace=True)

data2015.drop(['METODOLOGIA / MéTODOS UTILIZADOS',
           'PB/PA/DE',
           'CLASSIFICAçãO DE ATIVIDADE ECONÔMICA DA EMPRESA',
           'BARREIRA OU DESAFIO TENOLóGICO SUPERáVEL',
           'DESCRIçãO',
           'ELEMENTO TECNOLOGICAMENTE NOVO OU INOVADOR',
           'ID',
           'NOME DA ATIVIDADE',
           'DATA DE INíCIO / PREVISãO DE TéRMINO',
           'VALOR TOTAL DA ATIVIDADE'], axis = 1, inplace=True)
```

Code Listing 6.4 – Loading 2014 and 2015 data

Target outputs were then normalized into boolean values [0, 1], and all data were merged into a single dataframe.

```
data2014['APROVACAO'] = data2014['APROVACAO'].apply(lambda x: 0 if x == 'Não' else 1)
data2015['APROVACAO'] = data2015['APROVACAO'].apply(lambda x: 0 if x == '0' else 1)

frames = [data2014, data2015]

dataGeral = pd.concat(frames)
```

Code Listing 6.5 – Treating boolean values and merging data

In order to identify both input data and results obtained while dealing with MIA-modeled or non-MIA-modeled domains, two variables were inserted to control either the input origin and the output label.

```python
# Controls input scope for the model:
# If both years = dataGeral
data = dataGeral

# Labels the output file of the experiment
activeTry = 'rawDataGeral-LambdaCorrection-450-Try1'
```

Code Listing 6.6 – Informational scope variables

For the BERT model to utilize Brazilian Portuguese PTM, it is necessary to formally designate it as the main tokenizer. As stated before, the most suitable option available is BERTimbau by NeuralMind-AI[2].

```python
# Tokenizer
tokenizer = BertTokenizer.from_pretrained('neuralmind/bert-base-portuguese-cased')
```

Code Listing 6.7 – Tokenization

6.3.2 Model configuration

With data pre-processed and PTM instantiated, we proceed to model configuration. The first technological barrier encountered is max length of token processing, that is, with how many tokens can an NLP network can deal at once. The latest state-of-the-art models deals with 512 tokens at max, but setting up to this number exceeded learning session timeout of 12 hours established on Kaggle[3] platform.

To deal with this limitation, all 23.826 instances from the input data set had their tokenized length measured, with the results showed on table 15.

Table 15 – Number of data set inputs according to max token length

Max Token Length	# of instances	% of instances
256	13.070	54,85%
350	21.018	88,21%
384	21.253	89,20%
412	21.364	89,66%
450	21.559	90,48%
512	21.835	91,64%

Source: Produced by the author in August, 2022

Taking into account that either 450 and 512 max length configurations present at least 1.991 instances that would not be fully analyzed (will be truncated to fit max length), a differ-

[2] https://github.com/neuralmind-ai
[3] https://www.kaggle.com/

ence of only 276 instances between them is not relevant. Therefore, the variable MAX LENGTH was set to 450 tokens.

Available data for learning was addressed with a simple train/test split, with training phase divided into pure training (where the algorithm extracts first impressions of the data, fitting the model), validation (where an unbiased evaluation of the fitted model is taken aimimg free paramenters adjustments).

The batch size, or the number of data samples analyzed and predicted before comparing expected and output variables leading to error rate, was set to 4.

```
# Max number of tokens on each analysis
MAX_LENGTH = 450

TRAIN_RATIO = 0.7 # Can vary between 0.7 and 0.8, depending on data sizing
VAL_RATIO = 0.2 # Can vary between 0.2 and 0.15 and 0.1, depending on data
    sizing
TEST_RATIO = 0.1 # Can vary between 0.2 and 0.15 and 0.1, depending on data
     sizing

BATCH_SIZE = 4
```

Code Listing 6.8 – Default experiment scenario configuration

While tokenizing samples of different word lengths and combining them into a batch of 4 samples, differences between each batch iteration must be reduced, in order to assure that the model work evenly whether sample size is shorter or longer then the MAX LENGTH defined. Therefore, PADDING (completing with blank tokens if sample size is shorter then the max) and TRUNCATION (forcing max sample size, removing word tokens that exceed max value) variables were set to TRUE.

On Huggingface's *Transformer* documentation, attention mask is a variable that enables the algorithm to identify padded tokens, therefore, their values should not be accounted when predicting a sample.

```
# 'df_tokenized' is a dictionary with keys ['input_ids', 'token_type_ids',
    'attention_mask']

df_tokenized = tokenizer.batch_encode_plus(data['MERITO'], return_tensors='
    pt', padding=True, truncation=True, max_length=MAX_LENGTH).to(device)

# Position 0 access input_ids        ->  [0, DATA_LEN, MAX_LENGTH] =
    input_ids
# Position 1 access attention_masks  ->  [1, DATA_LEN, MAX_LENGTH] =
    attention_masks
# with STACK, both matrix are "glued" side by side
X = torch.stack((df_tokenized['input_ids'], df_tokenized['attention_mask'])
    , dim=0)
```

```

# Convert Approval/Non-approval variables into tensors
y = torch.Tensor(data['APROVACAO'].to_numpy())

# Dataloader to feed the model during training
class TextDataset(Dataset):

    def __init__(self, X, y):

        # assign features to an attribute
        self.X = X
        # sends features to RAM
        self.X = self.X.to(device)

        # assign target values to an attribute
        self.y = y
        # sends target values to RAM
        self.y.to(device)

        # get Dataset size
        self.len = len(y)

    def __len__(self):
        return self.len

    # Sends INPUT_IDS and ATTENTION_MASK to training instances
    def __getitem__(self, idx):
        return self.X[:, idx], self.y[idx]
```

Code Listing 6.9 – Tokenization

With tokens organized and normalized, train/test split can be executed.

```
# Initiate train, validation and test dataloaders
dataset = TextDataset(X, y)

# Calculate how many samples needs to be assigned to each set
num_train_instances = np.int(np.round(dataset.len * TRAIN_RATIO))
num_val_instances = np.int(np.round(dataset.len * VAL_RATIO))
num_test_instances = np.int(np.round(dataset.len * TEST_RATIO))
print(f"Treino: {num_train_instances}, Val: {num_val_instances}, Teste: {num_test_instances}")

# pytorch automaticc split
train_split, val_split, test_split = torch.utils.data.random_split(dataset, [num_train_instances, num_val_instances, num_test_instances])

# return splits to the pytorch Dataloader that will feed the model
```

```
train_loader = torch.utils.data.DataLoader(train_split, batch_size=
    BATCH_SIZE, shuffle=True)
val_loader = torch.utils.data.DataLoader(val_split, batch_size=BATCH_SIZE,
    shuffle=True)
test_loader = torch.utils.data.DataLoader(test_split, batch_size=BATCH_SIZE
    , shuffle=True)
```

Code Listing 6.10 – Train/test split procedure

For training session configuration, a twenty-epoch round was set with 50 training rounds, 50 samples for validation and a single sample for testing.

```
# Epoch quantity
epochs = 20
# Training rounds per epoch
steps_per_epoch = 50
# Validation samples per epoch
epoch_validation_samples = 50
# Training samples per epoch
epoch_test_samples = 1
```

Code Listing 6.11 – Training set configuration

The setup needed to be as technologically simplistic as possible (considering only algorithmic implementation as technology on this matter), in order to obtain results that would be originated exclusively from the data arrangement. As stated on Model selection, a PTM model of brazilian portuguese was loaded. Only two dropout variables configurations where done aiming to avoid overfitting: one for the attention mask sent by the tokenizer, one for hidden layers that had overfitted.

```
model = BertForSequenceClassification.from_pretrained('neuralmind/bert-base
    -portuguese-cased', attention_probs_dropout_prob=0.5,
    hidden_dropout_prob=0.5).to(device)
```

Code Listing 6.12 – Model import

No fine tunning was set. Loss function selected was CrossEntropy and optimizer was set to ADAM.

```
# Fine tunning. Set to FALSE if training time doesn't compensate.
for param in model.base_model.parameters():
    param.requires_grad = False

# LOSS function type
loss_func = torch.nn.CrossEntropyLoss()

# ADAM optimizer, with no learning rate alteration
optim = torch.optim.Adam(model.parameters())

# Accuracy percentage calculation method
```

```
acc_calc = lambda output, labels : (labels == output.argmax(axis=1)).sum()

# Learning rate decay, to avoid overfitting. If reaches 0.9997, learning
    decays.
scheduler = torch.optim.lr_scheduler.ExponentialLR(optim, 0.9997)

# TRAIN
epoch_metada = []
```

Code Listing 6.13 – Model setup

Start training and validation session.

```
for i in range(epochs):

    num_train_examples = 0
    num_val_examples = 0

    train_hits = 0
    val_hits = 0

    # TQDN logs
    train_bar = tqdm(total=steps_per_epoch, desc=f"Train", unit= "steps",
   position=0, leave=True)
    val_bar = tqdm(total=epoch_validation_samples, desc=f"Val", unit= "
   samples", position=0, leave=True)
    test_bar = tqdm(total=epoch_test_samples, desc=f"Test", unit= "steps",
   position=0, leave=True)


    # FOR loop in charge of training
    # First attribute (feature) is self.X[:, idx]
    # Second attribute (labels) is self.y[idx]
    for batch_number, (features, labels) in enumerate(train_loader):

        # initiate LOSS
        train_running_loss = 0

        # initiate the Model
        model.train()

        # get all input_id from batch sample
        # get all attention_mask from batch sample
        input_ids, input_masks = features[:, 0 , :], features[:, 1, :]

        # BertForSequenceClassification automatic return
        # valor de LOSS e LOGITS vem da biblioteca

        var_temp = model(input_ids, input_masks, labels=labels.long())
```

```
        loss, logits = var_temp[0], var_temp[1]

        # gradient descent propagation
        optim.zero_grad()
        loss.backward()
        optim.step()

        # LOSS based opmtimzation
        train_running_loss += loss.item()

        # predictions trhough LOGITS softmax, transforming them into normal probability
        softmax_predictions = torch.nn.functional.softmax(logits, dim=1)

        # handles LOGITS probability to acc_calc function and sets train_hits the obtained value
        train_hits += acc_calc(softmax_predictions, labels)

        # Display bar update
        train_bar.update(1)

        num_train_examples += features.shape[0]

        # Scheduler activation after some training performed
        #schedules.step()

        if (batch_number + 1) % steps_per_epoch == 0:
            train_bar.close()
            break


    # FOR loop in charge of validation
    for batch_number, (features, labels) in enumerate(val_loader):
        with torch.no_grad():
            val_running_loss = 0

            model.eval()

            input_ids, input_masks = features[:, 0, :], features[:, 1, :]

            var_temp = model(input_ids, input_masks, labels=labels.long())
            loss, logits = var_temp[0], var_temp[1]

            val_running_loss += loss.item()

            softmax_predictions = torch.nn.functional.softmax(logits, dim=1)
```

```
            val_hits += acc_calc(softmax_predictions, labels)

            num_val_examples += features.shape[0]

            #Update da display bar
            val_bar.update(1)

            # Break after a certain amount of steps in the current epoch
            if(batch_number + 1) % epoch_validation_samples == 0:
                val_bar.close()
                break

    train_acc = torch.true_divide(train_hits, num_train_examples)
    val_acc = torch.true_divide(val_hits, num_val_examples)

    print(f"EPOCH SUMMARY - {i +1} \t Train loss: {train_running_loss} \t Train Acc: {train_acc} \t Val loss: {val_running_loss} \t Val Acc: {val_acc}")

    model.save_pretrained(f'epochThirdTry_{i}')
```

Code Listing 6.14 – Training and validation session

Start test session.

```
num_test_examples = 0

train_hits = 0
test_hits = 0

test_running_loss = 0

for batch_number, (features, labels) in enumerate(test_loader):
    with torch.no_grad():
        test_running_loss = 0

        model.eval()

        input_ids, input_masks = features[:, 0, :], features[:, 1, :]

        var_temp = model(input_ids, input_masks, labels=labels.long())
        loss, logits = var_temp[0], var_temp[1]

        test_running_loss += loss.item()

        softmax_predictions = torch.nn.functional.softmax(logits, dim=1)
        test_hits += acc_calc(softmax_predictions, labels)

        num_test_examples += features.shape[0]
```

```python

            test_bar.update(1)

        test_acc = torch.true_divide(test_hits, num_test_examples)

        print(f"EPOCH SUMMARY - {i +1} \t Test loss: {test_running_loss} \t
    Test Acc: {test_acc}")
```

Code Listing 6.15 – Test session

6.3.3 Pre-test results

To better observe algorithm achievements after training, each full train/validation/test round was considered to be one experiment. As configured on code listing 6.11, each experiment has 20 epochs. To maximize reliability of results, 10 experiments on each of the thee possible original data arrangements were conducted: 2014 samples; 2015 samples and; 2014 and 2015 samples together. For each set, two variables were observed. ***Loss*** represents the difference between expected results and obtained results. This value is used for weight adjustment, which makes it possible to advance in learning throughout the experiment. Lower loss values indicate better network learning. ***Accuracy*** (acc) represents the percentage of correct answers obtained in each stage of the experiment. This variable represents model assertiveness given the input data.

To guide and facilitate results analysis, an average of training and validation sessions where calculated as a final value for the experiment. Individual results of each experiment can be found on Results on 2014 data, Results on 2015 data and Results on both 2014 and 2015 data respectively. On tables 16, 17 and 18 are presented final values of experiments within the three data scenarios, along with the final average result obtained.

Table 16 – Average results of training rounds for non-treated domain – 2014

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
1	0,7583105	53,93%	0,6966473	56,08%	0,8477135	55,06%
2	0,6941085	53,33%	0,6956549	55,10%	0,8298105	50,84%
3	0,6852451	52,03%	0,7175427	52,50%	0,7677265	54,04%
4	0,6929408	56,35%	0,6832065	54,38%	0,4169658	55,06%
5	0,7111670	54,63%	0,6851374	55,55%	0,8090266	58,78%
6	0,6795197	52,25%	0,6919467	54,18%	0,5093285	57,76%
7	0,6980966	53,85%	0,7164784	51,90%	0,6540617	57,90%
8	0,7352725	51,63%	0,7037263	53,35%	0,7235230	48,94%
9	0,6927016	53,93%	0,6906176	53,35%	0,6860660	52,88%
10	0,7404455	53,63%	0,6685302	58,83%	1,1722298	56,59%

(Continues...)

Table 16 – Conclusion

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
Avg	0,7087808	53,55%	0,6949488	54,52%	0,7416452	54,79%

Source: Produced by the author in August, 2022

Table 17 – Average results of training rounds for non-treated domain – 2015

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
1	0,6035555	76,50%	0,5704024	76,78%	0,4475300	80,38%
2	0,5368777	75,78%	0,6277597	66,43%	0,6394976	73,44%
3	0,5750452	76,00%	0,5308971	78,03%	0,4042923	79,29%
4	0,5191558	76,90%	0,5723158	76,05%	0,2344655	80,28%
5	0,5927629	75,78%	0,5697187	69,75%	0,2924765	76,71%
6	0,5454365	77,53%	0,5117102	76,45%	0,2730931	79,48%
7	0,5748496	75,68%	0,6097252	71,38%	1,1172572	72,55%
8	0,4515269	76,45%	0,6317885	75,00%	0,4269388	72,05%
9	0,6547995	76,50%	0,5721629	75,13%	0,2561494	75,82%
10	0,5733349	76,73%	0,5123417	76,40%	1,2919983	80,67%
Avg	0,5627345	76,38%	0,5708822	74,14%	0,4740491	77,57%

Source: Produced by the author in August, 2022

Table 18 – Average results of training rounds for non-treated domain – 2014 and 2015

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
1	0,7191852	61,85%	0,6853430	62,73%	0,4542553	65,93%
2	0,5353123	63,13%	0,7160808	53,88%	0,7725949	42,59%
3	0,6140570	63,38%	0,6674297	54,55%	0,5456628	63,91%
4	0,7100342	62,03%	0,6810634	58,40%	0,6563075	57,66%
5	0,5968441	63,90%	0,6862518	59,78%	0,6960490	59,04%
6	0,6500738	62,30%	0,6329108	61,13%	0,6772864	57,74%
7	0,6376591	63,63%	0,6471805	64,90%	0,8221304	66,39%
8	0,6674635	65,30%	0,6985908	56,45%	0,6876231	36,42%
9	0,6674635	65,30%	0,6985908	56,45%	0,4020247	66,51%
10	0,6651800	63,08%	0,6515664	62,58%	0,4284774	65,97%
Avg	0,6463273	63,39%	0,6765008	59,08%	0,6142412	58,22%

Source: Produced by the author in August, 2022

Best results were obtained with isolated 2015 data, both on average loss and accuracy. 2014 data has notably the worst results and gathering both data sets pulls results towards 2014's results instead of 2015's ones.

6.4 Applying MIA

As defined on the adopted Research Method, the post-conditioned test needs to apply an organization method at the informational context to be processed. On MIA contributions on Text Classification, a five-step procedure was proposed in order to obtain a new information configuration. On this section all five steps are followed and described.

6.4.1 Step 1: Identify context entities

First step to transform the information environment in question is identifying entities from each original context. Active subjects (natural persons) in the initial configuration analyze texts submitted according to 16 knowledge areas and classify them as approved or non-approved. As the classification is given through judgement of several individuals, when applying Kuroki Jr. (2018) MIA, the set of knowledge expressed in each area can be considered as a subject, thus obtaining 16 subjects.

Reflexively, the corpus of objects is also defined by this distinction of subjects, given that there is a semantic agreement between people who analyzed texts in each area. Difference resides in the fact that each knowledge area has a binary value — Approved or Non-approved — with 3 semantic groupings — Innovative Element, Technological Barrier and Methodology — resulting in 96 semantic contexts. In this sense, given that objects are expressed through attributes, only nouns are eligible as entities, considering their ability to absorb attributes through other semantic terms that modify them. To perform such extraction, three data pre-processing operations were taken: normalization, lemmatization and stop words cleanse.

Text normalization is used to reduce noise on user-generated content. Deviations from standard language and that should be normalized include spelling errors, abbreviations, mixed case words, acronyms, internet slang, hashtags, and emoticons (BERTAGLIA; NUNES, 2016, p.112). The selected library to perform such task was Bertaglia and Nunes (2016)'s Enelvo[4]

Text lemmatization is used to obtain a word's root form, removing inflections and also classifying the non-inflected form into morphological classes. For this step of processing Stanford's Stanza was selected. For stop words cleanse, NLTK was used as basis.

Code listing 6.16 presents all libraries. Full code can be seen on appendix Step 1 Code Listing - Text Normalization and Lemmatization. Figure 71 shows the amounts obtained by each of the 96 context for 2015.

```
# install enelvo portuguese NLP normalizer
!pip install enelvo

# install stanza portuguese lemmatizer
!pip install git+https://github.com/stanfordnlp/stanza.git
```

[4] https://github.com/thalesbertaglia/enelvo

```python

# import stopwords cleanse enabler
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import string
from string import punctuation
from string import digits
import re

# data handles
import numpy as np
import pandas as pd
from gensim.models import Word2Vec
import torch
from torch.utils.data import Dataset, DataLoader
```

Code Listing 6.16 – Informational scope variables

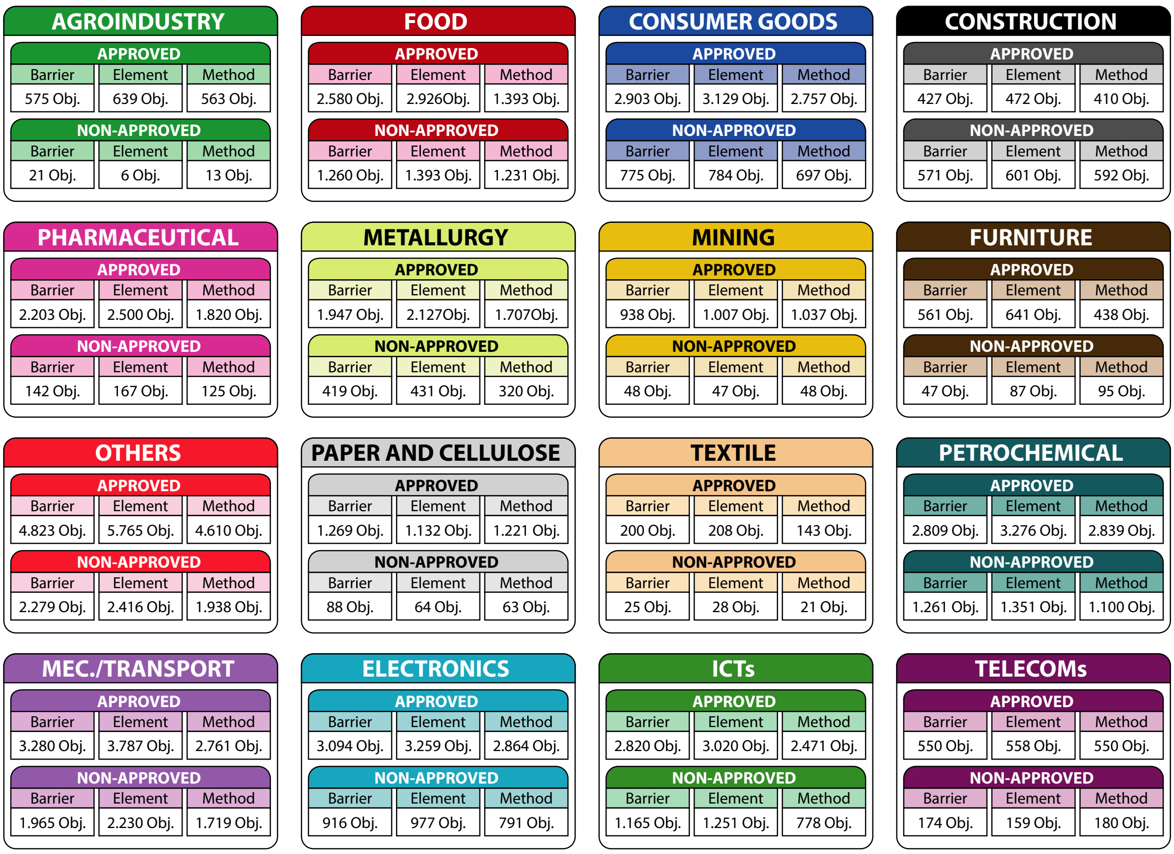


Figure 71 – Objects identified by context – 2015

Source: Produced by the author in September, 2022

6.4.2 Step 2: Identify entities correlations

The second stage for producing a MIA model is identifying correlations between subjects and objects in the domain. For this step, a technique called *Inverse Document Frequency* (IDF), originally proposed by Jones (1973) was used. It is a logarithmic measure of a term relevance considering a set of documents: lower the incidence of a given word in a text, greater the probability of its relevance. Entity selection procedure must identify words that are relevant to the model, maintaining relationship relevance between the potential entity and original context. In this sense, 5 stages of analysis are proposed:

(i) Obtain the IDF value of each entity within each of the 96 semantic domains;

(ii) Calculate IDF average of each entity considering all 96 semantic domains;

(iii) Calculate K value, expressed by the standard deviation of IDF averages;

(iv) Select all entities that IDF value is greater than K value;

(v) Identify objects through **DEFINITION**, **COMPARISON**, **FUSION** or **DECOMPOSITION**.

For 2015, 21.142 potential entities were identified. When applying procedures (i) to (iv) the number decreases to 513. Full results can be seen on appendix Step 2 - 2015 Identified entities throughout experiments. Among the potential entities, semantic sets [method, methodology], [manufacturing, production], [necessary, need], [productive, productivity], [end, final, result], [system, software] were identified. Attributes of these sets were analyzed through COMPARISON, in order to verify the need for DEFINITION of two terms or for FUSION in just one term.

To carry on **comparisons**, objects need to be put side by side in order to compare their attributes. When dealing with word attributes, as stated on the **definition** concept, the context in which a particular word appears will affect its status as object. Therefore, to proceed with the analysis, a valid path would be to obtain relations between potential objects and attributes. Departing from the data treatment through appendix Step 1 Code Listing - Text Normalization and Lemmatization, which has already separated only potential objects, a search for neighbor words was taken using code listing 6.17.

```
# Define all words to be analyzed within any semantic set
entities = {'novo','produto','desenvolvimento','sistema','projeto','
    processo','estudo','qualidade','pesquisa','controle','grande','aplicação
    ','teste','bom','produção','solução','tecnologia','técnico','tipo','
    forma','equipamento','necessário','alto','material','mercado','
    necessidade','possível','uso','melhoria','linha','base','característica'
    ,'dado','utilização','tempo','empresa','análise','desempenho','resultado
    ','tecnológico','estrutura','custo','final','pequeno','etapa','
```

```
    metodologia','principal','resistência','eficiência','cliente','
    conhecimento','método','padrão','problema','tratamento','avaliação','
    segurança','condição','existente','específico','objetivo','baixo','
    capacidade','redução','área','realização','operação','atual','componente
    ','produtivo','temperatura','meio','nível','criação','definição','
    ferramenta','informação','relação','elemento','viabilidade','ambiente','
    risco','água','produtividade','interno','atividade','especificação','
    software','físico','impacto','fabricação','alteração','diferente','parâ
    metro','mecânico','fim','ponto','identificação','campo','capaz','
    trabalho','performance'}

# staging variable to store neighbor words
context_key = {}
def getNeighbours(frases):

        for frase in frases:
                # separates sentences into an array of strings
                text = frase.split(" ")
                unique_set = set(text)

                for i,j in enumerate(unique_set):
                        if i in (0, len(text)-1):
                                continue

                        indices = [i for i, x in enumerate(text) if x == j]

                        contexts = []

                        for index in indices:
                                this_context = []
                                word = j

                                # verify if word is in the target set
                                if word in entities:

                                        # get word before and after
                                        word_before = text[i-1]
                                        word_after = text[i+1]

                                        this_context.append(word_before)
                                        this_context.append(word)
                                        this_context.append(word_after)

                                        contexts.append(this_context)

                                        context_key[j] = contexts
```

Code Listing 6.17 – Neighbor words search aiming semantic set comparision

An example of the results obtained is shown on table 19. All results can be seen on Step 2 - Semantic sets comparision.

Table 19 – Results on Comparision on the semantic set [system, software]

Entity	Sistema (system)	Software
além		x
acoplar	x	
amplo		x
apoio		x
banqueta	x	
companhia	x	
conceber		x
conseguir		x
crítica		x
definição	x	
eficiência	x	
elaboração	x	
ensaio		x
equipe	x	
estampa		x
etapa		x
fase		x
forma		x
garantia		x
gramatura		x
haver		x
hídrico	x	
injeção		x
já		x
layouts		x
lógica		x
máquina	x	
metalúrgica	x	
metodologia	**x**	**x**
net		x
partir	x	
passo	x	
perfil		x
período	x	

(Continues...)

Table 19 – Conclusion

Entity	Sistema (system)	Software
posteriormente	x	
prática	x	
processo		x
produção	x	
produto	x	
produzir	x	
progressivo	x	
realizar	x	
sequencial	x	
ser	x	
sistema	x	
software	x	
solução	x	
teste		x
validar	x	
velocidade	x	

Table 19 – Source: Produced by the author in August, 2022

With the results obtained, similarity percentage was calculated to guide decisions on whether opt to DEFINITION of two objects or FUSION on one term. Observing table 20 compiled percentage, all 513 potential entities were recognized and correlated as objects.

Table 20 – Results on Comparision on the semantic set [system, software]

Semantic set	Similarity percentage	Relation applied
método (method), metodologia (methodology)	3,50%	DEFINITION
fabricação (manufacturing), produção (production)	1,88%	DEFINITION
necessário (necessary), necessidade (need)	7,54%	DEFINITION
produtivo (productive), produtividade (productivity)	12,50%	DEFINITION
fim (end), final (final), resultado (result)	5,06%	DEFINITION
Sistema (system), software (software)	2,00%	DEFINITION

Table 20 – Source: Produced by the author in August, 2022

6.4.3 Step 3: Domain distinction

Once the 16 active subjects and 513 recognized objects of the original domain were identified, informational space configuration can be modified through description, inspection or verification. Since the path to obtain this configuration began with the analysis of a set of texts by natural persons, **verification** procedure seems the most assertive choice for domain distinction. It can be done following 3 steps:

(i) To inquire a group of subjects;

(ii) Identify common attributes;

(iii) Grouping of objects that share common attributes.

First step was carried out prior to the application of MIA, when text data were analyzed by natural persons, that is, it was carried out when the original data set was obtained, classified by knowledge area and approval/disapproval analysis. Second step was carried out in Step 2: Identify entities correlations, where 513 objects recognized by the 16 subjects of the initial context were obtained. To carry out the third stage, four procedures were performed:

(a) Object relevance calculation for the 16 subjects: each knowledge area has a binary value (approved/reproved) for three semantic contexts (Innovative Element, Technological Barrier and Methodology), resulting in six sub-domains. IDF values of each object are transformed into six parameters, obtaining the object's relevance value for each of the 16 subjects. An example of the calculus is shown in table 21. Full results can be checked on Step 3 - Domain distinction. This values represent how relevant objects are for the subjects.

(b) Subject's environment adherence index: endowed with all objects relevance value, the sum of all these values represents the adherence level of the subjects scope of knowledge to the analyzed context, as shown in table 22.

(c) Obtain the informational context dispersion index: given by the standard deviation of the adherence indexes calculated in the previous procedure, as a measure of how uniform the informational environment is.

(d) Domains conception, based on the dispersion index of the informational environment: greater the dispersion index, greater the number of data clusters, observing the need for balance between the subjects' adherence rates to the environment.

Table 21 – Example of object relevance calculation

Entity	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	AGR-TOTAL
Produto	0,736051	0	0,466190	0	0,848768	0	0,34183

Table 21 – Source: Produced by the author in August, 2022

Table 22 – Results of subjects' adherence index calculations - 2015

OTH	**MEC**	**PET**	**CSG**	**ICT**	**FOD**	**ELE**	**MET**
1793,171	1651,568	857,2069	820,3215	763,035	735,793	678,7399	507,0504
PHA	**PAP**	**MIN**	**FUR**	**CNS**	**AGR**	**TEL**	**TXT**
367,3042	141,1532	78,21621	77,86236	49,41759	37,0472	36,2289	3,868743

Table 22 – Source: Produced by the author in August, 2022

Dispersion index calculated based on step "(c)" for 2015 was 562.38, which divides the spectrum of values in table 22 into 4 ranges:

(1) Tier 4, from 0 to 562,38: gathering subjects Metallurgy, Pharmaceuticals, Paper and Cellulose, Mining, Furniture, Construction, Agroindustry, Telecommunications and Textile;

(2) Tier3, from 562,39 to 1.124,76: gathering subjects Petrochemical, Consumer Goods, ICTs, Food and Electronics;

(3) Tier 2, from 1.124,77 to 1.687,14: gathering the subject Mechanics and Transport;

(4) Tier 1, from 1.687,15 to 2.249,52: gathering the subject Others.

The smallest possible level of distinction/aggregation in the informational context, considering all 16 subjects, is the division into two domains. Such distinction must take subjects adherence to the informational context balance into account. Therefore, grouping [1, 4] and [2, 3] are the most balanced, giving rise to:

- **Potential domain 1**, formed by subjects Metallurgy, Pharmaceuticals, Paper and Cellulose, Mining, Furniture, Construction, Agroindustry, Telecommunications, Textile and Others;
- **Potential domain 2**, formed by subjects Petrochemicals, Consumer Goods, ICT, Food, Electronics and Mechanics and Transport.

6.4.4 Step 4: Relationship between domains

With two potential domains found in the previous step, we move on to establishing relationships between the knowledge areas and these domains, as well as between domains themselves. Figure 72 demonstrates Identity and Proximity relationships that originated both potential domains, as well as the extension of relationship between these domains.

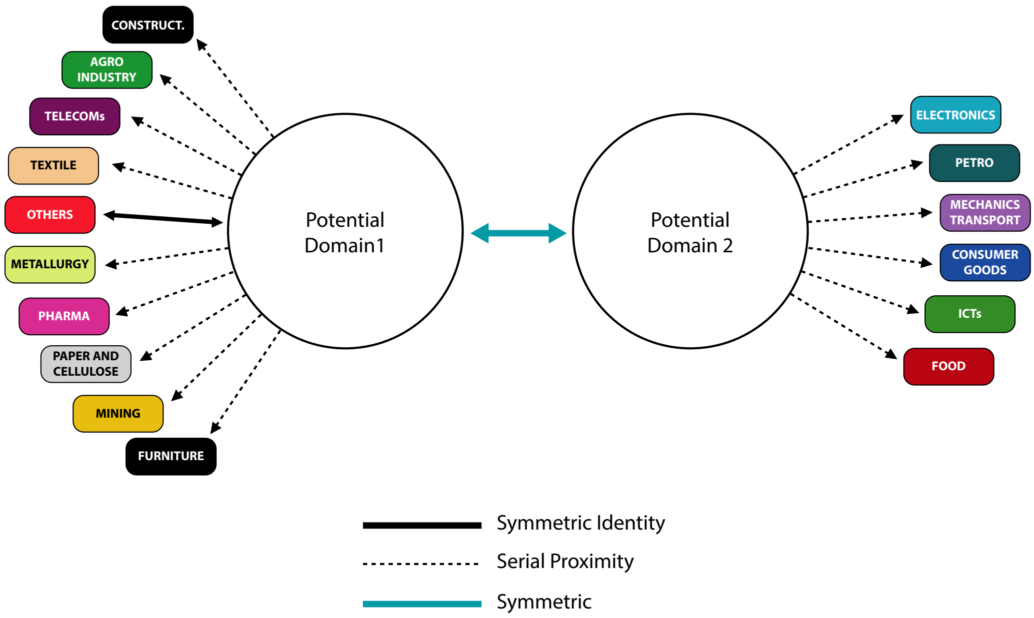


Figure 72 – Relationships between knowledge areas and potential domains – 2015

Source: Produced by the author in August, 2022

Only potential domain 1 presents a Symmetric Identity relation, since the knowledge area “Others” is the only one that has all objects mapped on Step 2: Identify entities correlations. All relationships identified while constructing potential domains 1 and 2 are reflexive, since this operation begins with common objects identification, which necessarily requires checking the existence of the object in the domain itself, and only then proceeding to verify the existence of the referred object in another domain.

Regarding the relationship between potential domains 1 and 2, there is a single symmetric relationship [1,2], given that all objects can be found in any possible configuration of both domains, which demonstrates that both coexist in independently being micro-organizations of the original informational context.

6.4.5 Step 5: Space-time context-based groupings

As described in Step 1: Identify context entities, text data from 2015 were used to conceive the distribution of domains obtained in Step 3: Domain distinction. In order to verify the temporal impact of the proposed architecture over the years, MIA cycle exposed in Figure 3 was applied together with procedures described in items Step 1: Identify context entities to Step 4: Relationship between domains for 2014 data, resulting on a different configuration of domains.

For Step 2: Identify entities correlations, the number of potential entities becomes 480 in 2014, in detriment of 513 obtained in 2015. Subjects adherence rates for 2014 are shown in table 23.

Table 23 – Results of subjects' adherence index calculations - 2014

OTH	**MEC**	**PET**	**CSG**	**ELE**	**FOD**	**ICT**	**MET**
2540,27	2435,69	1400,82	1109,60	911,96	831,62	759,01	736,24
PHA	**MIN**	**PAP**	**CON**	**FUR**	**TEL**	**AGR**	**TXT**
571,51	255,18	162,04	117,65	86,78	59,18	56,29	29,39

Table 23 – Source: Produced by the author in August, 2022

Informational context dispersion index for 2014 was 798,84. Been this value higher than 2015's, it resulted in a slightly different aggregation of subjects:

(1) Tier 3, from 0 to 798,84: gathering subjects Metallurgy, Pharmaceuticals, Paper and Cellulose, Mining, Furniture, Construction, Agroindustry, Telecommunications and Textiles;

(2) Tier 2, from 798,85 to 1.597,68: gathering subjects Petrochemicals, Consumer Goods, ICTs, Food and Electronics;

(3) Tier 1, from 2.396,53 a 3.195,37: gathering subjects Mechanics and Transport and Others.

Most notable changes are: gathering of subjects Mechanics and Transports and Others into tier 1 level (extinguishing tier 2 level); downgrading Information and Communications Technology subject to tier 2, below dispersion index; reordering of subjects on tier 3 level. Although changes are apparently negligible, balance between the subjects' adherence rates must be considered. Therefore, 3 potential domains are proposed for 2014:

- **Potential domain 3**, formed by subjects Mechanics and Transport and part of tier 3 subjects: Agroindustry, Furniture, Paper and Cellulose, Pharmaceuticals and ICTs;
- **Potential domain 4**, formed by subjects Others and the remaining part of tier 3 subjects: Textile, Telecommunications, Construction, Mining and Metallurgy;
- **Potential domain 5**, formed by all tier 2 subjects: Petrochemicals, Consumer Goods, Electronics and Food.

It demonstrates that the problem is highly sensitive to spatial-temporal distinctions: a MIA model used in one year cannot be assumed as applicable to a new temporal context. Confirmation comes when analyzing data from both 2014 and 2015. The number of potential entities identified turn to be 1.192 and subjects' adherence rates are shown in table 24.

Table 24 – Results of subjects' adherence index calculations - 2014 and 2015

OTH	**MEC**	**PET**	**FOD**	**ICT**	**ELE**	**MET**	**CSG**
33507,66	30169,63	18661,63	13322,53	12508,90	12446,05	10095,74	9955,814
PHA	**MIN**	**PAP**	**FUR**	**CON**	**AGR**	**TEL**	**TXT**
8342,05	2865,41	2442,17	1651,90	1452,87	843,38	802,71	289,99

Table 24 – Source: Produced by the author in August, 2022

The informational context dispersion index increased to 10.243,65, creating 3 different potential domains from those previously identified:

- **Potential domain 6**, comprising subjects Mechanical and Transport, Telecommunications, Construction, Paper and Celullose, Pharmaceuticals and Metallurgy;
- **Potential domain 7**, comprising subjects Others, Textiles, Agroindustry, Furniture, Mining and Consumer Goods;
- **Potential domain 8**, comprising subjects Petrochemicals, Food, ICTs and Electronics.

6.5 Post-conditioned simulation

As stated on the attempt of Pre-test results, to produce a predictive model based on untreated data from the selected problem is precarious. Equipped with MIA products obtained from Step 1: Identify context entities to Step 5: Space-time context-based groupings, the resulting model obtained will be validated.

For this purpose, data from 2014 and 2015 were divided and concatenated according to the potential domains 1 to 8 constructed and trained for 10 times each, maintaining all conditions described for Model configuration. Results average of the experiments based on 2015 data can be seen in on tables 25 and 26, while full results are listed on appendix Results on potential domain 1 - 2015 and Results on potential domain 2 - 2015. For 2014 data, results are shown in tables 27, 28 and 29, with full results on Results on potential domain 3 - 2014, Results on potential domain 4 - 2014 and Results on potential domain 5 - 2014. Union of 2014 and 2015 data are shown in tables 30, 31 and 32 which full results can be checked on Results on potential domain 6 - 2014 and 2015, Results on potential domain 7 - 2014 and 2015 and Results on potential domain 8 - 2014 and 2015.

Table 25 – Average results of training rounds for domain 1 – 2015

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
1	0,5129533	78,30%	0,5193378	78,73%	0,3949083	84,91%
2	0,5720047	77,45%	0,5646512	78,05%	0,2623984	86,19%
3	0,5283221	78,90%	0,5502585	78,88%	0,7948045	87,21%
4	0,4973660	78,55%	0,5912255	78,30%	0,4924088	82,10%
5	0,5848956	78,15%	0,4630070	82,80%	0,7062282	83,63%
6	0,4889876	78,35%	0,4742863	84,70%	0,3578030	86,70%
7	0,5133531	79,20%	0,5420564	81,40%	0,7402042	85,42%
8	0,5492220	78,80%	0,4877611	80,30%	0,5696760	83,89%
9	0,5161431	78,63%	0,5493867	78,88%	0,4614289	82,86%
10	0,5335131	77,95%	0,5444802	79,88%	0,1665477	85,93%
Avg	0,5296761	78,43%	0,5286451	80,19%	0,4946408	84,88%

Source: Produced by the author in August, 2022

Table 26 – Average results of training rounds for domain 2 – 2015

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
1	0,5719917	76,20%	0,5885240	72,00%	0,4979948	75,08%
2	0,5965808	74,13%	0,5253601	75,45%	0,3389537	75,89%
3	0,5767568	76,00%	0,5568006	70,28%	0,5497156	62,78%
4	0,4643696	75,38%	0,6077437	72,53%	0,5307578	74,76%
5	0,7093268	75,83%	0,5776335	72,65%	0,4770080	77,35%
6	0,5310474	77,03%	0,5469087	74,88%	0,8312402	75,73%
7	0,5368580	76,03%	0,6264463	74,85%	0,3573535	76,54%
8	0,5086706	76,03%	0,5217224	70,25%	0,6991765	78,96%
9	0,3936843	75,00%	0,5889291	69,60%	0,6196129	53,40%
10	0,6158512	76,10%	0,5772264	75,33%	0,8656888	76,05%
Avg	0,5505137	75,77%	0,5717295	72,78%	0,5767502	72,65%

Source: Produced by the author in August, 2022

Table 27 – Average results of training rounds for domain 3 – 2014

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
1	0,7278106	57,85%	0,6997347	57,40%	0,6521066	61,57%
2	0,7142360	54,75%	0,6806639	59,23%	0,5923843	56,61%
3	0,7017360	55,63%	0,6476322	60,63%	0,6732987	60,33%
4	0,6456280	56,20%	0,6842818	56,43%	0,6183876	57,85%
5	0,7135081	56,95%	0,6649378	54,53%	0,6730917	54,34%

(Continues...)

Table 27 – Conclusion

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	0,6820988	55,33%	0,6894644	55,18%	0,5774552	61,16%
7	0,6770503	56,73%	0,6497662	59,65%	0,8977550	59,30%
8	0,7157383	54,58%	0,6779911	57,83%	0,6763110	61,57%
9	0,7323236	54,10%	0,6602514	60,78%	0,6271839	57,02%
10	0,6963823	56,65%	0,6471352	58,40%	0,5894775	57,23%
Avg	0,7006512	55,88%	0,6701859	58,00%	0,6577451	58,70%

Source: Produced by the author in August, 2022

Table 28 – Average results of training rounds for domain 4 – 2014

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
1	0,7702437	54,10%	0,6811810	54,13%	0,5502236	65,85%
2	0,6474143	54,83%	0,7076121	52,18%	0,5770270	66,10%
3	0,7100286	54,75%	0,6995751	55,85%	0,6441435	52,44%
4	0,7199545	55,50%	0,6678906	60,30%	0,6774329	53,41%
5	0,7096332	55,33%	0,7089876	54,18%	0,7759141	66,59%
6	0,6942910	54,50%	0,7076588	56,23%	0,6276647	50,24%
7	0,7150368	56,58%	0,7157739	55,48%	0,6073157	55,12%
8	0,7171199	54,85%	0,7103200	58,58%	0,2738509	48,54%
9	0,7074180	57,25%	0,7422127	50,03%	0,7784818	43,90%
10	0,7927511	56,38%	0,7025513	51,58%	0,8012449	47,56%
Avg	0,7183891	55,41%	0,7043763	54,85%	0,6313299	54,98%

Source: Produced by the author in August, 2022

Table 29 – Average results of training rounds for domain 5 – 2014

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
1	0,6719953	52,13%	0,6982265	54,55%	0,7081883	50,31%
2	0,6924719	50,23%	0,6900896	51,30%	0,8076079	56,78%
3	0,7435451	51,93%	0,7059076	51,58%	0,5678497	51,15%
4	0,7170149	52,05%	0,6805235	52,45%	0,7613322	54,70%
5	0,7210464	52,10%	0,6842661	50,73%	0,6560815	50,94%
6	0,6738240	52,13%	0,6744902	53,98%	0,9006509	53,03%
7	0,7354406	52,75%	0,6657800	53,55%	0,7866319	50,94%
8	0,7253461	51,15%	0,7265689	51,73%	0,6818504	53,65%
9	0,7005433	51,60%	0,6993661	53,15%	0,6238798	56,99%
10	0,7304046	52,40%	0,7205458	53,48%	0,7831599	49,48%

(Continues...)

Table 29 – Conclusion

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
Avg	0,7111632	51,85%	0,6945764	52,65%	0,7277233	52,80%

Source: Produced by the author in August, 2022

Table 30 – Average results of training rounds for domain 6 – 2014 and 2015

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
1	0,7134288	63,85%	0,6291402	66,30%	0,5748977	63,78%
2	0,6497310	64,08%	0,6412286	64,83%	0,5791090	66,84%
3	0,7368975	63,80%	0,6390414	65,23%	0,6299263	64,29%
4	0,6416462	62,58%	0,6837455	58,25%	0,5149330	67,86%
5	0,6322966	61,33%	0,7066120	65,68%	0,4526347	64,03%
6	0,6815563	63,73%	0,6545747	62,48%	0,5362655	63,39%
7	0,6343259	62,73%	0,6590179	63,23%	0,7549970	62,37%
8	0,6576220	63,88%	0,6629911	61,50%	0,6957849	60,08%
9	0,6757157	64,35%	0,6231856	64,75%	0,5007331	64,54%
10	0,6061178	62,68%	0,6723432	61,75%	0,5938650	62,24%
Avg	0,6629338	63,30%	0,6571880	63,40%	0,5833146	63,94%

Source: Produced by the author in August, 2022

Table 31 – Average results of training rounds for domain 7 – 2014 and 2015

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
1	0,7174812	58,40%	0,6556425	57,23%	0,7341412	48,86%
2	0,6210499	60,45%	0,6849582	54,58%	0,5106040	60,15%
3	0,6371454	60,15%	0,6183966	61,93%	0,7091545	65,74%
4	0,6771975	59,43%	0,6408786	57,85%	0,7229948	56,73%
5	0,7356864	61,13%	0,6719587	53,10%	0,5299214	41,12%
6	0,6575459	60,33%	0,6841224	56,93%	0,7803552	66,37%
7	0,6575248	58,93%	0,6403240	57,50%	0,5849164	62,31%
8	0,6737332	59,18%	0,6752224	57,53%	0,7372152	46,70%
9	0,7160089	60,10%	0,6638499	54,25%	0,7833300	57,49%
10	0,7060478	59,45%	0,6996039	51,00%	0,7952233	46,07%
Avg	0,6799421	59,75%	0,6634957	56,19%	0,6887856	55,15%

Source: Produced by the author in August, 2022

Table 32 – Average results of training rounds for domain 8 – 2014 and 2015

Exp.	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
1	0,6557327	67,18%	0,6688480	64,63%	0,6203640	59,93%
2	0,6259937	66,68%	0,6088732	67,25%	0,6790950	66,46%
3	0,5834717	66,48%	0,7052097	62,93%	0,6292350	68,31%
4	0,5694470	67,78%	0,6009857	65,33%	0,6756247	50,55%
5	0,6668451	66,38%	0,7010962	59,65%	0,6555846	66,83%
6	0,6584263	65,33%	0,6430711	65,25%	0,7236459	64,73%
7	0,6626678	67,65%	0,6890748	57,65%	0,5070481	50,80%
8	0,6041629	67,60%	0,6408207	66,70%	0,6875782	67,08%
9	0,6156710	67,20%	0,6174439	64,45%	0,6469291	57,83%
10	0,6229127	68,80%	0,7270477	59,98%	0,7488835	63,26%
Avg	0,6265331	67,11%	0,6602471	63,38%	0,6573988	61,58%

Source: Produced by the author in August, 2022

6.5.1 2015 domains results analysis

Originally, 2015 obtained the best results as shown on table 17. Top test accuracy value reached 80,67% with an average of 77,57%. Figure 73 compare accuracy results between raw 2015 data and potential domain 1 while Figure 74 compares with potential domain 2.

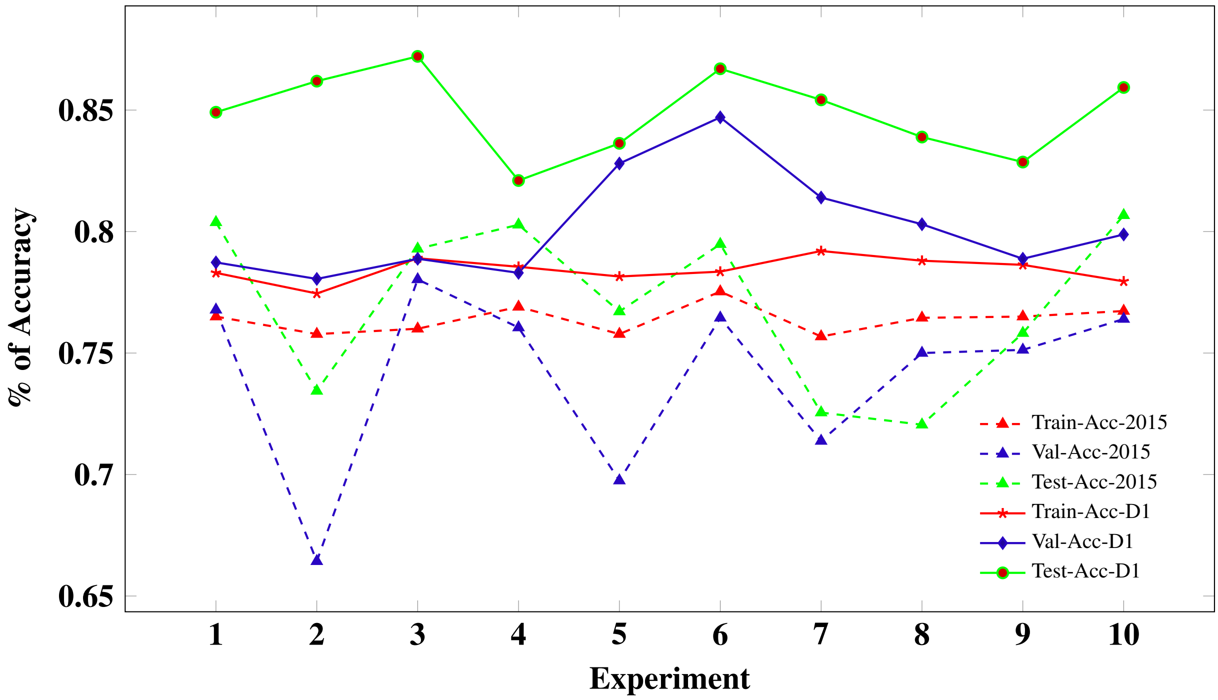


Figure 73 – Accuracy comparative graphic - raw 2015 data and Potential domain 1

Source: Produced by the author in August, 2022

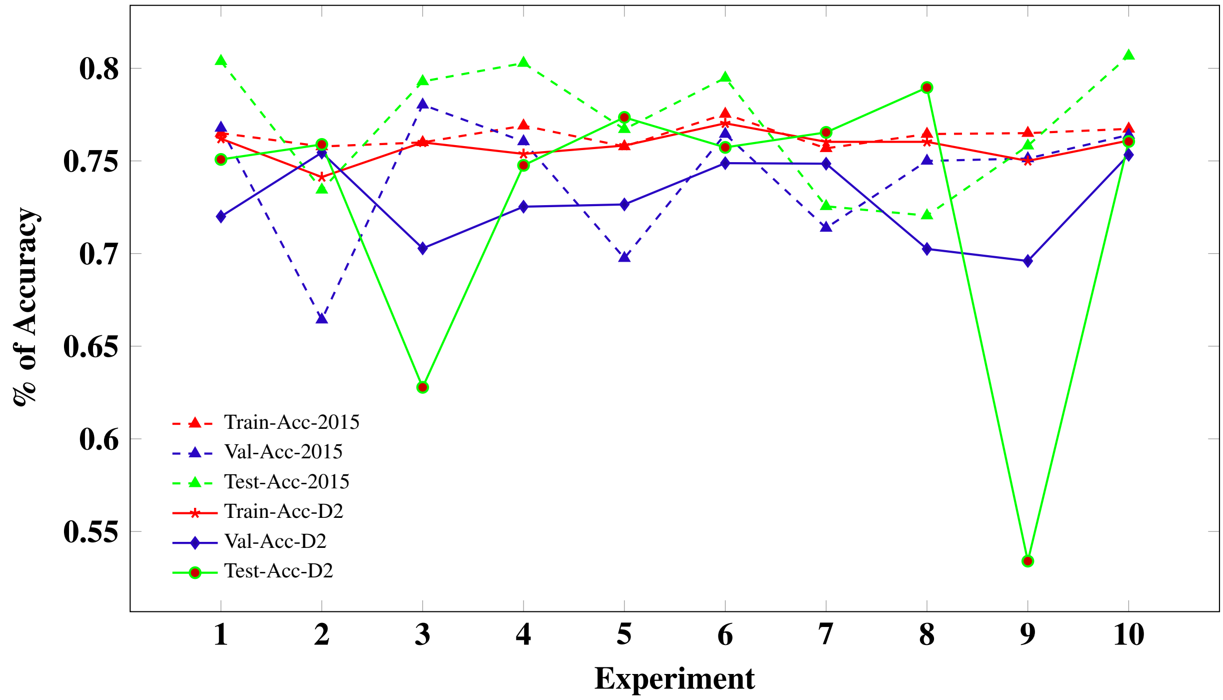


Figure 74 – Accuracy comparative graphic - raw 2015 data and Potential domain 2

Source: Produced by the author in August, 2022

Potential domain 1 presented gain on average results of 7,31%, reaching a max of 87,21% on test accuracy (a 6,54% gain on max). Proportionality between approved and not-approved instances suffer a minimal variation of 4% (originally 72/28, altered to 76/24).

At the other hand, potential domain 2 suffered a decrease of 4,92% on average of results, but only a 1,71% diminish on max accuracy. Approved/not-approved proportionality also suffered a minimal variation, kept on 3% from the original distribution, reaching 69/31.

For loss value, raw 2015 data scored an average of 0,4740491, with lowest value reaching 0,2344655. Potential domain 1 averaged 0,0205917 more (0,4946408), but got a low of 0,1665477, reducing 0,0679178. Potential domain 2 got worsen both average of results, reaching 0,5767502 (0,1027011 more), and lowest value, scoring 0,3573535 with gain of 0,122888.

Through figures 75 and 76 is possible to observe that probability adjustments are better on both cases, considering that neither domains presented any situation where loss value reached values over 1,00, a condition that occurred twice in raw 2015 data simulations.

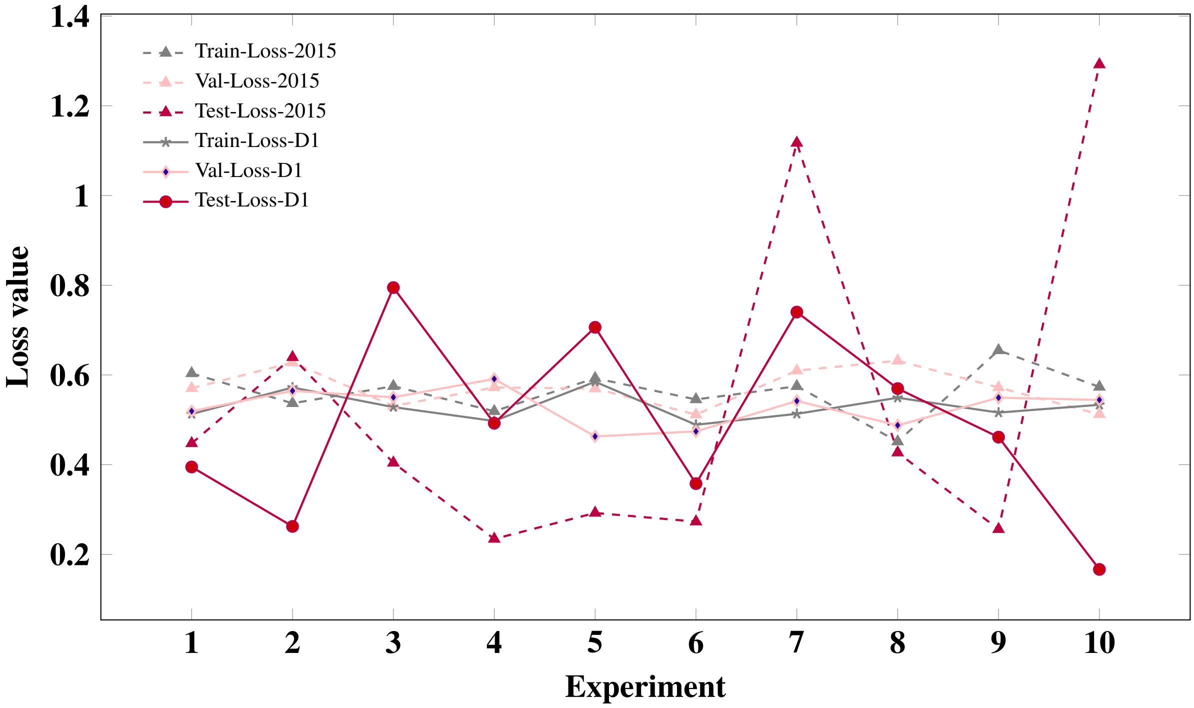


Figure 75 – Loss comparative graphic - raw 2015 data and Potential domain 1

Source: Produced by the author in August, 2022

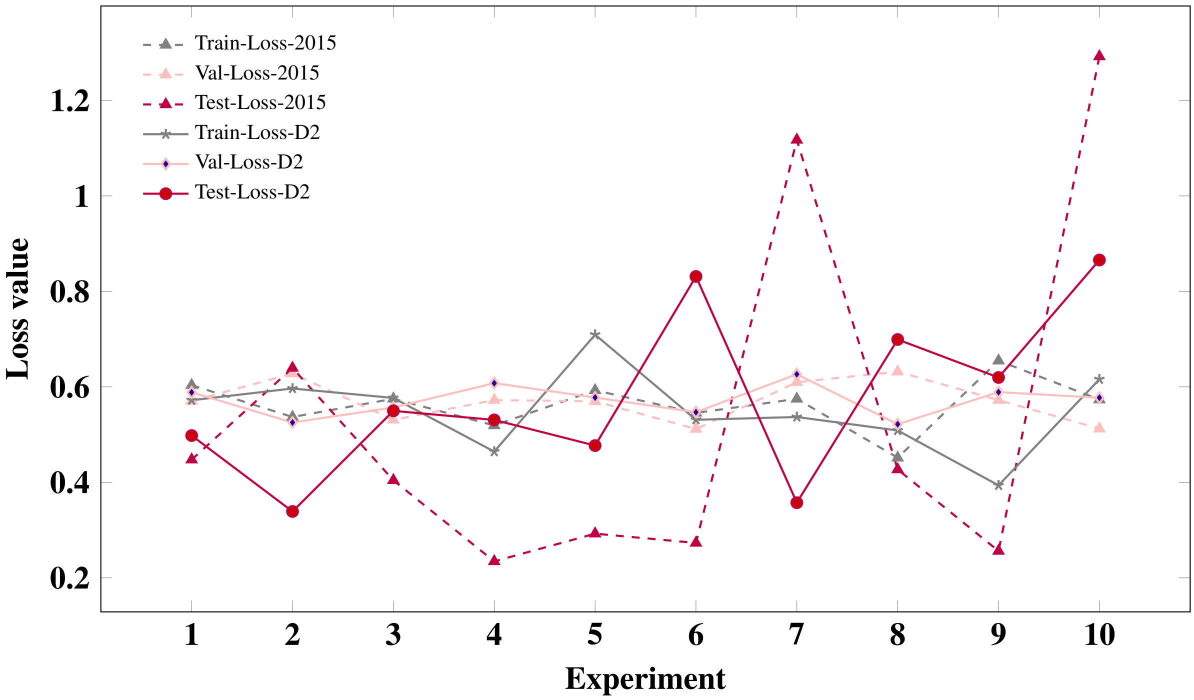


Figure 76 – Loss comparative graphic - raw 2015 data and Potential domain 2

Source: Produced by the author in August, 2022

6.5.2 2014 domains results analysis

Analyzing 2014 data, it presented the worst results on *out-of-the-box* simulations. As table 16 summarized, average result on accuracy was 54,79% with a top score of 58,78%. Figure 77 compares accuracy results between tables 16 and 27 concerning potential domain 3; Figure 78 compares table 16 with 28 on potential domain 4; Figure 79 does the same parallel on tables 16 and 29 for potential domain 5.

Potential domain 3 presented a 4,01% gain on average of results and 2,79% on top score. Original proportion on approved/non-approved instances of 54/46 suffered a minor variation of 2% on ratio, reaching 56/46, therefore, maintaining initial conditions unaltered.

Potential domain 4 presented a minor gain on average of results of 0,19%, which can be considered as irrelevant. At the other hand, top score gain reached 7,81% on experiment 5, with 66,59%. Original proportion on approved/non-approved instances of 54/46 maintained exactly the same ratio.

Potential domain 5 had both average of results diminished (from 54,79% to 52,80%) and top score (from 58,78% to 56,99%). Original proportion on approved/non-approved suffered a shift from 54/46 to 49/51.

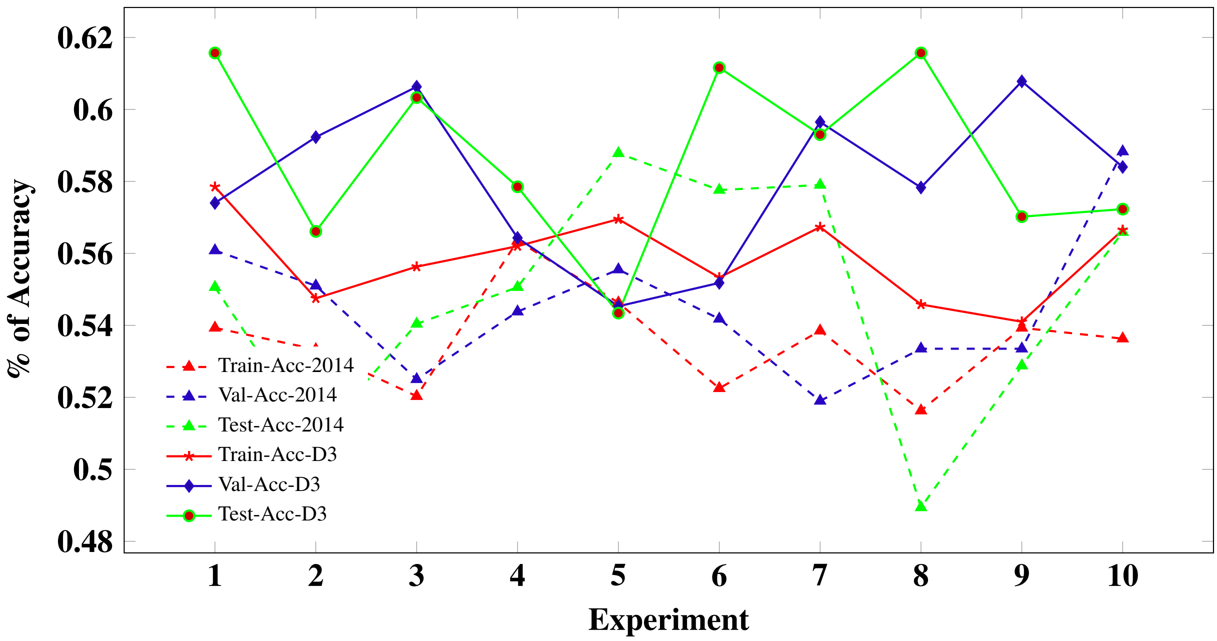


Figure 77 – Accuracy comparative graphic - raw 2014 data and Potential domain 3

Source: Produced by the author in August, 2022

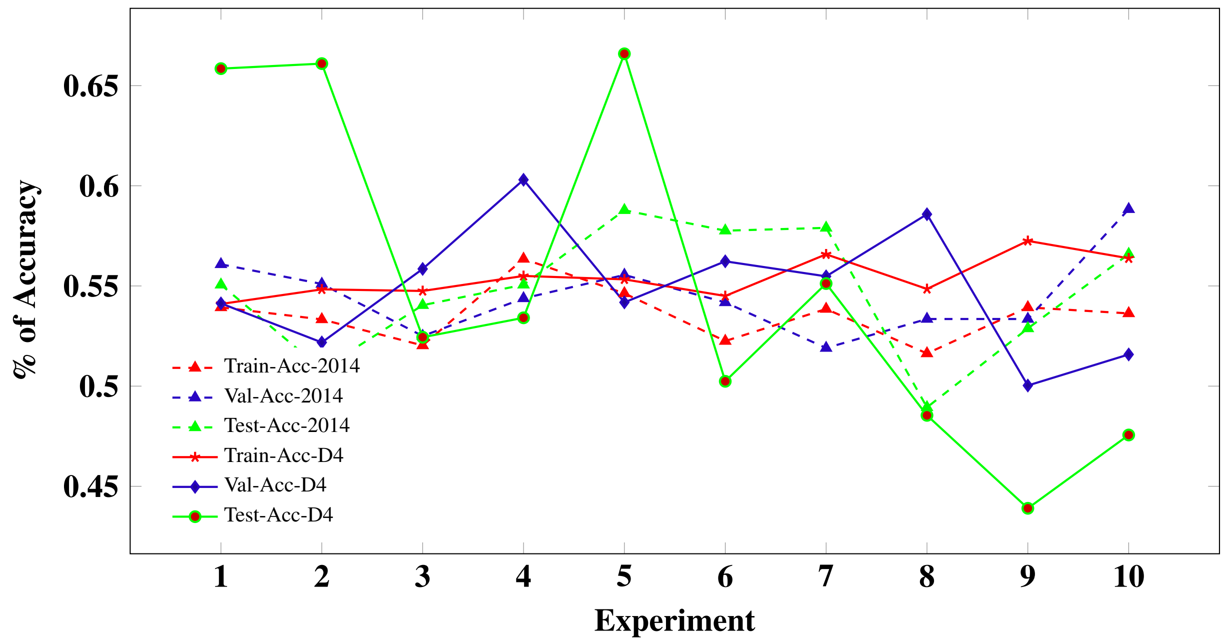


Figure 78 – Accuracy comparative graphic - raw 2014 data and Potential domain 4

Source: Produced by the author in August, 2022

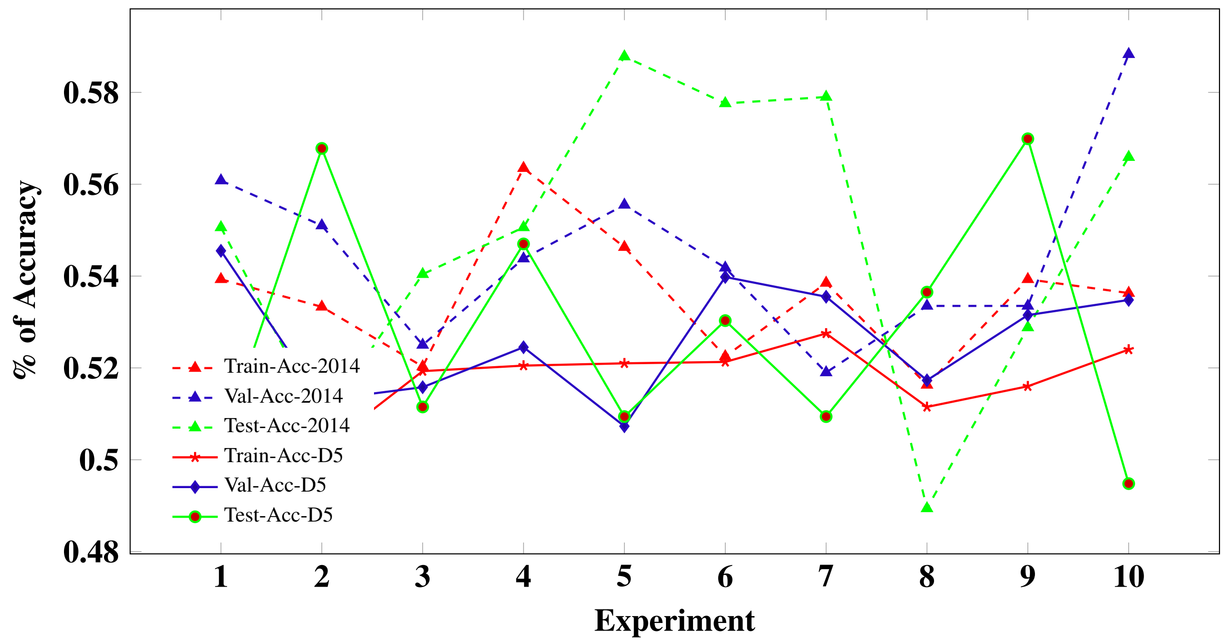


Figure 79 – Accuracy comparative graphic - raw 2014 data and Potential domain 4

Source: Produced by the author in August, 2022

Loss also was not satisfactory on neither Pre-conditioned simulation nor Post-conditioned simulation. On average, test loss on 2014 non-treated domain averaged 0,7416452 with lowest score of 0,4169658 and highest of 1,1722298. For potential domain 3, average loss was

0,6577451 (reduction of 0,0839001) with lower of 0,5774552 and higher of 0,8977550. Potential domain 4 averaged 0,6313299 (reduction of 0,1103153) with results varying from 0,2738509 to 0,8012449. Potential domain 5 averaged 0,7277233 (reduction of 0,0139219) with lower of 0,5678497 and higher of 0,9006509.

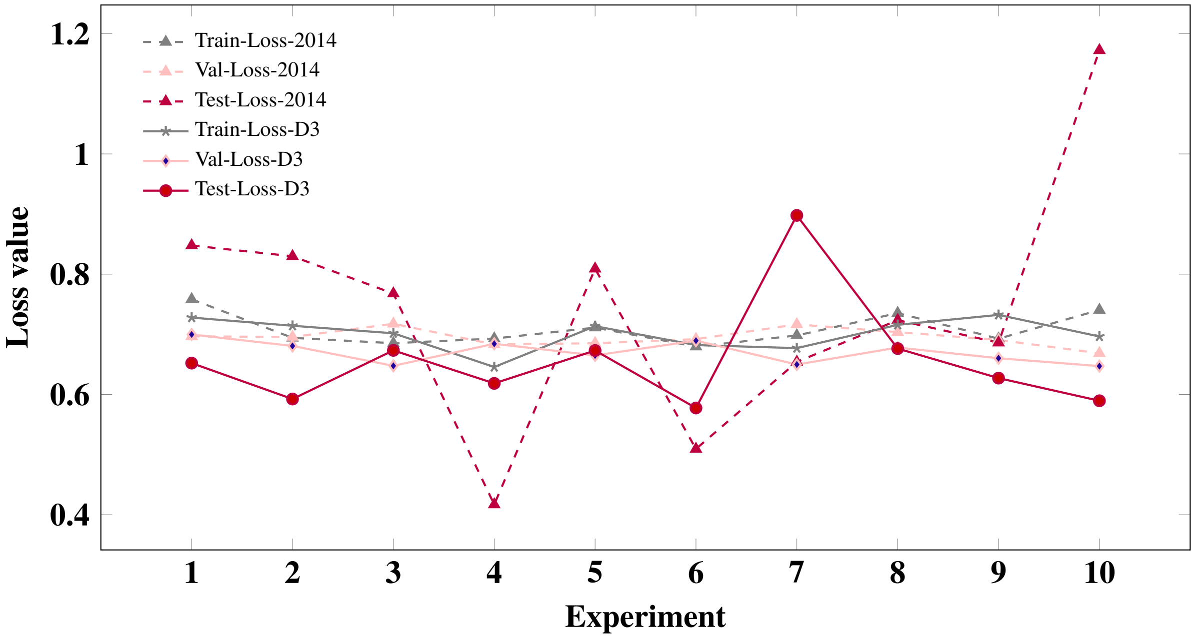


Figure 80 – Loss comparative graphic - raw 2014 data and Potential domain 3

Source: Produced by the author in August, 2022

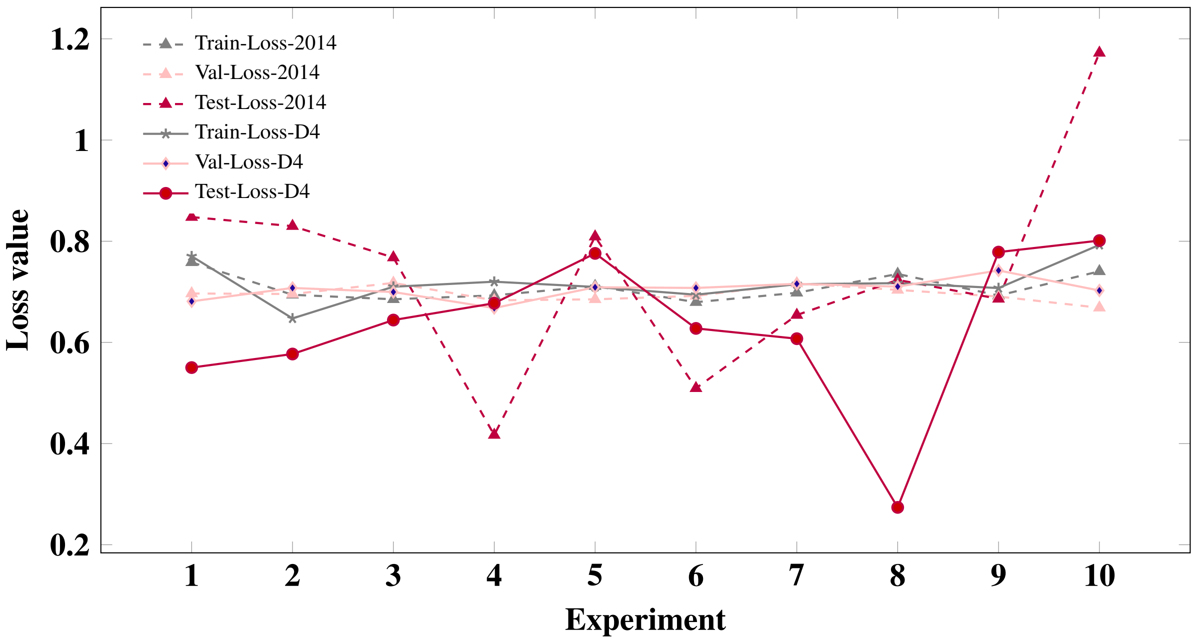


Figure 81 – Loss comparative graphic - raw 2014 data and Potential domain 4

Source: Produced by the author in August, 2022

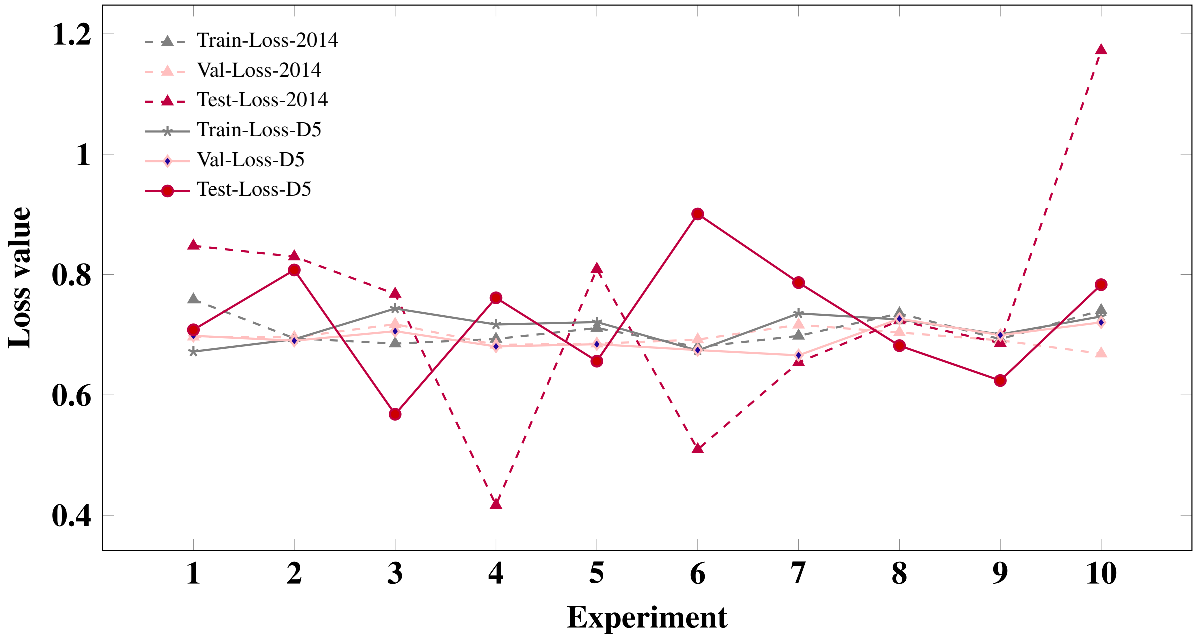


Figure 82 – Loss comparative graphic - raw 2014 data and Potential domain 5

Source: Produced by the author in August, 2022

6.5.3 2014 and 2015 domains results analysis

On Pre-conditioned simulation, when joining data from 2014 and 2015, results tend to get closer to 2014 results ratter then 2015. Potential domain 6 had a gain on average test accuracy of 5.72% (63.94% against 58.22% on raw 2014/2015 data). Approved/not-approved proportionality suffered a 5% variation going from 61/39 to 66/34. Figure 83 shows a graphical comparison of this analysis.

Potential domain 7 got a drop on average test result of 3.07% with a high/low gap of 25.25%. It is important to highlight that on 4 opportunities test accuracy didn't reached 50% (exp.1 with 48.86%, exp.5 with 41,12%, exp.8 with 46.70% and exp.10 with 46.07%),which can be graphically observed on figure 84. On comparative analysis (before MIA and after MIA), this domain can be taken as the worst result overall. Approved/not-approved proportionality suffered no variation, maintaining 61/39 ratio.

Potential domain 8 presented a shy gain of 3.07% on average test accuracy with a lower approved/not-approved variation of 3%, reaching 58/42. Figure 85 presents a plot based on the results.

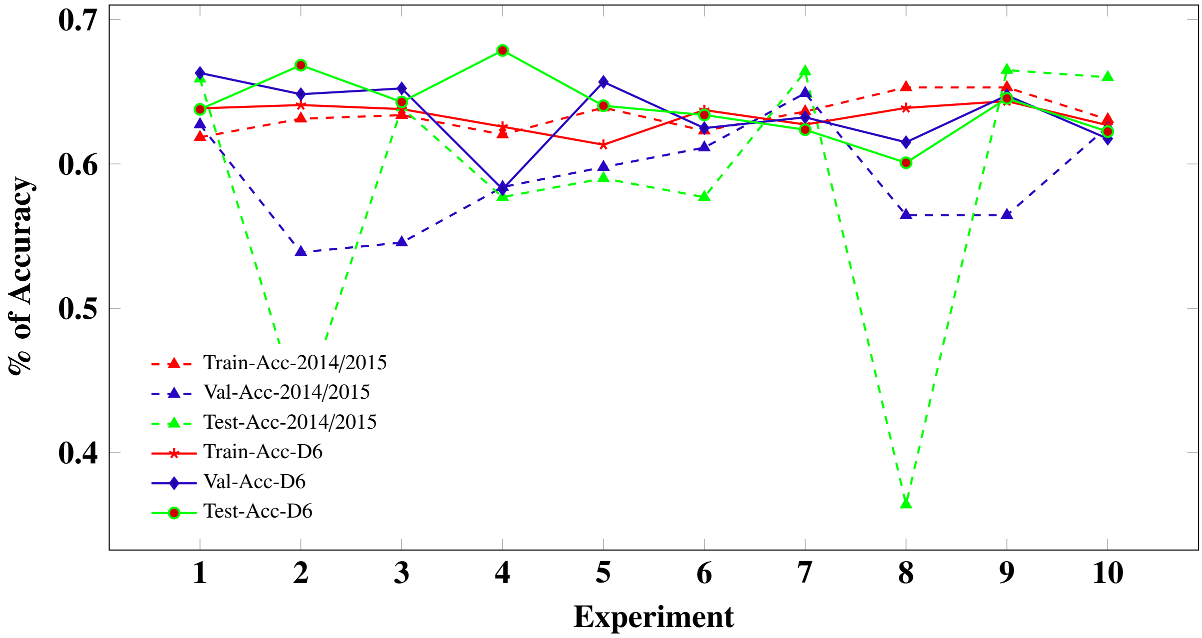


Figure 83 – Accuracy comparative graphic - raw 2014/2015 data and Potential domain 6

Source: Produced by the author in November, 2022

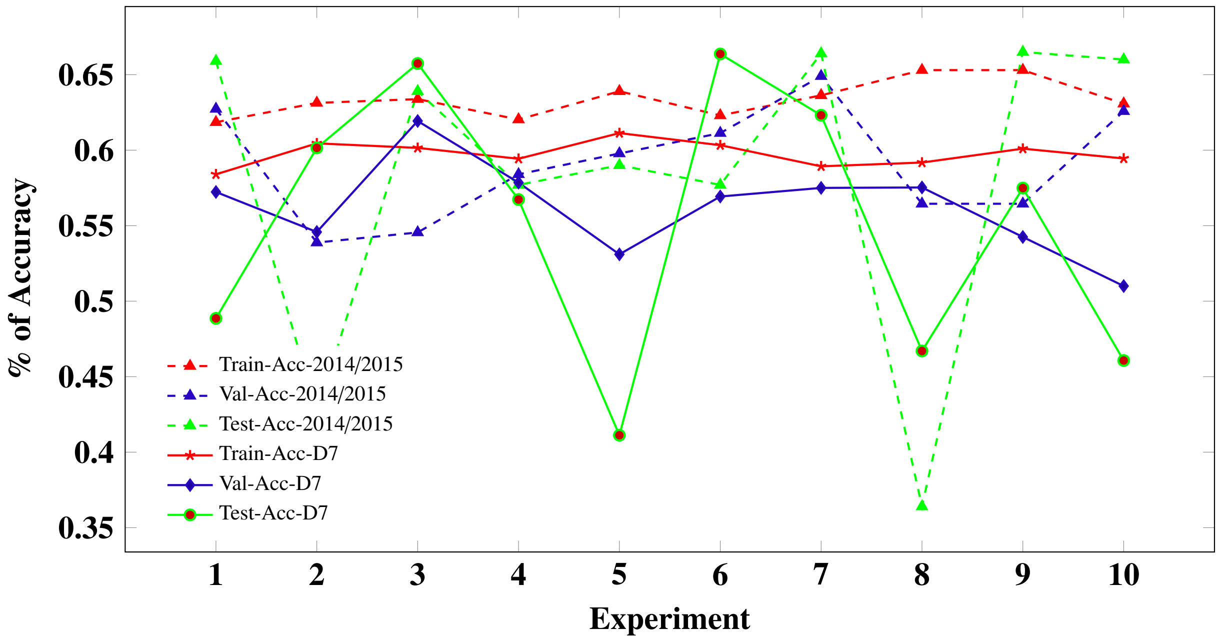


Figure 84 – Accuracy comparative graphic - raw 2014/2015 data and Potential domain 6

Source: Produced by the author in November, 2022

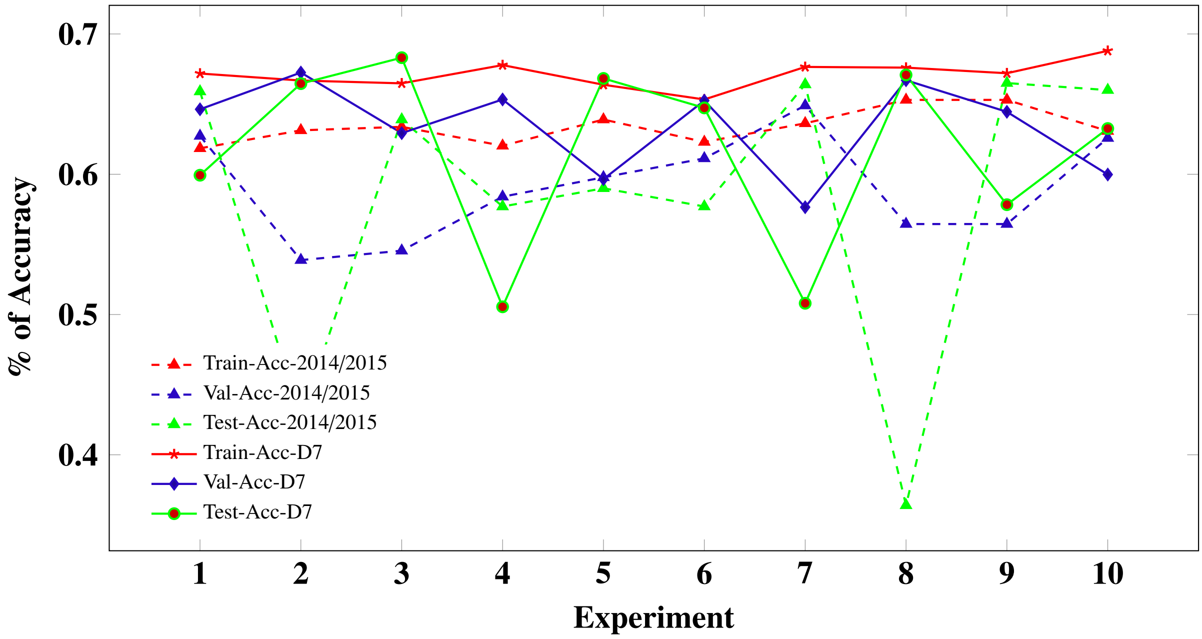


Figure 85 – Accuracy comparative graphic - raw 2014/2015 data and Potential domain 6

Source: Produced by the author in November, 2022

As for loss, only potential domain 6 presented better average of results on test (0.5833146) then Pre-conditioned simulation (0.6142412), being both not satisfactory. Even though potential domain 7 and potential domain 8 presented worst test loss results on average (0.6887856 and 0.7488835, respectively), their peak was lower then the one presented on raw data (0.7952233 and 0.7488835 against 0.8221304). Also, all potential domains scenarios had better high/low loss difference then raw data, as can be seen on figures 86, 87 and 88.

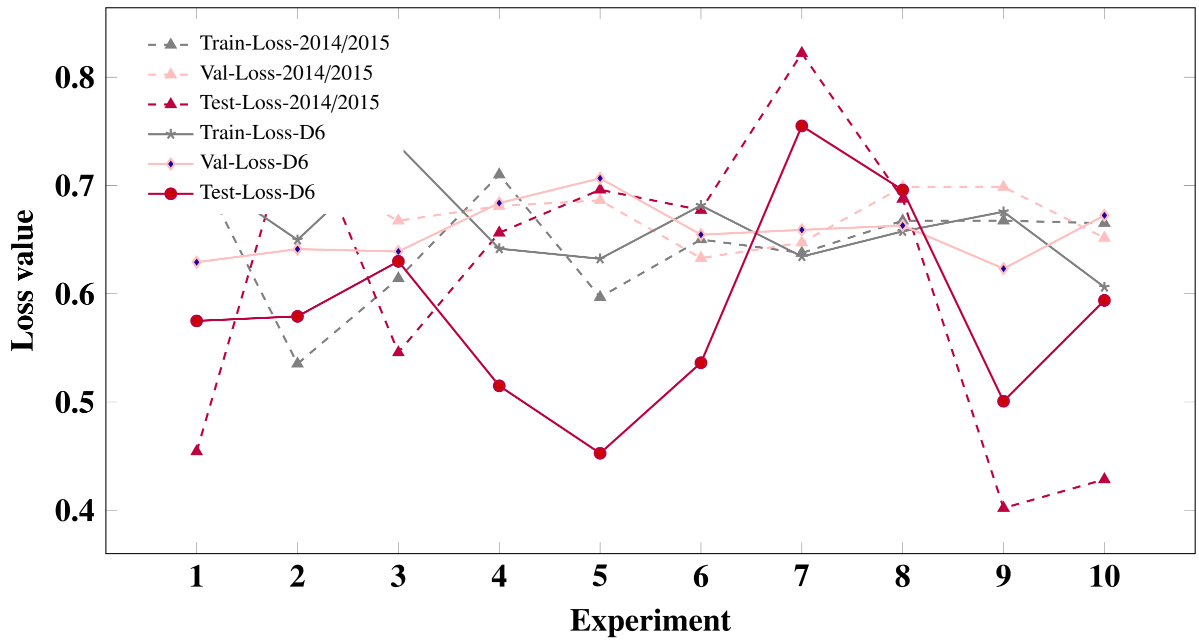


Figure 86 – Loss comparative graphic - raw 2014/2015 data and Potential domain 6

Source: Produced by the author in August, 2022

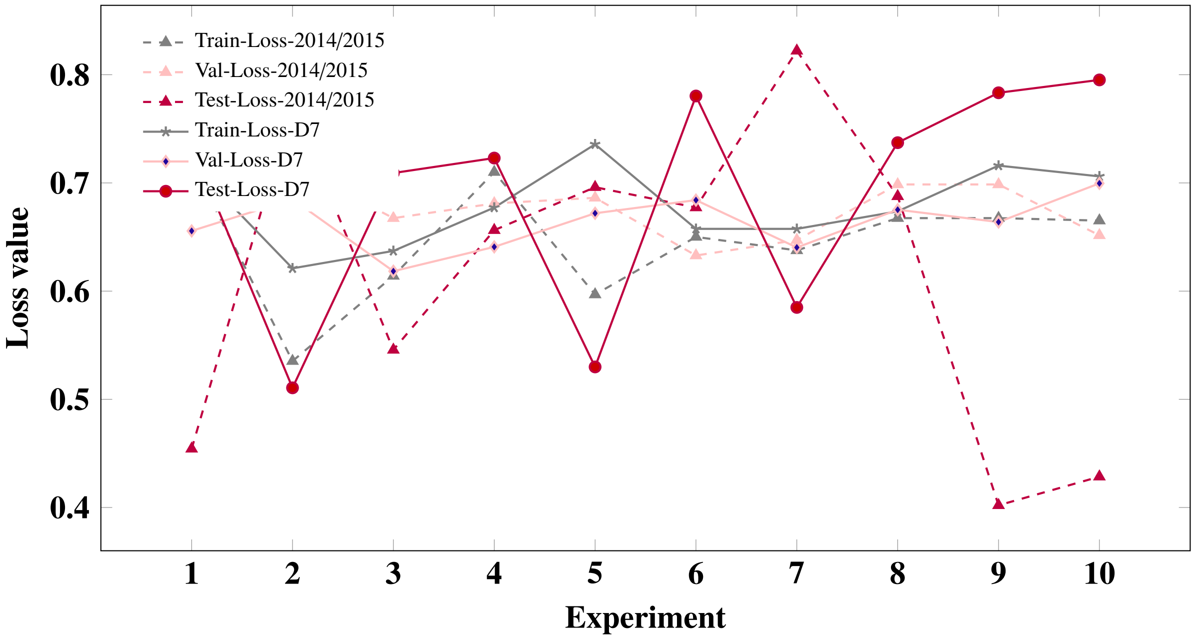


Figure 87 – Loss comparative graphic - raw 2014/2015 data and Potential domain 7

Source: Produced by the author in August, 2022

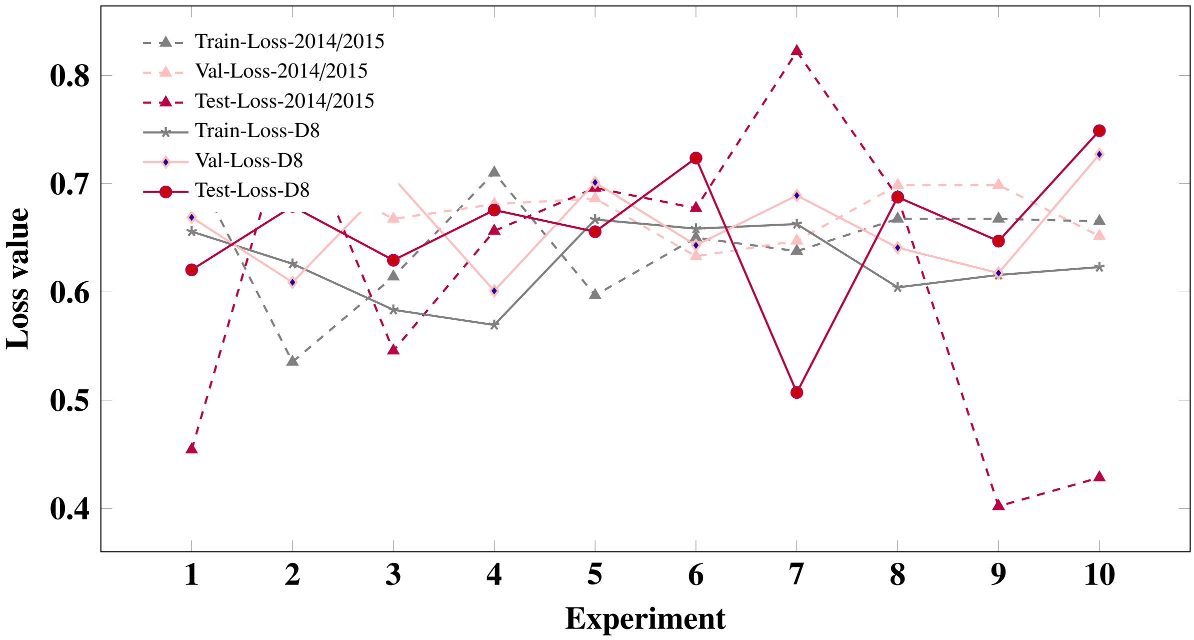


Figure 88 – Loss comparative graphic - raw 2014/2015 data and Potential domain 8

Source: Produced by the author in August, 2022

6.5.4 Learning extraction analysis

Of the 8 proposed domains, taking the test accuracy results as an analysis parameter, 4 (four) showed a gain, 3 (three) showed a loss, and 1 (one) maintained the previous levels, with a small increase. Based on this analysis, table 33 identifies the data sets that have more and less potential for learning extraction.

Table 33 – Analysis of learning potential by knowledge area

Knowledge area	2014	2015	2014/2015	Potential
Agribusiness	1	1	-1	1
Foodstuffs	-1	-1	1	-1
Consumer goods	-1	-1	-1	-3
Civil construction	0	1	1	2
Electro-electronics	-1	-1	1	-1
Pharmaceutical	1	1	1	3
Mechanics and Transportation	1	-1	1	1
Metallurgical	0	1	1	2
Mining	0	1	-1	0
Furniture	1	1	-1	1
Paper and Cellulose	1	1	1	3
Chemical/Petrochemical	-1	-1	1	-1
TICs	1	-1	1	1
Telecommunications	0	1	1	2
Textile	0	1	-1	0
Others	0	1	-1	0

Source: Produced by the author on January 2023

7 Goal achievements

As stated on Research Method, to verify how Multimodal Information Architecture impacts learning results on artificial neural networks dealing with natural language processing problems, a comparative analysis confronting accuracy and loss was proposed. Two scenarios were then assembled: a Pre-conditioned simulation and a Post-conditioned simulation.

To isolate the effects of MIA on both procedures, the key aspects of network architecture and activation function were unaltered as shown on Model instancing and data pre-processing and Model configuration

Pre-conditioned simulation used data analyzed and classified by experts regarding approval trend of each instance considering sixteen knowledge areas.

Post-conditioned simulation rearranged the same data set having as main leading directive a method developed through five steps based on MIA.

Van Gigch and Moigne (1989) served as methodological basis along chapters 4, 5 and 6. On this chapter, all achievements are reviewed and prepared to conclusion.

7.1 Epistemological findings

On Relevant concepts for a theoretical model, a wide and profound research for the origins and development of artificial intelligence went from the initial question raised by Turing (1950) up to Minaee et al. (2021) survey on the most used Natural Language Processing algorithms and techniques through the past years.

Through 56 years of research and development, artificial intelligence grounded its findings on a model of the human brain first attempted by Rosenblatt (1961) through his concept of *Perceptron*. From this fundamental unit, artificial neural networks were constructed, as an arrangement of several *perceptrons*, connected through weighted links, which are activated when mathematical functions obtain a certain value considering an input. Setbacks and development limitations were constant through these years, but generally, efforts addressed two main issues: network architecture and data volume required to achieve satisfactory results on learning, also addressed as data dimensionality by Bellman (1954) and Arel, Rose and Karnowski (2010).

Only with Hinton, Osindero and Teh (2006) algorithm, the architectural part of the problem had an major upside, giving rise to the term Deep Learning. From their work, neural networks could be arranged with more then three layers, overcoming complexity problems first identified with McClelland et al. (1986) and re-addressed by Minsky and Papert (1988).

From the development of more complex networks, it was identified on section 4.5 that two epistemological approaches for NLP were considered: a **Rationalist** one, based on the work of Noam Chomsky (1986), and an **Empiricist** one, which has as main author Zellig Harris (1951). Natural language processing evolved, majorly, through an empiricist approach, according to Manning and Schutze (1999). This point of view was confirmed on section 4.5.3 which described a vast list of NLP implementations, all based on statistical analysis of texts.

As an counter-measure for the exponential growth of data required to deal with more complex problems (on our case, NLP tasks), pre-processing techniques (particularly feature extraction) were widely used on FFNN-based models, RNN-based models, CNN-based models, Capsule Networks. These models aim to map relevant relationship between words considering a certain amount of space (the length of the sentence) or time (word dependence and text structure).

Through Attention mechanisms it was possible to verify that restricting sentence representation vectors to a certain length is not as efficient as letting the model search for relevance on the whole text for itself.

From this idea, Vaswani et al. (2017) proposed an architecture based on attention mechanisms called **Transformers**, which used **Neural GPUs** introduced by Kaiser and Sutskever (2015).

Since 2018 until present time, **PTMs** are widely dominating NLPs initiatives. Since they are pre-trained with considerably larger data sets then previous models, it is natural that their language representation model is far better.

On section 4.6, a short review on Multimodal Information Architecture started to enlighten another path to face data dimensionality, but departing from a different point of view: an architectural ***order*** on ***information*** can be achieved through distinguishing factual perception into multiple ***possible worlds***. These worlds, then, can be observed with a set of *applied rules* and *applied relations*, which together are denominated ***relational models***. Modal logic is then used to bring economy of relations, therefore, avoiding an attempt to portrait full reality, what would be not productive.

The epistemological goal defined on the Research Method is considered to be achieved since:

(a) Artificial Intelligence origin was identified on section 4.1 Intelligence and Artificial Intelligence, reporting McCarthy et al. (2006) first challenge related to algorithms that would simulate man actions and the dialog of ideas between Hebb (1949), Samuel (1959), McCulloch and Pitts (1943), Rosenblatt (1961), McCarthy and Hayes (1969), Minsky (1961) and McCarthy (1981);

(b) Artificial Neural Networks origins were described on section 4.2 Interactions between Agents and Environments, where the genesis of modeling the human brain was deeply discussed between McClelland et al. (1986) and Minsky and Papert (1988);

(c) Artificial Intelligence development was described on section 4.3 Artificial Neural Networks: definitions, development and applications where the realization of Rosenblatt (1961)'s *perceptron* through late 1980's until early 2000's (also challenged by Minsky and Papert (1988)), was reviewed by (FAUSETT, 1994), Hassoun et al. (1995), Basheer and Hajmeer (2000), Engelbrecht (2007), Haykin (2009) and Hagan, Demuth and Beale (2014). On this topic, the term Deep Learning was discovered as an specific implementation of ANNs that overcame *shallow architectures* restrictions;

(d) Deep learning appearance was then reviewed on section 4.4 Deep Learning: concepts and development, beginning from Hinton, Osindero and Teh (2006) as an inflection point from traditional shallow architectures (BENGIO; LECUN et al., 2007). Two main implementations of Deep learning architectures were identified: Convolutional Neural Networks and Recurrent Neural Networks;

(e) Natural Language Processing was reviewed as the main goal on this stage of research. A **Rationalist** and **Empiricist** approaches confronted, leading to eight main implementations of empiricist-based networks: FFNN-based models, RNN-based models, CNN-based models, Capsule Networks, Attention mechanisms, Memory-augmented networks, Tranformers and Pre-Trained Language Models — PTMs and Named entity recognition.

(f) Multimodal Information Architecture was addressed following a partial view of Van Gigch and Moigne (1989), dealing only with epistemological basis and scientific findings, in order to later produce constructs for NLP implementation. Short reviews on the definition of Information and Architecture were produced, as well as verifying basic concepts on Modal Logic. On section 4.6.2 seven adequations and six properties were identified, leading to MIA definition exposed on section 4.6.3.

7.2 Scientific proposals

Continuing on Van Gigch and Moigne (1989) methodological path, epistemological findings lead the pursue for solutions of scientific problems. Table 1 divided construct production into two groups. As group 1 focus on characteristics of the problem, its context and actors, group 2 goes towards findings paths for solutions through pursuing evidence and presentation modes, logical basis and rationality.

On Deep Learning and Text Classification open questions, a broad survey by Minaee et al. (2021) identified four unaddressed problems on text classification tasks when using deep learning neural networks. These problems goes from technological restrictions (such as memory and data storage limits) to lack of specialized data sets and difficult to model human commonsense knowledge.

As for the context in which NLP is developed through DL, Minaee et al. (2021) suggest a five-step tutorial for choosing a text classification neural network, but still reminds that the four open problems need to be faced.

On MIA contributions on Text Classification, it was concluded that the presentation mode Minaee et al. (2021) suggested for their procedure has gaps that Information Science, through MIA, could fulfill.

To address these voids, a complementary five-step procedure for domain establishment was proposed based on MIA's definition reviewed on section Defining MIA. Each property listed on table 8 was transposed to a step on the process as is shown on table 34.

Table 34 – MIA Concept compared to 5-step procedure proposed on MIA contributions on Text Classification

[PRP]	Contribution on the definition	Step on section 5.2
[PRP.1]	Distinction and construction of Architectural worlds	Domain distinctions
[PRP.2]	Through assumption of Relational Models	Propose relationship between domains
[PRP.3] and **[PRP.4]**	Grouped by Space-Time contexts	Space-time context-based groupings
[PRP.5]	Of Information states	Identify context entities
[PRP.6]	Correlated or not	Identify entities correlations

Source: Produced by the author on January 2023

Considering that any static model gets outdated, the 5-step procedure needs to be periodically rerun, culminating on a cyclic model as shown on figure 70. With these products, the scientific level of Methodology is considered to be achieved, since, based on the construct classes listed on table 1 of the Research Method:

(a) Person/Psychological type can be defined when the procedure Identify context entities is executed, specifically on steps **[i]** and **[iii]**.

(b) Type of problem can also be verified on Identify context entities, but through steps **[ii]** and **[iv]**.

(c) Organizational context is obtained with Domain distinctions, which can be done through Description, Inspection or Verification.

(d) Evidence/Presentation mode is achieved on Identify entities correlations making use of four correlations: Definition, Comparison, Fusion and Decomposition.

(e) Logical basis is achieved while proposing relationship between domains, and verifying if their nature is either of identity, proximity or incidental, and restricting their range verifying if they are reflexive, serial, symmetric, transitive or euclidean.

(f) Rationality is addressed on Space-time context-based groupings, where Economy of Relations and Rules are applied.

7.3 Technological achievements

After producing scientific constructs based on epistemological findings, a 5-step procedure was developed. To validate it, at the technological level of Van Gigch and Moigne (1989) (presented on figure 1), theory and models need to be applied on real-world problems.

The scenario selected aimed two of Minaee et al. (2021) NLP open problems: absence of data sets for more complex tasks and commomsense knowledge models. A text classification task was selected, where the goal is to evaluate if the input is either or not suitable as RD&I according to commonsense of researches on 16 knowledge areas. Data coming from projects submitted on 2014 and 2015 were assembled.

Following what was previously reviewed on Deep Learning and Text Classification open questions, **[Step.1]** of Minaee et al. (2021) 5-step procedure for text classification model selection was applied on Model selection. BERTimbau, a BERT adaptation for brazilian portuguese was selected.

To compare results with and without MIA as the Research Method defined, an out-of-the-box distribution of BERTimbau was used, as Model instancing and data pre-processing described. A standard procedure for each cycle of experiments was defined, being formed by 10 rounds of 20 epochs totaling 200 epochs on each data set. The average of results on both ***loss*** and ***accuracy*** was observed during training, validation and testing.

Pre-conditioned simulation presented better results coming from 2015 data (77,57% on average), while 2014 and 2014/2015 combined were barely above 50% (54,79% and 58,22% respectively).

MIA's 5-step procedure was applied on sections 6.4.1 to 6.4.5, where 8 potential domains were produced. Data was re-arranged to portrait these new groupings and a new round of experiments was conducted.

Post-conditioned simulation analyzed all results compared to Pre-conditioned simulation with gain on ***accuracy*** registered on Potential domain 1, Potential domain 3, Potential domain 4 and Potential domain 8. These results indicates that better values can be achieved with posterior data enrichment of these domains.

For ***loss*** better results were achieved on Potential domain 1, Potential domain 3, Potential domain 4, Potential domain 5, Potential domain 6, Potential domain 7 and Potential domain 8. It indicates that more assertive learning procedures on weight adjustment can be achieved with data enrichment.

Tables 35, 36 and 37 shows a resume of all results obtained on each potential domain compared to the original data set which they correspond to.

Table 35 – Comparison of results - 2015 data

Variable	2015	Domain 1	Domain 2
Training Loss	0,5627345	0,5296761	0,5505137
Training Accuracy	76,86%	78,43%	75,77%
Validation loss	0,5708822	0,5286451	0,5717295
Validation Accuracy	74,14%	80,19%	72,78%
Test loss	0,4740491	0,4946408	0,5767502
Accuracy in Testing	77,57%	84,88%	72,65%

Source: Produced by the author on January 2023

Table 36 – Comparison of results - 2014 data

Variable	2014	Domain 3	Domain 4	Domain 5
Training Loss	0,7087808	0,7006512	0,7183891	0,7111632
Training Accuracy	53,55%	55,88%	55,41%	51,85%
Validation loss	0,6949488	0,6701859	0,7043763	0,6945764
Validation Accuracy	54,52%	58,00%	54,85%	52,65%
Test loss	0,7416452	0,6577451	0,6313299	0,7277233
Accuracy in Testing	54,79%	58,70%	54,98%	52,80%

Source: Produced by the author on January 2023

Table 37 – Comparison of results - 2014/2015 data

Variable	2014/2015	Domain 6	Domain 7	Domain 8
Training Loss	0,6463273	0,6629338	0,6799421	0,6265331
Training Accuracy	63,39%	63,30%	59,75%	67,11%
Validation loss	0,6765008	0,6571880	0,6634957	0,6602471

(Continue...)

Table 37 – Conclusion

Variable	2014/2015	Domain 6	Domain 7	Domain 8
Validation Accuracy	59,08%	63,40%	56,19%	63,38%
Test loss	0,6142412	0,5833146	0,6887856	0,6573988
Accuracy in Testing	58,22%	63,94%	55,15%	61,58%

Source: Produced by the author on January 2023

With these products, the technological level of Methodology is considered to be achieved, since:

(a) Tables 35, 36 and 37 addresses question **a.** of the problem proposed on chapter Implementing MIA on a NLP problem.

(b) Table 33 addresses question **b.** of the problem proposed on chapter Implementing MIA on a NLP problem.

8 Conclusion

Through this research, we aimed to position Information Science as an integral part of the process of building artificial intelligence, figuring as a discipline prior to the formalization of neural network algorithms. The pre-processing of data provided by MIA can contribute to increase the accuracy of predictions by simply rearranging the data provided, that is, by imposing a sense of dynamic organization according to the space-time treated.

In chapter 4 it was identified that the current stage of development of NLP provides a diverse range of algorithmic implementations, however, the most used training techniques (such as supervised learning) still require large volumes of classified data and improvements in specific knowledge or common-sense models (focused on questions about the real world) and with incomplete information.

In chapter 5, MIA, and its treatment of Modes of meaning expression were presented, following Kress and Van Leeuwen (2001) and Kress (2009); through modal logical structures, according to Carnielli and Pizzi (2008) and Portner (2009). By combining the two schools of thought, it becomes possible to manage different semantics in the same informational context, a very common problem in NLP tasks. The MIA approach is based, among other principles, on economy and relevance to provide the best possible informational configuration. It uses a 5-step procedure to identify subjects and their correlations with objects, as well as the domains to which subjects and objects belong and the relations between these domains.

In chapter 6 the MIA product construction procedure is applied to a real problem of classifying texts coming from 16 knowledge areas. Eight subdomains were designed without any change in the original amount of data. Using a widely used NLP algorithm for the Brazilian Portuguese language, the results obtained from data treated by MIA were compared to those obtained without such treatment.

Although the observed values were numerically discrete from the point of view of prediction accuracy, there is room for improvement in most of the distinguished domains. Considering that no data enrichment procedure or improvement of the linguistic model was performed, it is plausible to conclude that MIA, by itself, indicated the best possible grouping of data in each temporal moment, based only on the records initially presented.

Finally, in this research, the choice of IDF technique initially proposed by Jones (1973) to obtain correlations between subjects and objects in section 6.4.2 - Step 2: Identify entities correlations, does not bind MIA to its use, and can be replaced by any other technique that provides a measure of object relevance for each subject. Investigation of other methods of obtaining such a level of relevance is encouraged.

Bibliography

ABBAGNANO, N. *Dicionário de Filosofia*. 5. ed. [S.l.]: wmfMartinsFontes, 2015. Quoted 6 times on pages 27, 36, 37, 38, 148, and 150.

ACKLEY, D. H.; HINTON, G. E.; SEJNOWSKI, T. J. A learning algorithm for boltzmann machines. *Cognitive science*, Elsevier, v. 9, n. 1, p. 147–169, 1985. Quoted 2 times on pages 100 and 101.

AGATONOVIC-KUSTRIN, S.; BERESFORD, R. Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis*, Elsevier, v. 22, n. 5, p. 717–727, 2000. Quoted on page 72.

ALY, R.; REMUS, S.; BIEMANN, C. Hierarchical multi-label classification of text with capsule networks. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. [S.l.: s.n.], 2019. p. 323–330. Quoted on page 138.

AREL, I.; ROSE, D. C.; KARNOWSKI, T. P. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. *IEEE computational intelligence magazine*, IEEE, v. 5, n. 4, p. 13–18, 2010. Quoted 5 times on pages 92, 96, 114, 166, and 228.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. Quoted on page 139.

BASHEER, I. A.; HAJMEER, M. Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, Elsevier, v. 43, n. 1, p. 3–31, 2000. Quoted 5 times on pages 69, 70, 71, 72, and 230.

BELLMAN, R. *The theory of dynamic programming*. [S.l.], 1954. Quoted 6 times on pages 91, 92, 114, 166, 167, and 228.

BENGIO, Y. *Learning deep architectures for AI*. [S.l.]: Now Publishers Inc, 2009. Quoted on page 96.

BENGIO, Y.; DUCHARME, R.; VINCENT, P. A neural probabilistic language model. *Advances in Neural Information Processing Systems*, v. 13, 2000. Quoted on page 132.

BENGIO, Y.; LECUN, Y. et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, v. 34, n. 5, p. 1–41, 2007. Quoted 3 times on pages 96, 99, and 230.

BERNARDINI, S.; BARONI, M.; EVERT, S. A wacky introduction. *WaCky*, Citeseer, p. 9–40, 2006. Quoted on page 189.

BERTAGLIA, T. F. C.; NUNES, M. d. G. V. Exploring word embeddings for unsupervised textual user-generated content normalization. In: *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*. [S.l.: s.n.], 2016. p. 112–120. Quoted on page 203.

BHATTACHERJEE, A. *Social Science Research: Principles, Methods, and Practices*. [S.l.]: USF Tampa Bay Open Access Textbooks Collection, 2012. Quoted 2 times on pages 26 and 30.

BRIER, S. Finding an information concept suited for a universal theory of information. *Progress in biophysics and molecular biology*, Elsevier, v. 119, n. 3, p. 622–633, 2015. Quoted 3 times on pages 149, 157, and 158.

BUSH, V. As we may think. *SIGPC Note.*, ACM, New York, NY, USA, v. 1, n. 4, p. 36–44, abr. 1979. ISSN 0163-5816. Disponível em: <http://doi.acm.org/10.1145/1113634.1113638>. Quoted on page 23.

CARNIELLI, W.; PIZZI, C. *Modalities and multimodalities*. [S.l.]: Springer Science & Business Media, 2008. Quoted 6 times on pages 151, 152, 156, 164, 177, and 235.

CHOMSKY, N. *Aspects of the Theory of Syntax*. 1. ed. [S.l.]: MIT press, 1965. Quoted 2 times on pages 124 and 129.

CHOMSKY, N. *Knowledge of language: Its nature, origin, and use*. [S.l.]: Greenwood Publishing Group, 1986. Quoted 2 times on pages 123 and 229.

CHOMSKY, N.; GALLEGO, Á. J.; OTT, D. Generative grammar and the faculty of language: Insights, questions, and challenges. *Catalan Journal of Linguistics*, p. 229–261, 2019. Quoted on page 125.

CHOMSKY, N. et al. *Language and mind*. [S.l.]: Cambridge University Press, 2006. Quoted on page 122.

CONNEAU, A. et al. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016. Quoted on page 136.

CRESSWELL, J. W. *Research design: qualitative and mixed-method approaches*. [S.l.]: Thousand Oaks, California: Sage Publications Inc, 2003. Quoted on page 26.

DECHTER, R.; PEARL, J. Network-based heuristics for constraint-satisfaction problems. In: *Search in artificial intelligence*. [S.l.]: Springer, 1988. p. 370–425. Quoted on page 101.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. Quoted on page 144.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern classification*. second. [S.l.]: John Wiley & Sons, 2006. Quoted 2 times on pages 92 and 166.

EDELMAN, S.; WEINSHALL, D. A self-organizing multiple-view representation of 3d objects. *Biological Cybernetics*, Springer, v. 64, n. 3, p. 209–219, 1991. Quoted on page 114.

ELMAN, J. L. Finding structure in time. *Cognitive science*, Wiley Online Library, v. 14, n. 2, p. 179–211, 1990. Quoted 3 times on pages 9, 120, and 121.

ENGELBRECHT, A. P. *Computational intelligence: an introduction*. [S.l.]: John Wiley & Sons, 2007. Quoted 15 times on pages 8, 37, 71, 72, 74, 75, 76, 77, 78, 79, 83, 85, 86, 96, and 230.

FAUSETT, L. V. *Fundamentals of neural networks: architectures, algorithms and applications*. [S.l.]: Prentice-Hall, Englewood Cliffs, NJ, 1994. Quoted 14 times on pages 66, 67, 69, 70, 72, 73, 74, 76, 77, 79, 80, 81, 96, and 230.

FILHO, J. A. W. et al. The brwac corpus: A new open resource for brazilian portuguese. In: *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*. [S.l.: s.n.], 2018. Quoted 4 times on pages 11, 189, 190, and 191.

FLORIDI, L. Open problems in the philosophy of information. *Metaphilosophy*, JSTOR, p. 554–582, 2004. Quoted on page 149.

FLORIDI, L. *The Blackwell guide to the philosophy of computing and information*. [S.l.]: John Wiley & Sons, 2008. Quoted 2 times on pages 149 and 158.

FÖLDIÁK, P. Learning invariance from transformation sequences. *Neural computation*, MIT Press, v. 3, n. 2, p. 194–200, 1991. Quoted on page 114.

FOWLER, C. A. Coarticulation and theories of extrinsic timing. *Journal of phonetics*, Elsevier, v. 8, n. 1, p. 113–133, 1980. Quoted on page 118.

FREUND, Y. Boosting a weak learning algorithm by majority. *Information and computation*, Elsevier, v. 121, n. 2, p. 256–285, 1995. Quoted on page 111.

FRIEDMAN, J. H.; STUETZLE, W. Projection pursuit regression. *Journal of the American statistical Association*, Taylor & Francis, v. 76, n. 376, p. 817–823, 1981. Quoted on page 111.

GRISHMAN, R.; SUNDHEIM, B. M. Message understanding conference-6: A brief history. In: *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. [S.l.: s.n.], 1996. Quoted 2 times on pages 145 and 146.

GUO, B. et al. Improving text classification with weighted word embeddings via a multi-channel textcnn model. *Neurocomputing*, Elsevier, v. 363, p. 366–374, 2019. Quoted on page 136.

HAGAN, M. T.; DEMUTH, H. B.; BEALE, M. *Neural network design*. [S.l.]: Martin Hagan, 2014. Quoted 9 times on pages 64, 65, 66, 72, 79, 81, 83, 84, and 230.

HARRIS, Z. S. *Methods in structural linguistics.* [S.l.]: University of Chicago Press, 1951. Quoted 2 times on pages 123 and 229.

HASSOUN, M. H. et al. *Fundamentals of artificial neural networks*. [S.l.]: MIT press, 1995. Quoted 6 times on pages 65, 66, 69, 70, 72, and 230.

HAYKIN, S. *Neural networks: a comprehensive foundation*. second. [S.l.]: Prentice Hall PTR, 1999. Quoted 2 times on pages 68 and 69.

HAYKIN, S. *Neural Networks and Learning Machines*. third. [S.l.]: Prentice Hall PTR, 2009. Quoted 24 times on pages 8, 67, 68, 70, 71, 72, 74, 79, 80, 81, 82, 83, 85, 86, 87, 88, 89, 90, 92, 96, 114, 117, 166, and 230.

HEBB, D. O. *The organization of behavior: a neuropsychological theory*. [S.l.]: J. Wiley; Chapman & Hall, 1949. Quoted 3 times on pages 56, 57, and 229.

HINTON, G. E. *Relaxation and its role in vision*. Tese (Doutorado), 1977. Quoted 2 times on pages 101 and 102.

HINTON, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation*, MIT Press, v. 14, n. 8, p. 1771–1800, 2002. Quoted 3 times on pages 109, 110, and 111.

HINTON, G. E.; KRIZHEVSKY, A.; WANG, S. D. Transforming auto-encoders. In: SPRINGER. *International conference on artificial neural networks*. [S.l.], 2011. p. 44–51. Quoted 3 times on pages 9, 137, and 138.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, MIT Press, v. 18, n. 7, p. 1527–1554, 2006. Quoted 17 times on pages 8, 17, 18, 21, 22, 96, 99, 100, 107, 108, 109, 110, 111, 113, 114, 228, and 230.

HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. *science*, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006. Quoted on page 114.

HINTON, G. E.; SEJNOWSKI, T. J. Optimal perceptual inference. In: CITESEER. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. [S.l.], 1983. v. 448. Quoted on page 101.

HJØRLAND, B. What is knowledge organization (ko)? *Knowledge organization. International journal devoted to concept theory, classification, indexing and knowledge representation*, ERGON-Verlag GmbH,, 2008. Quoted on page 21.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 79, n. 8, p. 2554–2558, 1982. Quoted 2 times on pages 101 and 102.

HUANG, G. et al. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 4700–4708. Quoted on page 136.

HUBEL, D. H.; WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, Wiley Online Library, v. 160, n. 1, p. 106–154, 1962. Quoted on page 114.

IYYER, M. et al. Deep unordered composition rivals syntactic methods for text classification. In: *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*. [S.l.: s.n.], 2015. p. 1681–1691. Quoted on page 133.

JONES, K. S. Index term weighting. *Information storage and retrieval*, Elsevier, v. 9, n. 11, p. 619–633, 1973. Quoted 3 times on pages 146, 205, and 235.

JORDAN, M. *Serial order: a parallel distributed processing approach. Technical report, June 1985-March 1986*. [S.l.], 1986. Quoted 6 times on pages 9, 117, 118, 119, 120, and 121.

KAISER, Ł.; SUTSKEVER, I. Neural gpus learn algorithms. *arXiv preprint arXiv:1511.08228*, 2015. Quoted 2 times on pages 141 and 229.

KALCHBRENNER, N.; GREFENSTETTE, E.; BLUNSOM, P. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014. Quoted 3 times on pages 9, 135, and 136.

KEARNS, M.; VALIANT, L. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, ACM New York, NY, USA, v. 41, n. 1, p. 67–95, 1994. Quoted on page 111.

KENNEDY, G. *An introduction to corpus linguistics*. [S.l.]: Routledge, 2014. Quoted on page 128.

KIM, J. et al. Text classification using capsules. *Neurocomputing*, Elsevier, v. 376, p. 214–221, 2020. Quoted on page 138.

KOTHARI, C. R. *Research Methodology*. 2. ed. [S.l.]: New Age International Publisher, 2009. Quoted 2 times on pages 26 and 29.

KRESS, G. R. What is mode? In: JEWITT, C. (Ed.). *The Routledge Handbook of Multimodal Analysis*. [S.l.]: Routledge, 2009. p. 54–67. Quoted 6 times on pages 23, 154, 155, 158, 169, and 235.

KRESS, G. R.; Van Leeuwen, T. *Multimodal discourse: The modes and media of contemporary communication*. [S.l.]: JSTOR, 2001. Quoted 6 times on pages 147, 150, 153, 158, 172, and 235.

KRIPKE, S. Semantical considerations of the modal logic. *Acta Philosophica Fennica*, v. 16, p. 83–94, 1963. Quoted on page 42.

KUHN, T. S. *The Structure of Scientific Revolutions.* [S.l.]: Chicago: The University of Chicago Press, 2003. Quoted on page 27.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. *The annals of mathematical statistics*, JSTOR, v. 22, n. 1, p. 79–86, 1951. Quoted on page 109.

KUROKI JR., G. H. *Sobre uma arquitetura da informação multimodal: reflexões sobre uma proposta epistemológica*. Dissertação (Mestrado) — Universidade de Brasília, Fevereiro 2018. Quoted 26 times on pages 18, 23, 147, 148, 151, 152, 153, 154, 155, 156, 157, 159, 160, 161, 162, 163, 164, 165, 169, 171, 172, 175, 176, 178, 181, and 203.

LAKOFF, G. *Women, fire, and dangerous things: What categories reveal about the mind*. [S.l.]: University of Chicago press, 2008. Quoted on page 132.

LASHLEY, K. S. *The problem of serial order in behavior*. [S.l.]: Bobbs-Merrill Oxford, United Kingdom, 1951. Quoted on page 118.

LE, H. T.; CERISARA, C.; DENIS, A. Do convolutional networks need to be deep for text classification? In: *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2018. Quoted on page 136.

LECUN, Y. Efficient learning and second-order methods. *A tutorial at NIPS*, v. 93, p. 61, 1993. Quoted 2 times on pages 88 and 166.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015. Quoted on page 22.

LECUN, Y.; BENGIO, Y. et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, v. 3361, n. 10, p. 1995, 1995. Quoted on page 114.

LECUN, Y. et al. Backpropagation applied to handwritten zip code recognition. *Neural computation*, MIT Press, v. 1, n. 4, p. 541–551, 1989. Quoted 4 times on pages 8, 96, 115, and 137.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Ieee, v. 86, n. 11, p. 2278–2324, 1998. Quoted 4 times on pages 96, 114, 115, and 135.

LIU, P. et al. Multi-timescale long short-term memory neural network for modelling sentences and documents. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. [S.l.: s.n.], 2015. p. 2326–2335. Quoted on page 134.

MANNING, C.; SCHUTZE, H. *Foundations of statistical natural language processing*. [S.l.]: MIT press, 1999. Quoted 6 times on pages 122, 123, 124, 128, 129, and 229.

MCCANN, B. et al. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, v. 30, 2017. Quoted on page 144.

MCCARTHY, J. Epistemological problems of artificial intelligence. In: *Readings in artificial intelligence*. [S.l.]: Elsevier, 1981. p. 459–465. Quoted 3 times on pages 32, 42, and 229.

MCCARTHY, J.; HAYES, P. J. Some philosophical problems from the standpoint of artificial intelligence, machine intelligence. In: . [S.l.: s.n.], 1969. v. 4, p. 463–502. Quoted 7 times on pages 32, 33, 39, 40, 41, 42, and 229.

MCCARTHY, J. et al. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, v. 27, n. 4, p. 12–14, 2006. Quoted 3 times on pages 31, 32, and 229.

MCCLELLAND, J. L. et al. Parallel distributed processing. *Explorations in the Microstructure of Cognition*, MIT Press Cambridge, Ma, v. 2, p. 216–271, 1986. Quoted 11 times on pages 8, 58, 59, 60, 61, 62, 63, 79, 86, 228, and 230.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943. Quoted 2 times on pages 43 and 229.

MEMISEVIC, R.; HINTON, G. E. Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural computation*, MIT Press, v. 22, n. 6, p. 1473–1492, 2010. Quoted on page 81.

METROPOLIS, N. et al. Equation of state calculations by fast computing machines. *The journal of chemical physics*, American Institute of Physics, v. 21, n. 6, p. 1087–1092, 1953. Quoted on page 104.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. Quoted 4 times on pages 132, 133, 137, and 144.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, v. 26, 2013. Quoted 2 times on pages 139 and 144.

MINAEE, S. et al. Deep learning–based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 54, n. 3, p. 1–40, 2021. Quoted 10 times on pages 24, 139, 166, 167, 169, 170, 184, 228, 231, and 232.

MINSKY, M. Steps toward artificial intelligence. *Proceedings of the IRE*, IEEE, v. 49, n. 1, p. 8–30, 1961. Quoted 5 times on pages 35, 36, 39, 42, and 229.

MINSKY, M. L.; PAPERT, S. A. Perceptrons: expanded edition. MIT press, 1988. Quoted 12 times on pages 17, 55, 56, 57, 58, 59, 62, 63, 64, 114, 228, and 230.

MIYASHITA, Y. Neuronal correlate of visual associative long-term memory in the primate temporal cortex. *Nature*, Nature Publishing Group, v. 335, n. 6193, p. 817–820, 1988. Quoted on page 114.

MIYASHITA, Y. Inferior temporal cortex: where visual perception meets memory. *Annual review of neuroscience*, v. 16, n. 1, p. 245–263, 1993. Quoted on page 114.

MOORE, G. E. et al. *Cramming more components onto integrated circuits*. [S.l.]: McGraw-Hill New York, NY, USA:, 1965. Quoted 2 times on pages 24 and 25.

MUNKHDALAI, T.; YU, H. Neural semantic encoders. In: NIH PUBLIC ACCESS. *Proceedings of the conference. Association for Computational Linguistics. Meeting*. [S.l.], 2017. v. 1, p. 397. Quoted 3 times on pages 9, 140, and 141.

NEAL, R. M. Connectionist learning of belief networks. *Artificial intelligence*, Elsevier, v. 56, n. 1, p. 71–113, 1992. Quoted 6 times on pages 100, 102, 104, 105, 106, and 107.

NEAL, R. M.; HINTON, G. E. A view of the em algorithm that justifies incremental, sparse, and other variants. In: *Learning in graphical models*. [S.l.]: Springer, 1998. p. 355–368. Quoted on page 112.

NIELSEN, M. A. *Neural networks and deep learning*. [S.l.]: Determination press San Francisco, CA, USA:, 2015. Quoted on page 66.

PAPINENI, K. Why inverse document frequency? In: *Second Meeting of the North American Chapter of the Association for Computational Linguistics*. [S.l.: s.n.], 2001. Quoted on page 146.

PEARL, J. Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, Elsevier, v. 29, n. 3, p. 241–288, 1986. Quoted on page 107.

PEARL, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. [S.l.]: Morgan Kaufmann, 1988. Quoted 3 times on pages 105, 106, and 107.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. [S.l.: s.n.], 2014. p. 1532–1543. Quoted 4 times on pages 133, 137, 139, and 144.

PETERS, M. E. et al. *Deep contextualized word representations. In NAACL*. [S.l.]: Association for Computational Linguistics New Orleans, Louisiana, USA, 2018. Quoted on page 144.

POLLIO, M. V. *The Ten Books of Architecture*. [S.l.]: Dover Publications, INC., 1960. Quoted on page 148.

PORTNER, P. *Modality*. [S.l.]: Oxford University Press, 2009. Quoted 5 times on pages 152, 177, 178, 181, and 235.

QIU, X. et al. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, Springer, v. 63, n. 10, p. 1872–1897, 2020. Quoted 5 times on pages 9, 142, 143, 144, and 172.

RADFORD, A. et al. Improving language understanding by generative pre-training. 2018. Quoted on page 144.

REN, H.; LU, H. Compositional coding capsule network with k-means routing for text classification. *arXiv preprint arXiv:1810.09177*, 2018. Quoted on page 138.

RESMINI, A.; ROSATI, L. A brief history of information architecture. *Journal of Information Architecture*, v. 3, n. 2, 2012. Quoted on page 147.

ROSENBLATT, F. The perceptron–a perciving and recognizing automation. *Cornell Aeronautical Laboratory*, 1957. Quoted 3 times on pages 47, 95, and 101.

ROSENBLATT, F. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. [S.l.], 1961. Quoted 19 times on pages 8, 17, 43, 44, 45, 46, 48, 53, 54, 55, 57, 58, 66, 67, 95, 101, 228, 229, and 230.

ROSENFELD, L.; MORVILLE, P. *Information Architecture for the World Wide Web*. 3. ed. [S.l.]: O'Reilly Media, Inc., 2006. Quoted on page 147.

RUMELHART, D. E.; NORMAN, D. A. Simulating a skilled typist: A study of skilled cognitive-motor performance. *Cognitive science*, Wiley Online Library, v. 6, n. 1, p. 1–36, 1982. Quoted on page 118.

RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. [S.l.]: New Jersey; Pearson Education Inc,, 2010. Quoted 9 times on pages 33, 34, 35, 37, 38, 39, 42, 72, and 96.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, IBM, v. 3, n. 3, p. 210–229, 1959. Quoted 3 times on pages 57, 96, and 229.

SCHAPIRE, R. E. The strength of weak learnability. *Machine learning*, Springer, v. 5, n. 2, p. 197–227, 1990. Quoted on page 111.

SCHMIDHUBER, J.; HOCHREITER, S. et al. Long short-term memory. *Neural Comput*, v. 9, n. 8, p. 1735–1780, 1997. Quoted 3 times on pages 9, 121, and 122.

SHEN, D. et al. Deconvolutional latent-variable model for text sequence matching. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2018. v. 32, n. 1. Quoted on page 139.

SINHA, P.; POGGIO, T. Role of learning in three-dimensional form perception. *Nature*, Nature Publishing Group, v. 384, n. 6608, p. 460–463, 1996. Quoted on page 114.

SOMMERVILLE, I. *Engenharia de Software*. nineth. [S.l.]: Pearson, 2011. Quoted 2 times on pages 65 and 66.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. BERTimbau: pretrained BERT models for Brazilian Portuguese. In: *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*. [S.l.: s.n.], 2020. Quoted on page 189.

STONE, J. V. Object recognition using spatiotemporal signatures. *Vision research*, Elsevier, v. 38, n. 7, p. 947–951, 1998. Quoted on page 114.

STRYKER, M. P. Temporal associations. *Nature*, Nature Publishing Group, v. 354, n. 6349, p. 108–109, 1991. Quoted on page 114.

SUTTON, R. S. Learning to predict by the methods of temporal differences. *Machine learning*, Springer, v. 3, n. 1, p. 9–44, 1988. Quoted 4 times on pages 96, 97, 98, and 99.

TAI, K. S.; SOCHER, R.; MANNING, C. D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015. Quoted on page 134.

TESAURO, G. Practical issues in temporal difference learning. *Machine learning*, Springer, v. 8, n. 3, p. 257–277, 1992. Quoted 4 times on pages 97, 98, 99, and 166.

TURING, A. M. Computing machinery and intelligence. *Mind*, JSTOR, v. 59, n. 236, p. 433–460, 1950. Quoted 4 times on pages 31, 34, 164, and 228.

Van Gigch, J. P.; MOIGNE, J. L. L. A paradigmatic approach to the discipline of information systems. *Behavioral Science*, Wiley Online Library, v. 34, n. 2, p. 128–147, 1989. Quoted 9 times on pages 27, 28, 125, 148, 159, 184, 228, 230, and 232.

van GIGCH, J. P.; PIPINO, L. L. In search for a paradigm for the discipline of information systems. *Future Computing Systems*, v. 1, n. 1, p. 71–97, 1986. Quoted 2 times on pages 11 and 29.

VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017. Quoted 2 times on pages 141 and 229.

WALLIS, G. Temporal order in human object recognition learning. *Journal of Biological Systems*, World Scientific, v. 6, n. 03, p. 299–313, 1998. Quoted on page 114.

WALLIS, G.; BADDELEY, R. Optimal, unsupervised learning in invariant object recognition. *Neural computation*, MIT Press, v. 9, n. 4, p. 883–894, 1997. Quoted on page 114.

WALLIS, G.; BÜLTHOFF, H. Temporal correlations in presentation order during learning affects human object recognition. *Perception*, Sage Publications Sage UK: London, England, v. 26, n. 1_suppl, p. 33–33, 1997. Quoted on page 114.

WALLIS, G.; ROLLS, E. T. Invariant face and object recognition in the visual system. *Progress in neurobiology*, Elsevier, v. 51, n. 2, p. 167–194, 1997. Quoted on page 114.

WANG, G. et al. Joint embedding of words and labels for text classification. *arXiv preprint arXiv:1805.04174*, 2018. Quoted 3 times on pages 9, 139, and 140.

WASON, R. Deep learning: Evolution and expansion. *Cognitive Systems Research*, Elsevier, v. 52, p. 701–708, 2018. Quoted 5 times on pages 22, 24, 96, 166, and 170.

WICKELGREN, W. A. Context-sensitive coding, associative memory, and serial order in (speech) behavior. *Psychological Review*, American Psychological Association, v. 76, n. 1, p. 1, 1969. Quoted on page 118.

WILSON, D.; SPERBER, D. Relevance theory. In: HORN, L.; WARD, G. (Ed.). *Handbook of Pragmatics*. [S.l.]: Oxford: Blackwell, 2002. Quoted 3 times on pages 147, 158, and 170.

WITTGENSTEIN, L. *Philosophical Investigations [Philosophische Untersuchungen]*. 3. ed. [S.l.]: Oxford: Basil Blackwell, 1968. Quoted on page 129.

YADAV, V.; BETHARD, S. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*, 2019. Quoted on page 146.

YANG, M. et al. Investigating the transferring capability of capsule networks for text classification. *Neural Networks*, Elsevier, v. 118, p. 247–261, 2019. Quoted on page 138.

ZHANG, Y.; WALLACE, B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015. Quoted on page 136.

ZHAO, W. et al. Towards scalable and reliable capsule networks for challenging nlp applications. *arXiv preprint arXiv:1906.02829*, 2019. Quoted on page 138.

ZHAO, W. et al. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*, 2018. Quoted on page 138.

Apendices

Apendix A – Results on 2014 data

Table 38 – Results of first experiment on 2014 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7949004	52,50%	0,6185652	61,00%	n/a	n/a
1	0,6851865	54,00%	0,6277750	60,50%	n/a	n/a
2	0,6460773	56,50%	0,5691973	57,00%	n/a	n/a
3	0,9384468	62,50%	0,6354637	57,00%	n/a	n/a
4	0,7959859	56,00%	0,7159493	55,00%	n/a	n/a
5	0,6406864	49,50%	0,7077476	50,00%	n/a	n/a
6	0,5439230	55,50%	0,9228228	53,00%	n/a	n/a
7	0,9046478	60,50%	0,7917049	57,00%	n/a	n/a
8	0,7784222	57,50%	0,6355917	54,00%	n/a	n/a
9	0,7808352	53,00%	0,7459832	47,00%	n/a	n/a
10	0,7223143	46,00%	0,6575130	56,00%	n/a	n/a
11	0,7167006	55,50%	0,6795235	57,50%	n/a	n/a
12	0,7220683	50,50%	0,7169086	62,50%	n/a	n/a
13	0,5682819	47,50%	0,6416584	60,50%	n/a	n/a
14	0,8151908	50,50%	0,7347826	58,00%	n/a	n/a
15	0,6142252	57,50%	0,7082616	53,50%	n/a	n/a
16	0,7020993	51,50%	0,7001876	57,00%	n/a	n/a
17	0,8601171	51,50%	0,6538169	57,00%	n/a	n/a
18	1,2581174	53,00%	0,6383157	56,00%	n/a	n/a
19	0,6779841	57,50%	0,8311784	52,00%	n/a	n/a
Avg	0,7583105	53,93%	0,6966473	56,08%	0,8477135	55,06%

Source: Produced by the author in August, 2022

Table 39 – Results of second experiment on 2014 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6486155	52,50%	0,9040995	61,00%	n/a	n/a
1	0,7102854	55,50%	0,6156383	64,50%	n/a	n/a
2	0,5232702	57,50%	0,6741303	61,00%	n/a	n/a
3	0,6958539	52,50%	0,7105627	56,00%	n/a	n/a

(Continues...)

Table 39 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
4	0,6780108	50,00%	0,5459799	54,00%	n/a	n/a
5	0,7378047	51,50%	0,8357359	55,00%	n/a	n/a
6	0,4383443	53,50%	0,6049650	48,50%	n/a	n/a
7	0,6191703	56,50%	0,7898081	60,50%	n/a	n/a
8	0,6108978	52,50%	0,6124638	51,00%	n/a	n/a
9	0,8074877	49,00%	0,6332464	47,50%	n/a	n/a
10	0,6653479	51,50%	0,6570226	49,50%	n/a	n/a
11	0,8204926	56,00%	0,7154694	57,50%	n/a	n/a
12	0,8106873	55,50%	0,6781090	52,00%	n/a	n/a
13	0,8511988	58,50%	0,7376844	56,00%	n/a	n/a
14	0,6309580	49,00%	0,7395000	45,00%	n/a	n/a
15	0,6406996	55,00%	0,6049246	59,00%	n/a	n/a
16	0,7635577	56,50%	0,7698976	54,50%	n/a	n/a
17	0,6818399	51,50%	0,6083590	55,50%	n/a	n/a
18	0,6403782	51,50%	0,6891563	62,50%	n/a	n/a
19	0,9072688	50,50%	0,7863453	51,50%	n/a	n/a
Avg	0,6941085	53,33%	0,6956549	55,10%	0,8298105	50,84%

Source: Produced by the author in August, 2022

Table 40 – Results of third experiment on 2014 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,9327805	53,50%	0,5806274	52,00%	n/a	n/a
1	0,4474103	56,00%	0,6500346	58,50%	n/a	n/a
2	0,6831384	47,00%	0,7054921	54,50%	n/a	n/a
3	0,8477424	51,50%	0,7480712	46,50%	n/a	n/a
4	0,5707380	58,00%	0,7549136	49,00%	n/a	n/a
5	0,8612103	51,00%	0,7275118	41,00%	n/a	n/a
6	0,7256172	46,00%	0,7749674	55,00%	n/a	n/a
7	0,8034533	50,50%	0,9256303	59,00%	n/a	n/a
8	0,4687912	55,00%	0,6039168	53,50%	n/a	n/a
9	0,7070037	50,50%	0,7927691	57,50%	n/a	n/a
10	0,5317877	56,50%	0,8061046	53,50%	n/a	n/a
11	0,5875042	47,00%	0,6330793	65,50%	n/a	n/a
12	0,8862221	51,00%	0,6615312	54,00%	n/a	n/a
13	0,6237794	61,50%	0,6241574	56,50%	n/a	n/a
14	0,6609762	55,00%	0,6999649	49,50%	n/a	n/a

(Continues...)

Table 40 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
15	0,6394306	50,00%	0,7413539	46,00%	n/a	n/a
16	0,7074765	50,00%	0,6529260	51,00%	n/a	n/a
17	0,7608759	51,50%	0,7027460	46,50%	n/a	n/a
18	0,4610442	54,00%	0,7979902	48,00%	n/a	n/a
19	0,7979199	45,00%	0,7670668	53,00%	n/a	n/a
Avg	0,6852451	52,03%	0,7175427	52,50%	0,7677265	54,04%

Source: Produced by the author in August, 2022

Table 41 – Results of fourth experiment on 2014 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6532227	52,00%	0,7106130	40,50%	n/a	n/a
1	0,6801084	53,50%	0,6829406	46,00%	n/a	n/a
2	0,7185272	51,00%	0,8262768	58,50%	n/a	n/a
3	0,6176741	63,50%	0,6197854	55,00%	n/a	n/a
4	0,4692507	53,50%	0,6905543	59,50%	n/a	n/a
5	0,9574915	56,50%	0,6336297	60,00%	n/a	n/a
6	0,6262383	56,50%	0,5421263	57,50%	n/a	n/a
7	0,3499850	56,50%	0,6778589	52,00%	n/a	n/a
8	0,9504956	60,50%	0,7924860	47,50%	n/a	n/a
9	0,9076158	59,50%	0,6952178	50,50%	n/a	n/a
10	0,6484290	56,50%	0,7071191	52,50%	n/a	n/a
11	0,4642982	54,50%	0,6031833	56,50%	n/a	n/a
12	1,1034963	49,00%	0,8254137	57,50%	n/a	n/a
13	1,0156878	59,00%	0,9628505	58,50%	n/a	n/a
14	0,5186751	60,00%	0,5958272	53,50%	n/a	n/a
15	0,3567230	59,50%	0,7587935	53,00%	n/a	n/a
16	0,7760822	56,00%	0,6626908	60,00%	n/a	n/a
17	0,7086740	52,00%	0,7191935	57,00%	n/a	n/a
18	0,8123195	58,00%	0,6384875	55,00%	n/a	n/a
19	0,5238216	59,50%	0,3190822	57,00%	n/a	n/a
Avg	0,6929408	56,35%	0,6832065	54,38%	0,4169658	55,06%

Source: Produced by the author in August, 2022

Table 42 – Results of fifth experiment on 2014 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7175596	60,00%	0,8083842	52,50%	n/a	n/a
1	0,7052526	52,00%	0,6252323	54,50%	n/a	n/a
2	0,7411849	48,50%	0,7037268	59,00%	n/a	n/a
3	0,6235311	49,50%	0,6869599	49,50%	n/a	n/a
4	0,7685363	55,00%	0,6910932	50,00%	n/a	n/a
5	0,7432334	56,00%	0,7421734	49,00%	n/a	n/a
6	0,5873052	59,50%	0,6401047	58,50%	n/a	n/a
7	0,4645016	55,00%	0,8425777	54,50%	n/a	n/a
8	0,9380800	53,00%	0,6195529	59,50%	n/a	n/a
9	0,5117927	55,50%	0,4984117	53,50%	n/a	n/a
10	0,8747706	66,00%	0,6781937	52,00%	n/a	n/a
11	0,6212189	57,50%	0,6949012	59,50%	n/a	n/a
12	0,8567361	52,50%	0,7161428	58,50%	n/a	n/a
13	0,6714982	58,00%	0,8062156	62,00%	n/a	n/a
14	0,9746661	53,00%	0,5162292	57,50%	n/a	n/a
15	0,8352370	57,50%	0,7576380	56,50%	n/a	n/a
16	0,7007771	51,00%	0,5978177	57,50%	n/a	n/a
17	0,5559333	56,00%	0,7881820	58,50%	n/a	n/a
18	0,6726776	45,50%	0,6987426	50,50%	n/a	n/a
19	0,6588482	51,50%	0,5904691	58,00%	n/a	n/a
Avg	0,7111670	54,63%	0,6851374	55,55%	0,8090266	58,78%

Source: Produced by the author in August, 2022

Table 43 – Results of sixth experiment on 2014 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8045921	52,00%	0,6747758	52,00%	n/a	n/a
1	0,7166131	52,50%	0,7100624	39,50%	n/a	n/a
2	0,8133516	52,50%	0,7354256	41,50%	n/a	n/a
3	0,7035443	47,00%	0,7286836	53,00%	n/a	n/a
4	0,5054749	51,00%	0,6737986	45,00%	n/a	n/a
5	0,6465486	53,00%	0,6798636	45,50%	n/a	n/a
6	0,9764628	50,50%	0,6654010	65,50%	n/a	n/a
7	0,4833288	44,50%	0,6995611	55,00%	n/a	n/a
8	0,7465910	60,50%	0,9786991	57,50%	n/a	n/a
9	0,6612189	51,50%	0,7359458	58,00%	n/a	n/a

(Continues...)

Table 43 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
10	0,5845181	56,50%	0,6110937	62,50%	n/a	n/a
11	0,6117185	54,00%	0,7483517	53,00%	n/a	n/a
12	0,7791609	50,50%	0,6419594	58,00%	n/a	n/a
13	0,6326749	54,50%	0,7044830	51,50%	n/a	n/a
14	0,6389975	55,00%	0,5248328	58,50%	n/a	n/a
15	0,6337082	56,50%	0,8344196	59,00%	n/a	n/a
16	0,5720615	54,00%	0,5757610	57,50%	n/a	n/a
17	0,6263018	46,00%	0,7144926	61,00%	n/a	n/a
18	0,5658270	54,50%	0,5750313	53,00%	n/a	n/a
19	0,8876996	48,50%	0,6262930	57,00%	n/a	n/a
Avg	0,6795197	52,25%	0,6919467	54,18%	0,5093285	57,76%

Source: Produced by the author in August, 2022

Table 44 – Results of seventh experiment on 2014 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7454246	52,50%	0,7077975	40,00%	n/a	n/a
1	0,6236817	50,00%	0,8306026	42,50%	n/a	n/a
2	0,5640859	56,00%	0,7019941	49,00%	n/a	n/a
3	0,8212044	54,50%	0,8478966	58,00%	n/a	n/a
4	0,6890299	62,00%	0,7332617	54,50%	n/a	n/a
5	0,7467417	48,50%	0,5487180	45,00%	n/a	n/a
6	0,8211492	54,00%	0,8354251	41,50%	n/a	n/a
7	0,9365124	48,50%	0,5292770	48,00%	n/a	n/a
8	0,9095407	53,50%	0,6182691	47,00%	n/a	n/a
9	0,5653713	58,00%	0,6985181	58,00%	n/a	n/a
10	0,5765682	49,00%	0,6586609	57,00%	n/a	n/a
11	0,5982000	60,00%	0,7178566	62,50%	n/a	n/a
12	0,6617945	58,00%	0,6999197	58,00%	n/a	n/a
13	0,5874418	52,00%	0,8334048	54,50%	n/a	n/a
14	0,5974406	56,50%	0,7432246	44,00%	n/a	n/a
15	0,6778941	51,50%	0,4992696	65,00%	n/a	n/a
16	0,7323034	52,00%	0,6400969	54,00%	n/a	n/a
17	0,7706330	52,50%	0,8757213	53,00%	n/a	n/a
18	0,7255475	53,50%	0,7794642	52,00%	n/a	n/a
19	0,6113665	54,50%	0,8301891	54,50%	n/a	n/a
Avg	0,6980966	53,85%	0,7164784	51,90%	0,6540617	57,90%

(Continues...)

Table 44 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc

Source: Produced by the author in August, 2022

Table 45 – Results of eighth experiment on 2014 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,4814613	56,00%	0,7116921	47,50%	n/a	n/a
1	0,5681200	55,00%	0,7261175	45,50%	n/a	n/a
2	0,7710028	50,00%	0,9324701	52,50%	n/a	n/a
3	1,3083013	56,00%	0,3897251	54,50%	n/a	n/a
4	0,6337368	41,50%	0,7005144	57,00%	n/a	n/a
5	0,7569084	53,00%	0,7273713	50,00%	n/a	n/a
6	0,8898364	48,00%	0,9487137	54,00%	n/a	n/a
7	0,7550828	50,50%	0,6116062	53,00%	n/a	n/a
8	0,9303432	53,50%	0,7566437	51,00%	n/a	n/a
9	0,6950443	57,00%	0,6854038	58,00%	n/a	n/a
10	0,5701983	57,00%	0,6552128	54,00%	n/a	n/a
11	0,8695951	50,00%	0,6764731	53,00%	n/a	n/a
12	0,6724850	54,50%	0,5829391	61,00%	n/a	n/a
13	0,6714001	55,00%	0,7261515	57,00%	n/a	n/a
14	0,6426371	47,00%	0,7008697	56,00%	n/a	n/a
15	0,6691715	51,00%	0,6342636	55,50%	n/a	n/a
16	0,8496302	52,00%	0,8282161	49,50%	n/a	n/a
17	0,6145908	49,00%	0,7179464	50,00%	n/a	n/a
18	0,7018415	44,50%	0,5784381	54,50%	n/a	n/a
19	0,6540631	52,00%	0,7837570	53,50%	n/a	n/a
Avg	0,7352725	51,63%	0,7037263	53,35%	0,7235230	48,94%

Source: Produced by the author in August, 2022

Table 46 – Results of ninth experiment on 2014 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8860396	55,50%	0,6606434	62,50%	n/a	n/a
1	0,7393599	48,00%	0,6528517	61,00%	n/a	n/a
2	0,5778838	60,00%	0,6818920	50,00%	n/a	n/a
3	0,7018158	53,50%	0,7664292	49,00%	n/a	n/a
4	0,6624800	58,50%	0,6387773	48,50%	n/a	n/a
5	0,6034922	60,00%	0,7417393	45,50%	n/a	n/a

(Continues...)

Table 46 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	0,5826126	49,50%	0,6450889	48,00%	n/a	n/a
7	0,7242246	50,00%	0,5616950	52,50%	n/a	n/a
8	0,8213151	56,00%	0,6863160	57,50%	n/a	n/a
9	0,4721715	59,00%	0,6761529	51,00%	n/a	n/a
10	0,8243940	48,50%	0,6786346	50,00%	n/a	n/a
11	0,7129989	52,50%	0,7494006	60,00%	n/a	n/a
12	0,7718071	54,00%	0,8378677	56,00%	n/a	n/a
13	0,7455776	51,50%	0,6296467	50,50%	n/a	n/a
14	0,8604016	55,00%	0,6602813	53,50%	n/a	n/a
15	0,6472018	53,00%	0,5960555	55,00%	n/a	n/a
16	0,6549451	58,50%	0,9333178	54,50%	n/a	n/a
17	0,7113680	49,00%	0,6921424	46,00%	n/a	n/a
18	0,6853189	53,50%	0,6943157	59,00%	n/a	n/a
19	0,4686239	53,00%	0,6291050	57,00%	n/a	n/a
Avg	0,6927016	53,93%	0,6906176	53,35%	0,6860660	52,88%

Source: Produced by the author in August, 2022

Table 47 – Results of tenth experiment on 2014 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	1,0973585	54,50%	0,5444326	62,00%	n/a	n/a
1	0,8269335	54,00%	0,7130373	53,50%	n/a	n/a
2	0,5560229	53,00%	0,7677134	64,00%	n/a	n/a
3	0,6305745	48,50%	0,6205800	63,50%	n/a	n/a
4	0,6249843	54,50%	0,5494733	58,00%	n/a	n/a
5	0,7102541	53,50%	0,5878264	58,00%	n/a	n/a
6	0,8686517	58,50%	0,7500201	54,00%	n/a	n/a
7	0,8156230	50,00%	0,7652848	61,50%	n/a	n/a
8	1,1285733	53,50%	0,7464621	61,50%	n/a	n/a
9	0,7706653	50,50%	0,7534029	60,00%	n/a	n/a
10	0,5835296	59,00%	0,7744377	52,50%	n/a	n/a
11	0,5723345	58,00%	0,7399536	61,00%	n/a	n/a
12	0,6026642	55,00%	0,5826787	56,50%	n/a	n/a
13	0,7724863	51,00%	0,8698609	54,00%	n/a	n/a
14	0,6263317	57,50%	0,4364531	60,50%	n/a	n/a
15	0,8593588	53,00%	0,6712778	61,50%	n/a	n/a
16	0,8818342	46,00%	0,6171725	62,00%	n/a	n/a

(Continues...)

Table 47 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
17	0,6750432	49,50%	0,7062212	61,50%	n/a	n/a
18	0,5625139	54,50%	0,7564864	54,00%	n/a	n/a
19	0,6431732	58,50%	0,4178292	57,00%	n/a	n/a
Avg	0,7404455	53,63%	0,6685302	58,83%	1,1722298	56,59%

Source: Produced by the author in August, 2022

Apendix B – Results on 2015 data

Table 48 – Results of first experiment on 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5919667	73,50%	0,5353845	74,50%	n/a	n/a
1	0,5386323	70,00%	0,6951187	74,00%	n/a	n/a
2	0,2103295	78,00%	0,5956293	74,50%	n/a	n/a
3	0,6071352	77,00%	0,4455880	81,50%	n/a	n/a
4	0,6085314	79,00%	0,5254045	75,00%	n/a	n/a
5	0,7125048	76,50%	0,3278480	73,00%	n/a	n/a
6	0,3237929	77,00%	0,5384229	76,50%	n/a	n/a
7	0,5962031	75,50%	0,7972370	75,50%	n/a	n/a
8	0,5280744	73,00%	0,8054680	77,00%	n/a	n/a
9	0,2553644	82,50%	0,5925614	75,00%	n/a	n/a
10	0,8291055	76,00%	0,4694168	71,50%	n/a	n/a
11	0,7714212	80,00%	0,5142957	75,00%	n/a	n/a
12	0,9107401	73,50%	0,4812995	77,50%	n/a	n/a
13	0,6067253	74,00%	0,5024042	76,50%	n/a	n/a
14	0,1330328	79,50%	0,6822599	80,50%	n/a	n/a
15	0,1672173	75,00%	0,6962919	81,50%	n/a	n/a
16	0,7329937	80,50%	0,5386171	80,50%	n/a	n/a
17	0,7448115	77,00%	0,5000089	77,00%	n/a	n/a
18	0,8063686	77,00%	0,3587924	82,50%	n/a	n/a
19	1,3961593	75,50%	0,8059990	76,50%	n/a	n/a
Avg	0,6035555	76,50%	0,5704024	76,78%	0,4475300	80,38%

Source: Produced by the author in August, 2022

Table 49 – Results of second experiment on 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,2441293	75,00%	0,7100154	60,50%	n/a	n/a
1	0,5296401	74,50%	0,5814722	62,50%	n/a	n/a
2	0,9115253	76,00%	0,7185400	54,00%	n/a	n/a
3	1,3060549	77,50%	0,6929620	50,50%	n/a	n/a

(Continues...)

Table 49 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
4	0,5106158	72,50%	0,7072119	51,50%	n/a	n/a
5	0,3814978	74,00%	0,6951286	46,50%	n/a	n/a
6	0,2650309	77,50%	0,5991213	72,50%	n/a	n/a
7	0,1717300	80,50%	0,6646199	68,00%	n/a	n/a
8	0,1559172	80,00%	0,5719781	70,50%	n/a	n/a
9	0,2747818	73,00%	0,7294047	63,50%	n/a	n/a
10	0,5110408	74,50%	0,6397282	56,50%	n/a	n/a
11	0,6464033	81,00%	0,5108713	72,00%	n/a	n/a
12	1,0337794	74,50%	0,5616741	68,50%	n/a	n/a
13	0,9504415	78,50%	0,5788271	69,00%	n/a	n/a
14	0,7164844	74,50%	0,5786614	80,00%	n/a	n/a
15	0,3725447	73,00%	0,5862333	74,00%	n/a	n/a
16	0,3142716	76,50%	0,6246682	77,00%	n/a	n/a
17	0,2030429	70,50%	0,6907289	77,00%	n/a	n/a
18	0,6591719	77,00%	0,5098976	77,00%	n/a	n/a
19	0,5794507	75,00%	0,6034501	77,50%	n/a	n/a
Avg	0,5368777	75,78%	0,6277597	66,43%	0,6394976	73,44%

Source: Produced by the author in August, 2022

Table 50 – Results of third experiment on 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,3054702	73,00%	0,7749377	71,00%	n/a	n/a
1	0,5826857	74,00%	0,4474430	77,50%	n/a	n/a
2	0,7439518	75,00%	0,5768518	75,00%	n/a	n/a
3	0,2286867	73,00%	0,7811539	79,00%	n/a	n/a
4	0,8383491	75,00%	0,5974174	79,50%	n/a	n/a
5	0,4783671	73,00%	0,8362600	76,50%	n/a	n/a
6	0,6465051	76,50%	0,3099444	72,50%	n/a	n/a
7	0,1714644	79,00%	0,3391671	77,50%	n/a	n/a
8	0,2627703	75,00%	0,4835074	84,50%	n/a	n/a
9	0,6161329	76,50%	0,3116934	80,00%	n/a	n/a
10	0,7710413	80,50%	0,3939205	79,50%	n/a	n/a
11	0,7342073	76,50%	0,4411526	78,50%	n/a	n/a
12	0,2244423	81,00%	0,4718508	74,50%	n/a	n/a
13	0,9913752	75,00%	0,8691555	81,00%	n/a	n/a
14	0,3477624	75,00%	0,3595903	78,50%	n/a	n/a

(Continues...)

Table 50 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
15	1,0105927	73,00%	0,6633825	79,50%	n/a	n/a
16	0,6794361	75,00%	0,4911578	74,50%	n/a	n/a
17	0,7133491	76,00%	0,5074158	78,00%	n/a	n/a
18	0,7063704	83,50%	0,5279266	85,00%	n/a	n/a
19	0,4479445	74,50%	0,4340127	78,50%	n/a	n/a
Avg	0,5750452	76,00%	0,5308971	78,03%	0,4042923	79,29%

Source: Produced by the author in August, 2022

Table 51 – Results of fourth experiment on 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,2256222	81,00%	0,6759819	81,00%	n/a	n/a
1	0,5912940	74,00%	0,7311602	78,00%	n/a	n/a
2	0,2259254	76,50%	0,5081422	72,00%	n/a	n/a
3	0,3929793	78,50%	0,4086522	72,50%	n/a	n/a
4	0,3369058	76,00%	0,4903743	79,50%	n/a	n/a
5	0,4363535	75,50%	0,4368210	78,50%	n/a	n/a
6	0,5621735	73,00%	0,5025112	76,50%	n/a	n/a
7	0,7633084	80,50%	0,3937404	71,50%	n/a	n/a
8	0,2544874	77,50%	0,6062434	75,00%	n/a	n/a
9	0,9765195	75,00%	0,6005446	80,50%	n/a	n/a
10	0,3062845	74,00%	0,5730585	74,00%	n/a	n/a
11	0,3269576	77,00%	0,7336128	77,00%	n/a	n/a
12	0,5171427	80,50%	0,3797305	79,00%	n/a	n/a
13	0,3458786	81,50%	0,5484667	68,50%	n/a	n/a
14	0,8894989	74,50%	0,5581068	69,50%	n/a	n/a
15	0,8231086	75,50%	0,5732239	77,00%	n/a	n/a
16	0,8759883	79,50%	0,9086243	76,00%	n/a	n/a
17	0,5650457	77,00%	0,5119247	76,50%	n/a	n/a
18	0,4391822	75,50%	0,5554785	79,00%	n/a	n/a
19	0,5284597	75,50%	0,7499183	79,50%	n/a	n/a
Avg	0,5191558	76,90%	0,5723158	76,05%	0,2344655	80,28%

Source: Produced by the author in August, 2022

Table 52 – Results of fifth experiment on 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7285171	62,50%	0,5859945	77,00%	n/a	n/a
1	0,6872408	76,00%	0,3834957	78,50%	n/a	n/a
2	0,8960449	76,50%	0,5182625	74,00%	n/a	n/a
3	0,2711399	76,00%	0,4271541	74,50%	n/a	n/a
4	0,6141436	78,50%	0,4353042	75,50%	n/a	n/a
5	0,3616088	74,00%	0,7932988	72,50%	n/a	n/a
6	1,7281718	78,00%	0,6247676	66,00%	n/a	n/a
7	0,6682110	76,50%	0,7042753	49,50%	n/a	n/a
8	0,4913838	69,50%	0,8575317	40,50%	n/a	n/a
9	0,5668409	76,00%	0,3735896	76,00%	n/a	n/a
10	0,3754464	78,00%	0,5293664	76,50%	n/a	n/a
11	0,3278203	78,50%	0,4262120	77,00%	n/a	n/a
12	0,7562330	80,00%	0,4685021	74,50%	n/a	n/a
13	0,1976856	76,50%	0,6901255	71,00%	n/a	n/a
14	0,4139877	76,50%	0,6098819	74,00%	n/a	n/a
15	0,5542701	77,00%	0,6235631	68,50%	n/a	n/a
16	0,4128354	75,00%	0,6052501	54,00%	n/a	n/a
17	0,3040583	75,00%	0,5610760	71,50%	n/a	n/a
18	1,2656769	75,50%	0,6169575	69,50%	n/a	n/a
19	0,2339421	80,00%	0,5597643	74,50%	n/a	n/a
Avg	0,5927629	75,78%	0,5697187	69,75%	0,2924765	76,71%

Source: Produced by the author in August, 2022

Table 53 – Results of sixth experiment on 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5863527	75,50%	0,5864198	74,00%	n/a	n/a
1	0,5375255	77,50%	0,3765526	76,50%	n/a	n/a
2	0,3328282	74,00%	0,5415517	74,00%	n/a	n/a
3	1,5692167	80,50%	0,5109328	71,50%	n/a	n/a
4	0,6348511	78,50%	0,8908367	77,00%	n/a	n/a
5	0,5985556	76,50%	0,5571322	73,50%	n/a	n/a
6	0,1705558	77,00%	0,3061916	74,00%	n/a	n/a
7	0,1979906	78,50%	0,5727470	80,00%	n/a	n/a
8	0,4115731	77,50%	0,2791476	75,50%	n/a	n/a
9	1,0037740	78,50%	0,4316104	73,50%	n/a	n/a

(Continues...)

Table 53 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
10	0,1295910	77,50%	0,1986080	77,00%	n/a	n/a
11	0,4648623	73,50%	0,5347815	76,50%	n/a	n/a
12	0,2442098	79,00%	0,8995459	74,00%	n/a	n/a
13	0,3549423	80,50%	0,3703164	81,50%	n/a	n/a
14	0,6402529	73,50%	0,3609849	82,50%	n/a	n/a
15	0,8533948	84,50%	0,7773508	80,00%	n/a	n/a
16	0,4189608	76,50%	0,5307012	76,50%	n/a	n/a
17	0,4290383	78,50%	0,5459332	76,00%	n/a	n/a
18	0,6705388	77,50%	0,4613760	74,50%	n/a	n/a
19	0,6597159	75,50%	0,5014837	81,00%	n/a	n/a
Avg	0,5454365	77,53%	0,5117102	76,45%	0,2730931	79,48%

Source: Produced by the author in August, 2022

Table 54 – Results of seventh experiment on 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,3096785	69,00%	0,4304023	75,50%	n/a	n/a
1	1,0115857	70,50%	0,5849230	77,00%	n/a	n/a
2	0,2103916	79,50%	0,5037295	81,00%	n/a	n/a
3	0,2301192	77,50%	0,8984795	73,00%	n/a	n/a
4	0,3283833	74,50%	0,5455024	84,50%	n/a	n/a
5	1,0563787	81,50%	0,5957892	79,50%	n/a	n/a
6	0,3307182	75,00%	0,5942986	70,50%	n/a	n/a
7	0,8816885	73,00%	0,4328967	74,50%	n/a	n/a
8	0,9199513	78,50%	0,4859911	84,50%	n/a	n/a
9	1,0722793	74,50%	0,5684364	81,00%	n/a	n/a
10	0,1900768	77,50%	0,4787036	78,50%	n/a	n/a
11	0,2001379	78,50%	0,6781726	77,50%	n/a	n/a
12	0,3372349	71,00%	0,7007944	51,00%	n/a	n/a
13	0,4410232	75,00%	0,7664533	32,50%	n/a	n/a
14	0,6228316	67,50%	0,6265409	63,50%	n/a	n/a
15	0,2558668	77,00%	0,7528162	68,50%	n/a	n/a
16	0,5500527	80,50%	0,5639186	71,50%	n/a	n/a
17	0,5349680	73,50%	0,6879840	69,00%	n/a	n/a
18	1,3391886	77,00%	0,6271147	59,50%	n/a	n/a
19	0,6744369	82,50%	0,6715565	75,00%	n/a	n/a
Avg	0,5748496	75,68%	0,6097252	71,38%	1,1172572	72,55%

(Continues...)

Table 54 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc

Source: Produced by the author in August, 2022

Table 55 – Results of eighth experiment on 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,1790002	73,00%	0,5291491	76,50%	n/a	n/a
1	0,5283285	72,00%	0,6398566	68,00%	n/a	n/a
2	0,2680543	80,00%	0,4331490	74,00%	n/a	n/a
3	0,6292998	69,50%	0,6136703	73,50%	n/a	n/a
4	0,2083032	73,50%	0,6193783	67,50%	n/a	n/a
5	0,6742515	79,00%	0,8574091	76,50%	n/a	n/a
6	0,1346239	77,50%	0,5091112	78,00%	n/a	n/a
7	0,5971509	82,00%	0,8518654	75,00%	n/a	n/a
8	0,4595541	77,00%	0,5760021	77,00%	n/a	n/a
9	0,4355873	75,00%	0,8122280	72,00%	n/a	n/a
10	0,5165604	81,00%	0,5699927	76,50%	n/a	n/a
11	0,6303987	78,50%	0,6708144	74,00%	n/a	n/a
12	0,3225009	71,00%	0,7328194	70,50%	n/a	n/a
13	0,6505342	74,00%	0,6232228	77,50%	n/a	n/a
14	0,6256593	76,00%	0,5834588	81,50%	n/a	n/a
15	0,4434034	76,00%	0,5943404	79,50%	n/a	n/a
16	0,3479732	77,50%	0,4591535	76,50%	n/a	n/a
17	0,7218008	80,50%	0,4507836	77,00%	n/a	n/a
18	0,3625420	79,00%	0,6442187	78,50%	n/a	n/a
19	0,2950112	77,00%	0,8651473	70,50%	n/a	n/a
Avg	0,4515269	76,45%	0,6317885	75,00%	0,4269388	72,05%

Source: Produced by the author in August, 2022

Table 56 – Results of ninth experiment on 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6028005	74,50%	0,5057543	75,00%	n/a	n/a
1	1,4692701	75,00%	0,5415965	75,00%	n/a	n/a
2	0,6187277	76,50%	0,5178742	75,50%	n/a	n/a
3	0,4818570	79,50%	0,7872961	77,50%	n/a	n/a
4	0,5550278	75,00%	0,5398790	68,50%	n/a	n/a
5	0,1819807	78,50%	0,5337733	74,00%	n/a	n/a

(Continues...)

Table 56 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	0,7859756	77,50%	0,4610662	80,00%	n/a	n/a
7	0,8080217	75,00%	0,5074123	78,50%	n/a	n/a
8	0,7560703	72,50%	0,6281604	75,50%	n/a	n/a
9	0,6881427	79,50%	0,6922425	75,00%	n/a	n/a
10	0,2299731	75,50%	0,4286608	72,00%	n/a	n/a
11	0,2080807	75,50%	0,8011672	77,00%	n/a	n/a
12	0,7162681	79,50%	0,4599836	78,50%	n/a	n/a
13	0,6142604	77,00%	0,5762404	74,50%	n/a	n/a
14	1,0540851	80,00%	0,4018570	79,50%	n/a	n/a
15	0,3378013	80,50%	0,5960414	69,50%	n/a	n/a
16	0,5986524	73,00%	0,5536824	65,00%	n/a	n/a
17	0,8071145	75,00%	0,7843536	72,50%	n/a	n/a
18	0,8875253	77,50%	0,4522455	79,00%	n/a	n/a
19	0,6943554	73,00%	0,6739711	80,50%	n/a	n/a
Avg	0,6547995	76,50%	0,5721629	75,13%	0,2561494	75,82%

Source: Produced by the author in August, 2022

Table 57 – Results of tenth experiment on 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,3408393	75,50%	0,4955066	71,50%	n/a	n/a
1	0,4385881	74,50%	0,4651158	70,00%	n/a	n/a
2	0,5445462	80,50%	0,3344733	71,00%	n/a	n/a
3	0,9943987	74,50%	0,5262893	79,50%	n/a	n/a
4	0,2662215	78,00%	0,3864289	77,50%	n/a	n/a
5	0,5241067	79,50%	0,6389092	76,50%	n/a	n/a
6	0,6566597	72,00%	0,6156737	68,50%	n/a	n/a
7	0,6190189	81,50%	0,4499844	79,50%	n/a	n/a
8	0,2110641	74,50%	0,6627436	70,50%	n/a	n/a
9	0,6733099	79,50%	0,7012867	70,00%	n/a	n/a
10	1,1275575	81,50%	0,3446533	75,50%	n/a	n/a
11	0,3397878	75,00%	0,4986269	83,00%	n/a	n/a
12	0,7919068	71,50%	0,7247266	77,00%	n/a	n/a
13	0,2252327	81,00%	0,3937976	79,50%	n/a	n/a
14	0,3012863	74,50%	0,5636654	74,00%	n/a	n/a
15	0,4537689	73,00%	0,4257833	81,00%	n/a	n/a
16	0,6889637	78,00%	0,5886508	76,50%	n/a	n/a

(Continues...)

Table 57 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
17	0,5476603	73,50%	0,6271926	81,00%	n/a	n/a
18	0,7216163	81,00%	0,5328943	80,00%	n/a	n/a
19	1,0001640	75,50%	0,2704313	86,00%	n/a	n/a
Avg	0,5733349	76,73%	0,5123417	76,40%	1,2919983	80,67%

Source: Produced by the author in August, 2022

Apendix C – Step 1 Code Listing - Text Normalization and Lemmatization

```
# install enelvo portuguese NLP normalizer
!pip install enelvo

# install stanza portuguese lemmatizer
!pip install git+https://github.com/stanfordnlp/stanza.git

# import stopwords cleanse enabler
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import string
from string import punctuation
from string import digits
import re

# data handles
import numpy as np
import pandas as pd
from gensim.models import Word2Vec
import torch
from torch.utils.data import Dataset, DataLoader

data2014 = pd.read_csv('../input/entitydomainanalysis/LB-2014-Labels.tsv',
sep='\t',
engine='python',
encoding='latin-1')

data2015 = pd.read_csv('../input/entitydomainanalysis/LB-2015-Labels.tsv',
sep='\t',
engine='python',
encoding='latin-1')

data2014.rename(columns={'Coluna1': 'COMITE'}, inplace=True)
data2015.rename(columns={'Coluna1': 'COMITE'}, inplace=True)

# Separate, from each knowledge area, Element/Barrier/Methodology values
    divided into two sets: aprroved and not approved.
data2014.rename(columns = {'COMITE':'Comite'}, inplace = True)
```

```
data2014.rename(columns = {'METODOLOGIA / MéTODOS UTILIZADOS':'Metodo'},
    inplace = True)
data2014.rename(columns = {'BARREIRA OU DESAFIO TENOLóGICO SUPERáVEL':'
    Barreira'}, inplace = True)
data2014.rename(columns = {'ELEMENTO TECNOLOGICAMENTE NOVO OU INOVADOR':'
    Elemento'}, inplace = True)

data2015.rename(columns = {'COMITE':'Comite'}, inplace = True)
data2015.rename(columns = {'METODOLOGIA / MéTODOS UTILIZADOS':'Metodo'},
    inplace = True)
data2015.rename(columns = {'BARREIRA OU DESAFIO TENOLóGICO SUPERáVEL':'
    Barreira'}, inplace = True)
data2015.rename(columns = {'ELEMENTO TECNOLOGICAMENTE NOVO OU INOVADOR':'
    Elemento'}, inplace = True)

data2014['APROVACAO'] = data2014['APROVACAO'].apply(lambda x: 0 if x == 'Nã
    o' else 1)
data2015['APROVACAO'] = data2015['APROVACAO'].apply(lambda x: 0 if x == '0'
     else 1)

frames = [data2014, data2015]
data = pd.concat(frames)

print(data['Comite'].unique())

comiteAtivo = '' # insert what is the knowledge area treated at each time

targetDataAprovado = data[data.Comite == str(comiteAtivo)]
targetDataReprovado = data[data.Comite == str(comiteAtivo)]

targetDataAprovado.drop(targetDataAprovado.index[targetDataAprovado['
    APROVACAO'] == 0], inplace=True)
targetDataReprovado.drop(targetDataReprovado.index[targetDataReprovado['
    APROVACAO'] == 1], inplace=True)

nltk.download('stopwords')
stopwords = set(nltk.corpus.stopwords.words('portuguese') + list(
    punctuation) + list(digits))

# Approved set for Barrier/Element/Method
frasesBarreiraAprovado = targetDataAprovado['Barreira'].tolist()
frasesElementoAprovado = targetDataAprovado['Elemento'].tolist()
frasesMetodoAprovado = targetDataAprovado['Metodo'].tolist()

# Not approved set for Barrier/Element/Method
frasesBarreiraReprovado = targetDataReprovado['Barreira'].tolist()
frasesElementoReprovado = targetDataReprovado['Elemento'].tolist()
```

```python
frasesMetodoReprovado = targetDataReprovado['Metodo'].tolist()

# Start normalization
from enelvo.normaliser import Normaliser
norm = Normaliser(tokenizer='readable')

def normaliseData(frases):
        print("normalising data...")
        for i in range (len(frases)):
                frases[i] = norm.normalise(frases[i])

normaliseData(frasesBarreiraAprovado)
normaliseData(frasesElementoAprovado)
normaliseData(frasesMetodoAprovado)

normaliseData(frasesBarreiraReprovado)
normaliseData(frasesElementoReprovado)
normaliseData(frasesMetodoReprovado)

# Start data cleanse
punctuation = '!"#$%&\'()*+,-.:;<=>?@[\\]^_`{}~'
slashes = '/|'
def cleanData(frases):
        print("cleaning data...")
        for i in range (len(frases)):
                frases[i] = frases[i].lower()
                for character in punctuation:
                        frases[i] = frases[i].replace(character, '')
                for character in slashes:
                        frases[i] = frases[i].replace(character, ' ')
                frases[i] = " ".join(frases[i].split())

cleanData(frasesBarreiraAprovado)
cleanData(frasesElementoAprovado)
cleanData(frasesMetodoAprovado)

cleanData(frasesBarreiraReprovado)
cleanData(frasesElementoReprovado)
cleanData(frasesMetodoReprovado)

# Start Lemmatizing
import stanza

stanza.download('pt')
nlp = stanza.Pipeline('pt')

def lemmatizeData(frases):
```

```
        print("lemmatizing data...")
        for i in range (len(frases)):
                lemma = ""
                for frase in nlp(frases[i]).sentences:
                        for word in frase.words:
                                if (word.upos == 'ADJ' or word.upos == 'NOUN'):
                                        lemma += word.lemma + " "
        frases[i] = lemma

lemmatizeData(frasesBarreiraAprovado)
lemmatizeData(frasesElementoAprovado)
lemmatizeData(frasesMetodoAprovado)

lemmatizeData(frasesBarreiraReprovado)
lemmatizeData(frasesElementoReprovado)
lemmatizeData(frasesMetodoReprovado)

# Final data cleanse

def filterData(frases, res):
        print("filtering data...")
        for frase in frases:
                filtered_sentence = []
                word_tokens = word_tokenize(frase)
                for word in word_tokens:
                        if word not in stopwords:
                                filtered_sentence.append(word)
                res.append(filtered_sentence)

resBarreiraAprovado = []
resElementoAprovado = []
resMetodoAprovado = []
resBarreiraReprovado = []
resElementoReprovado = []
resMetodoReprovado = []

filterData(frasesBarreiraAprovado, resBarreiraAprovado)
filterData(frasesElementoAprovado, resElementoAprovado)
filterData(frasesMetodoAprovado, resMetodoAprovado)

filterData(frasesBarreiraReprovado, resBarreiraReprovado)
filterData(frasesElementoReprovado, resElementoReprovado)
filterData(frasesMetodoReprovado, resMetodoReprovado)

```

Code Listing C.1 – Text Normalization and Lemmatization

Apendix D – Step 2 Code Listing - TF-IDF

```
# make a wordset of all words after normalizing, lemmatizing and cleansing
def makeWordSet(res):
        wordset = {}
        for i in range (len(res)):
                sentence = res[i]
                wordset = set(wordset).union(set(sentence))
        return wordset

# Wordset for approved instances
wordsetBarreiraAprovado = makeWordSet(resBarreiraAprovado)
wordsetElementoAprovado = makeWordSet(resElementoAprovado)
wordsetMetodoAprovado = makeWordSet(resMetodoAprovado)

# Wordset for not approved instances
wordsetBarreiraReprovado = makeWordSet(resBarreiraReprovado)
wordsetElementoReprovado = makeWordSet(resElementoReprovado)
wordsetMetodoReprovado = makeWordSet(resMetodoReprovado)

def populateWordDict (res, worddict, wordset):
        for i in range (len(res)):
                worddict.append(dict.fromkeys(wordset,0))

def evaluateWordDict (res, worddict):
        for i in range (len(worddict)):
                for word in res[i]:
                        worddict[i][word]+=1

worddictBarreiraAprovado = []
tfBowBarreiraAprovado = []

worddictElementoAprovado = []
tfBowElementoAprovado = []

worddictMetodoAprovado = []
tfBowMetodoAprovado = []

worddictBarreiraReprovado = []
tfBowBarreiraReprovado = []

worddictElementoReprovado = []
```

```python
tfBowElementoReprovado = []

worddictMetodoReprovado = []
tfBowMetodoReprovado = []

populateWordDict(resBarreiraAprovado, worddictBarreiraAprovado,
    wordsetBarreiraAprovado)
populateWordDict(resElementoAprovado, worddictElementoAprovado,
    wordsetElementoAprovado)
populateWordDict(resMetodoAprovado, worddictMetodoAprovado,
    wordsetMetodoAprovado)

populateWordDict(resBarreiraReprovado, worddictBarreiraReprovado,
    wordsetBarreiraReprovado)
populateWordDict(resElementoReprovado, worddictElementoReprovado,
    wordsetElementoReprovado)
populateWordDict(resMetodoReprovado, worddictMetodoReprovado,
    wordsetMetodoReprovado)

evaluateWordDict(resBarreiraAprovado, worddictBarreiraAprovado)
evaluateWordDict(resElementoAprovado, worddictElementoAprovado)
evaluateWordDict(resMetodoAprovado, worddictMetodoAprovado)

evaluateWordDict(resBarreiraReprovado, worddictBarreiraReprovado)
evaluateWordDict(resElementoReprovado, worddictElementoReprovado)
evaluateWordDict(resMetodoReprovado, worddictMetodoReprovado)

def computeTF(wordDict,bow,tfBow):
        tfDict = {}
        bowCount = len(bow)
        for word,count in wordDict.items():
                if bowCount > 0:
                        tfDict[word] = count / float(bowCount)
                else:
                        tfDict[word] = 0
        tfBow.append(tfDict)

def batchTF (res,worddict,tfbow):
        for i in range (len(res)):
                computeTF(worddict[i], res[i], tfbow)

batchTF (resBarreiraAprovado, worddictBarreiraAprovado,
    tfBowBarreiraAprovado)
batchTF (resElementoAprovado, worddictElementoAprovado,
    tfBowElementoAprovado)
batchTF (resMetodoAprovado, worddictMetodoAprovado, tfBowMetodoAprovado)

```

```
batchTF (resBarreiraReprovado, worddictBarreiraReprovado,
    tfBowBarreiraReprovado)
batchTF (resElementoReprovado, worddictElementoReprovado,
    tfBowElementoReprovado)
batchTF (resMetodoReprovado, worddictMetodoReprovado, tfBowMetodoReprovado)

def computeIDF (docList):
        import math
        idfDict = {}
        N = len(docList)
        idfDict = dict.fromkeys(docList[0],0)
        for doc in docList:
                for word,val in doc.items():
                        if val > 0:
                                idfDict[word]+=1
        for word, value in idfDict.items():
                idfDict[word] = math.log(N/float(value))
        return idfDict

idfsBarreiraAprovado = computeIDF(tfBowBarreiraAprovado)
idfsElementoAprovado = computeIDF(tfBowElementoAprovado)
idfsMetodoAprovado = computeIDF(tfBowMetodoAprovado)

idfsBarreiraReprovado = computeIDF(tfBowBarreiraReprovado)
idfsElementoReprovado = computeIDF(tfBowElementoReprovado)
idfsMetodoReprovado = computeIDF(tfBowMetodoReprovado)

def computeTFIDF(tfBow,idfs,tfidf):
        uniTFIDF = {}
        for word,val in tfBow.items():
                uniTFIDF[word] = val * idfs[word]
        tfidf.append(uniTFIDF)

def batchTFIDF (tfBow,idfs,tfidf):
        for i in range (len(tfBow)):
                computeTFIDF(tfBow[i],idfs,tfidf)

tfidfBarreiraAprovado = []
tfidfElementoAprovado = []
tfidfMetodoAprovado = []

tfidfBarreiraReprovado = []
tfidfElementoReprovado = []
tfidfMetodoReprovado = []

batchTFIDF(tfBowBarreiraAprovado,idfsBarreiraAprovado,tfidfBarreiraAprovado
    )
```

```
batchTFIDF(tfBowElementoAprovado,idfsElementoAprovado,tfidfElementoAprovado)
batchTFIDF(tfBowMetodoAprovado,idfsMetodoAprovado,tfidfMetodoAprovado)

batchTFIDF(tfBowBarreiraReprovado,idfsBarreiraReprovado,tfidfBarreiraReprovado)
batchTFIDF(tfBowElementoReprovado,idfsElementoReprovado,tfidfElementoReprovado)
batchTFIDF(tfBowMetodoReprovado,idfsMetodoReprovado,tfidfMetodoReprovado)

tfidfComiteBarreiraAprovado = pd.DataFrame(tfidfBarreiraAprovado)
tfidfComiteElementoAprovado = pd.DataFrame(tfidfElementoAprovado)
tfidfComiteMetodoAprovado = pd.DataFrame(tfidfMetodoAprovado)

tfidfComiteBarreiraReprovado = pd.DataFrame(tfidfBarreiraReprovado)
tfidfComiteElementoReprovado = pd.DataFrame(tfidfElementoReprovado)
tfidfComiteMetodoReprovado = pd.DataFrame(tfidfMetodoReprovado)

tfidfComiteBarreiraAprovado.loc["Total"] = tfidfComiteBarreiraAprovado.sum()
tfidfComiteElementoAprovado.loc["Total"] = tfidfComiteElementoAprovado.sum()
tfidfComiteMetodoAprovado.loc["Total"] = tfidfComiteMetodoAprovado.sum()

tfidfComiteBarreiraReprovado.loc["Total"] = tfidfComiteBarreiraReprovado.sum()
tfidfComiteElementoReprovado.loc["Total"] = tfidfComiteElementoReprovado.sum()
tfidfComiteMetodoReprovado.loc["Total"] = tfidfComiteMetodoReprovado.sum()

dfObjBarreiraAprovado = tfidfComiteBarreiraAprovado.sort_values(by ='Total', axis=1, ascending=False)
dfObjElementoAprovado = tfidfComiteElementoAprovado.sort_values(by ='Total', axis=1, ascending=False)
dfObjMetodoAprovado = tfidfComiteMetodoAprovado.sort_values(by ='Total', axis=1, ascending=False)

dfObjBarreiraReprovado = tfidfComiteBarreiraReprovado.sort_values(by ='Total', axis=1, ascending=False)
dfObjElementoReprovado = tfidfComiteElementoReprovado.sort_values(by ='Total', axis=1, ascending=False)
dfObjMetodoReprovado = tfidfComiteMetodoReprovado.sort_values(by ='Total', axis=1, ascending=False)

dfObjBarreiraAprovado = dfObjBarreiraAprovado.iloc[-1]
dfObjElementoAprovado = dfObjElementoAprovado.iloc[-1]
dfObjMetodoAprovado = dfObjMetodoAprovado.iloc[-1]
```

```

dfObjBarreiraReprovado = dfObjBarreiraReprovado.iloc[-1]
dfObjElementoReprovado = dfObjElementoReprovado.iloc[-1]
dfObjMetodoReprovado = dfObjMetodoReprovado.iloc[-1]

finalObjBarreiraAprovado = dfObjBarreiraAprovado.to_frame()
finalObjElementoAprovado = dfObjElementoAprovado.to_frame()
finalObjMetodoAprovado = dfObjMetodoAprovado.to_frame()

finalObjBarreiraReprovado = dfObjBarreiraReprovado.to_frame()
finalObjElementoReprovado = dfObjElementoReprovado.to_frame()
finalObjMetodoReprovado = dfObjMetodoReprovado.to_frame()

finalObjBarreiraAprovado.to_csv('TF-IDF-' + comiteAtivo + ' - Barreira -
    Aprovado.csv', sep='\t', encoding='utf-8', decimal=',')
finalObjElementoAprovado.to_csv('TF-IDF-' + comiteAtivo + ' - Elemento -
    Aprovado.csv', sep='\t', encoding='utf-8', decimal=',')
finalObjMetodoAprovado.to_csv('TF-IDF-' + comiteAtivo + ' - Metodo -
    Aprovado.csv', sep='\t', encoding='utf-8', decimal=',')

finalObjBarreiraReprovado.to_csv('TF-IDF-' + comiteAtivo + ' - Barreira -
    Reprovado.csv', sep='\t', encoding='utf-8', decimal=',')
finalObjElementoReprovado.to_csv('TF-IDF-' + comiteAtivo + ' - Elemento -
    Reprovado.csv', sep='\t', encoding='utf-8', decimal=',')
finalObjMetodoReprovado.to_csv('TF-IDF-' + comiteAtivo + ' - Metodo -
    Reprovado.csv', sep='\t', encoding='utf-8', decimal=',')
```

Code Listing D.1 – TF-IDF calculation

Apendix E – Step 2 - 2015 Identified entities throughout experiments

Table 58 – 2015 scores for identified entities - Agroindustry, Food and Consumer Goods

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
produto	0,34184	10,24679	8,14527	0,44989	6,23895	7,00296	18,32348	6,13287	0,96621	7,53080	19,88180	1,78877	10,71777	5,21807	0,16186	0,03797	6,44908	7,75996
novo	0,59388	8,84481	8,08089	0,51486	6,12484	3,96149	21,40812	5,39427	0,86203	2,20597	19,71628	1,52453	9,17346	8,07478	0,48140	0,05905	6,06379	7,78034
processo	0,34960	8,44510	9,30615	0,45670	4,99695	5,76632	16,60599	6,87767	1,00866	0,61658	17,34182	1,58766	8,89643	7,40267	0,37218	0,07903	5,63184	6,93674
desenvolvimento	0,45379	7,64748	7,80815	0,49250	6,52649	5,48402	17,27029	5,34976	0,72716	1,13845	17,18458	1,52526	8,27004	7,70495	0,43417	0,04443	5,50384	6,73888
projeto	0,21635	6,65927	8,19005	0,59903	6,35712	3,70845	18,34480	6,33585	0,93452	1,69403	14,49100	1,18160	6,62444	7,29613	0,36869	0,03847	5,18999	6,51819
sistema	0,23601	3,76440	7,58568	0,55863	6,99969	1,24343	17,50002	3,43819	0,65947	0,47941	15,79372	0,86384	5,34629	8,63933	0,37660	0,03907	4,59524	6,42679
anexo	0,06604	4,59360	2,04789	0,00000	1,43280	0,26053	25,05176	0,15270	0,27465	0,15987	7,45546	0,00000	0,55771	18,50409	0,00000	0,00000	3,78482	9,23035
teste	0,32568	5,29704	5,42272	0,35221	4,46750	2,86433	11,30995	3,90311	0,57374	0,47919	12,59983	1,02011	6,43298	4,68618	0,27606	0,00990	3,75128	5,59869
aplicação	0,25902	4,27495	5,14841	0,37837	3,56395	1,29498	10,17503	3,13383	0,41303	0,14874	11,02968	1,09412	7,86189	5,72760	0,19163	0,01308	3,41927	4,44995
grande	0,38907	4,78526	5,11049	0,42604	3,84318	2,62744	10,30808	3,25128	0,45738	0,76681	10,48403	1,10914	5,16748	4,99467	0,18403	0,03102	3,37096	4,68818
estudo	0,39271	6,01164	4,20955	0,44577	3,91971	3,83931	10,21468	3,57384	0,59745	0,88053	10,91361	1,02591	4,59914	3,12394	0,25803	0,01953	3,37659	4,65089
tecnologia	0,25771	3,74093	3,89808	0,16310	4,82887	1,88245	8,39811	1,40691	0,53449	0,79236	12,05078	0,94842	5,50632	6,25534	0,27992	0,00000	3,18399	4,42946
material	0,38590	2,57096	5,49995	0,44127	2,95885	1,01469	11,11879	4,74128	0,65811	0,90402	11,24140	0,95139	5,03368	0,60292	0,03422	0,01465	3,01076	4,35713
pesquisa	0,35672	5,63971	4,59292	0,21923	3,26644	2,57778	7,81261	3,46385	0,62856	0,47858	11,12813	0,99414	4,39393	4,12913	0,24094	0,03345	3,12226	4,47453
produção	0,35402	5,61829	3,89115	0,18131	2,40148	2,80165	8,50527	3,71808	0,57352	0,38667	10,57527	1,34063	6,27161	2,00545	0,09098	0,02414	3,04622	3,82485
alto	0,43911	3,63257	4,04944	0,30434	3,30661	2,91676	7,48035	2,60766	0,69001	0,33107	8,16128	0,84054	5,77194	3,43279	0,20528	0,03532	2,76282	3,70696
equipamento	0,12806	5,09665	4,83820	0,42257	4,10385	1,65963	13,00827	2,22617	0,34160	0,37609	7,45458	0,48662	3,01943	1,73064	0,30763	0,06144	2,82884	4,02494
necessário	0,21776	3,53043	4,29246	0,39381	2,81747	3,09755	8,13375	2,96689	0,44909	0,25870	9,41858	0,73512	3,55635	4,15767	0,14476	0,00000	2,76065	3,79259
mercado	0,15625	4,41009	3,87946	0,14094	2,94974	1,90895	8,93793	2,96280	0,39032	0,44507	7,07772	0,89206	5,90288	3,62589	0,22362	0,00939	2,74457	3,55539
forma	0,14679	3,39525	4,31776	0,24404	3,41887	2,87262	7,55647	1,83140	0,37659	0,26407	8,57574	0,91678	3,89110	5,12468	0,21857	0,01110	2,69761	3,61438
solução	0,06445	2,47506	3,68448	0,28677	3,99383	1,62717	7,54094	1,77595	0,31381	0,17991	9,64080	0,37193	3,52642	6,90270	0,40009	0,01263	2,67481	3,85715
análise	0,27108	4,67981	3,54965	0,23386	3,61021	2,06862	9,20221	2,18953	0,50100	0,16149	9,04549	0,83252	3,26255	4,29387	0,14699	0,00964	2,75366	4,47116
qualidade	0,24934	5,05890	3,71174	0,31107	3,12065	1,74018	7,14752	2,79772	0,37922	0,35678	8,62074	1,34261	3,78996	2,54176	0,16656	0,03219	2,58543	3,22537
bom	0,43458	4,32589	3,14123	0,21394	2,60058	1,81340	6,26906	2,19025	0,46033	0,26830	8,74437	1,03771	6,20366	3,39317	0,11950	0,02624	2,57764	3,24585
linha	0,08889	5,10728	4,20156	0,10487	3,47341	1,91533	9,53459	2,40661	0,11211	1,01360	7,79150	0,58871	3,49865	1,17139	0,11977	0,03575	2,57275	3,72736
técnico	0,16171	3,54429	4,31646	0,39784	3,62054	1,92335	8,73559	2,33208	0,37485	0,27692	7,90812	0,59182	3,39591	3,24542	0,25438	0,02705	2,56915	3,68049
utilização	0,13608	3,63837	3,42972	0,34495	2,35759	1,51215	6,23250	2,15399	0,36383	0,23910	11,57443	0,83539	3,54269	3,97852	0,14202	0,00000	2,53008	3,62168
dado	0,04534	1,91766	2,97821	0,20349	4,49548	0,92911	4,89312	2,10648	0,28842	0,04806	10,52774	0,42937	1,60105	8,70409	0,43954	0,00000	2,47545	4,04196
tecnológico	0,16062	4,56468	2,76811	0,10273	2,75708	1,84853	6,49763	2,48937	0,58527	0,27513	7,44030	0,66811	4,15840	4,22079	0,18402	0,03709	2,42237	3,59252
informação	0,06879	4,39462	2,74783	0,10058	2,78506	0,59759	6,63776	0,38502	0,12871	0,16981	11,16515	0,14812	1,41442	7,80528	0,28647	0,01144	2,42791	4,31447
metodologia	0,17280	2,94114	2,90254	0,49420	4,30894	2,54439	5,86261	1,46241	0,46590	0,46060	8,59293	0,66019	3,92836	4,11325	0,20650	0,00000	2,44480	3,98747

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
controle	0,40830	2,90878	3,66387	0,20723	3,76622	1,69822	7,20777	1,86549	0,34856	0,08053	6,35718	0,88365	3,35644	3,98783	0,17834	0,03322	2,30948	2,98681
tempo	0,02751	3,48711	4,03343	0,18193	2,57262	1,25666	7,50571	2,08023	0,33821	0,21255	6,32613	0,44928	4,38532	4,12951	0,15347	0,01110	2,32192	3,09552
ferramenta	0,00000	0,51639	3,76777	0,20351	2,44173	0,08993	9,21676	7,03961	0,24909	0,85586	5,11584	0,30158	0,83349	5,49611	0,13675	0,00000	2,26653	3,95448
característica	0,37883	5,64372	3,77673	0,27891	2,34235	1,79530	5,37901	1,85906	0,43123	0,20786	7,64686	0,86334	4,95243	1,53478	0,02928	0,02474	2,32153	3,14351
necessidade	0,04231	2,71532	3,29894	0,18637	3,32362	1,00259	6,37182	2,20973	0,19428	0,33408	7,40234	0,66627	3,71274	4,32292	0,17552	0,01915	2,24863	3,03958
formulação	0,18388	6,47269	1,04781	0,03509	0,58818	5,00172	0,82675	0,57147	0,13400	0,00000	12,56167	0,81419	8,76144	0,14591	0,00000	0,00852	2,32208	4,85310
base	0,09049	3,76090	2,76853	0,30183	2,19633	1,43873	4,42224	1,50631	0,43942	0,50234	8,59758	0,59580	4,56032	3,99820	0,14359	0,04312	2,21036	3,03723
principal	0,22610	3,24305	3,24766	0,21828	2,97959	1,80825	6,42482	1,86737	0,58543	0,11009	6,45491	0,45290	3,52629	3,42686	0,10404	0,00000	2,16723	3,30699
empresa	0,03439	4,25633	3,13567	0,31713	2,03666	1,73413	6,76784	2,18012	0,29474	0,64773	6,09661	0,51283	3,21486	3,73494	0,21605	0,03009	2,20063	2,81761
custo	0,13765	3,80173	3,77045	0,11655	3,29828	0,69473	6,77938	1,96001	0,22492	0,46822	6,34511	0,69502	4,05608	2,16646	0,14503	0,00000	2,16623	2,86724
componente	0,04545	1,37204	5,58890	0,24099	3,02546	1,14103	10,30434	1,44492	0,23470	0,23174	4,30336	0,15617	2,21468	3,08926	0,13908	0,00000	2,09576	3,30435
avaliação	0,39416	3,19860	3,08339	0,14397	2,47226	2,70278	5,61793	1,95883	0,26157	0,07543	8,84024	0,71260	4,24597	1,49865	0,18259	0,01948	2,21303	3,81174
elemento	0,11877	2,04752	3,80082	0,20166	2,45779	1,43080	8,12307	2,52712	0,34515	0,22827	5,91791	0,65877	2,84124	2,37013	0,13558	0,01371	2,07614	3,59549
desafio	0,14553	3,26579	2,81941	0,20724	2,23329	1,91150	5,88838	1,92700	0,26909	0,30432	6,69857	0,48627	3,27995	3,58665	0,15518	0,01900	2,07482	4,34732
uso	0,12240	2,89807	3,03330	0,07739	3,10817	1,93308	4,32960	1,57248	0,32002	0,39980	6,97336	0,67262	3,98761	3,50237	0,12953	0,02544	2,06783	2,70351
tipo	0,11801	4,31677	3,62960	0,22485	2,02845	1,40999	5,64239	1,79681	0,55608	0,58447	6,86461	0,73180	3,73771	1,74441	0,13055	0,01898	2,09597	2,70602
redução	0,29456	3,38394	3,67641	0,15933	1,98076	1,29292	7,15672	2,37328	0,23776	0,15211	6,88673	0,68034	3,35710	1,24567	0,00000	0,00000	2,05485	3,06704
cliente	0,00000	3,33161	2,03520	0,07664	1,80955	0,05018	4,48710	2,31185	0,20957	0,38324	7,28500	0,53659	5,18962	5,70950	0,28352	0,00000	2,10620	2,95142
método	0,15066	2,80396	4,01641	0,15673	3,37591	1,86129	4,54382	1,16294	0,41757	0,13658	7,59979	0,51434	3,38544	2,83790	0,13411	0,00000	2,06859	3,01927
resistência	0,48115	2,01353	3,28865	0,41619	1,03347	0,23405	6,33210	2,86827	0,54117	0,37330	5,75785	1,06894	7,21281	0,19874	0,08305	0,01342	1,99479	2,99376
nome	0,00000	0,12553	8,41748	0,00000	8,29465	0,04053	4,39640	1,04326	0,00000	0,00000	1,35605	4,12196	3,54063	0,06149	0,00000	0,00000	1,96237	6,06080
software	0,00000	0,94004	3,32066	0,15571	4,93470	0,00000	7,75295	2,29480	0,00000	0,16467	5,56780	0,03249	0,84792	5,18404	0,16197	0,03216	1,96187	3,28504
fabricação	0,00000	2,84582	5,13757	0,12154	1,95900	3,71873	7,07832	3,66908	0,15326	0,52756	3,53403	0,32860	1,83647	0,27318	0,01757	0,02200	1,95142	2,87316
modelo	0,01720	1,05230	3,28121	0,06795	3,50058	0,26036	7,48334	0,51772	0,50202	0,25954	6,84339	0,95229	1,21350	4,56880	0,13101	0,00000	1,91570	3,00929
resultado	0,29946	2,93522	2,97033	0,17544	2,19141	1,58431	4,38740	2,72704	0,29428	0,14645	6,68048	0,63357	3,66555	2,43799	0,08086	0,02054	1,95189	2,75622
melhoria	0,15613	2,74869	3,37931	0,08747	2,34746	0,77682	6,40345	1,59019	0,22770	0,81631	5,62358	0,69655	2,63756	2,47451	0,10682	0,01854	1,88069	2,39464
desempenho	0,16372	4,04347	2,29613	0,10913	2,58304	1,01133	4,98804	1,81303	0,20503	0,06470	4,76164	0,27521	4,29188	3,03746	0,16438	0,03706	1,86533	2,45886
possível	0,15042	2,91133	2,94642	0,25741	1,88313	0,80051	5,50037	1,60946	0,26661	0,26774	6,00162	0,49669	2,77547	3,06275	0,17784	0,01144	1,81995	2,43726
etapa	0,20669	2,82694	2,67161	0,16664	2,29184	3,44740	4,08366	2,61493	0,24190	0,11116	5,23613	0,38245	2,62349	2,21125	0,10285	0,03303	1,82825	2,62695
baixo	0,10658	1,83910	2,31329	0,17800	2,37989	1,54312	4,34300	1,26516	0,57764	0,12057	6,60808	0,69116	3,91240	2,01739	0,13948	0,01998	1,75343	2,70475
final	0,10744	4,52134	3,51241	0,17790	1,38659	0,91145	3,95512	1,96650	0,24733	0,12848	5,06864	0,66146	4,34392	1,87040	0,10459	0,02067	1,81152	2,40675
segurança	0,12686	1,20303	2,21065	0,22536	2,17656	3,51185	5,20470	1,05058	0,02277	0,05889	5,91110	0,35412	1,43970	4,15145	0,16513	0,00964	1,73890	2,53517

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
temperatura	0,12208	3,48974	3,85208	0,10993	2,15210	0,62085	4,89614	2,41493	0,51038	0,12690	5,21519	0,43720	3,88108	0,39961	0,07725	0,02108	1,77041	2,49294
condição	0,43177	2,15069	2,06699	0,45125	2,09510	0,83116	4,74667	1,40622	0,46352	0,02592	6,84771	0,83397	4,59688	0,84828	0,01479	0,00964	1,73878	2,40899
conhecimento	0,13682	3,44479	2,53434	0,24123	1,91231	0,50147	5,00325	2,27600	0,25329	0,16475	5,92481	0,52278	2,47399	2,43394	0,07302	0,00000	1,74355	2,35913
risco	0,09156	1,61591	2,84541	0,17529	1,80002	1,29568	4,70525	2,07399	0,26719	0,16320	6,41846	0,52127	2,52137	2,60235	0,02196	0,02702	1,69662	2,80538
conceito	0,01859	1,36590	3,19578	0,11833	1,95546	0,25646	8,33492	1,18519	0,12752	0,04813	5,56850	0,33629	1,10051	3,63356	0,23294	0,00000	1,71738	2,88604
estrutura	0,03827	2,05040	2,74746	0,46909	1,87092	0,51531	5,06499	1,07122	0,19287	0,25441	7,21608	0,68563	2,42592	2,67417	0,15784	0,05332	1,71799	2,58119
operação	0,09429	1,40928	2,81773	0,20588	3,04147	0,30260	6,93469	2,07526	0,24828	0,05336	4,54651	0,26266	1,93774	3,19820	0,12824	0,00000	1,70351	2,50379
mecânico	0,00000	0,62512	4,17248	0,19402	2,25087	0,15774	7,78264	3,82584	0,24910	0,18954	4,32030	0,37221	2,50865	0,09795	0,08028	0,02652	1,67833	2,96465
pequeno	0,23447	2,48858	2,75708	0,16171	1,55060	1,15089	5,09394	2,01384	0,24316	0,56949	5,99636	0,49907	2,74533	1,50921	0,06940	0,01144	1,69341	2,45719
requisito	0,04744	0,92940	2,19383	0,11325	2,09219	0,29997	7,86676	1,65208	0,06072	0,11594	4,35554	0,16442	3,14700	3,44251	0,12536	0,00000	1,66290	2,84287
inovador	0,11582	2,07357	2,75250	0,09301	2,13748	1,73319	5,09559	1,33458	0,27682	0,22244	5,04877	0,52281	2,81533	2,04563	0,04314	0,01254	1,64520	3,37888
peça	0,00000	0,41114	3,55323	0,22996	0,78306	0,02927	11,32331	3,06068	0,03870	0,92560	4,04921	0,28953	1,35747	0,27698	0,00000	0,01077	1,64618	3,19295
realização	0,21867	2,48614	2,48999	0,31322	1,96368	1,40011	4,77848	1,61434	0,27170	0,12236	5,96035	0,46314	3,45154	1,86528	0,13761	0,00990	1,72166	3,01857
eficiência	0,17485	2,54348	2,63144	0,18756	1,84021	0,71066	5,62462	1,02260	0,30448	0,18250	5,24991	0,49230	3,09673	1,58461	0,03761	0,02226	1,60661	2,15072
máquina	0,07268	1,31238	4,36525	0,23984	1,01018	0,12428	9,24447	2,53023	0,05990	0,14240	3,41280	0,97881	0,92077	1,27305	0,00000	0,00000	1,60544	2,83673
protótipo	0,06191	2,17374	4,17382	0,10894	2,67957	0,21728	5,75530	1,48323	0,15326	0,11886	5,17140	0,30702	2,36979	1,95066	0,11564	0,02409	1,67903	3,45396
tratamento	0,23104	2,29810	1,30169	0,17537	0,98846	4,29877	3,06832	2,06258	0,32823	0,12007	5,84404	0,41517	2,87323	1,84219	0,11452	0,00000	1,62261	2,52917
ensaio	0,39651	1,23590	4,48785	0,37652	2,58324	0,69430	4,91744	2,35802	0,26183	0,17392	4,41258	0,12142	3,37464	0,09316	0,07877	0,00000	1,59788	3,03748
descritivo	0,00000	0,08328	0,11533	0,00000	0,02321	0,00000	21,79991	0,04013	0,00000	0,00000	0,19980	0,02640	0,77232	1,58497	0,00000	0,00000	1,54033	7,55341
objetivo	0,12165	3,53424	2,30697	0,09182	1,73160	1,04759	4,10829	1,69422	0,36763	0,14102	4,78673	0,46134	2,49608	2,21432	0,12827	0,00000	1,57699	2,15469
água	0,19068	3,78258	2,18253	0,40674	1,67023	1,20488	2,79878	0,80027	0,47050	0,10680	6,80800	0,81270	4,07558	0,04413	0,08212	0,00000	1,58978	2,58270
performance	0,03762	2,76164	1,20238	0,10944	0,86496	0,31133	4,30254	0,72623	0,20085	0,04279	6,61292	0,36183	3,98016	3,31406	0,21227	0,01665	1,56610	2,19898
barreira	0,08074	2,42742	2,42360	0,06876	1,59672	1,31795	4,23438	1,25740	0,32599	0,14535	4,97252	0,58269	2,67859	2,19362	0,08912	0,00000	1,52468	3,31789
produtividade	0,33908	1,82780	2,55270	0,25522	0,88874	0,66915	5,48406	1,38290	0,27207	0,18861	4,60715	0,60901	3,61670	1,50617	0,01657	0,03280	1,51555	2,43411
elétrico	0,00000	0,49323	5,32288	0,18648	4,60614	0,00000	8,47286	0,81161	0,06530	0,05561	2,64296	0,14085	0,50208	0,59098	0,09310	0,00000	1,49901	2,94653
definição	0,07272	1,63533	2,12115	0,20943	2,15053	0,78679	5,15553	1,11041	0,24080	0,10157	5,72154	0,30476	2,19733	2,87340	0,19234	0,04095	1,55716	2,67157
estabilidade	0,22568	3,14355	1,23657	0,12631	0,67739	4,08517	2,05425	0,49987	0,23706	0,04171	5,57771	0,15274	5,80659	0,55860	0,03456	0,00000	1,52861	2,59147
campo	0,24994	2,91551	1,96839	0,21752	2,28602	0,06055	8,19081	0,78622	0,33928	0,08879	3,77053	0,48791	2,65297	1,42060	0,22432	0,00000	1,60371	3,44888
específico	0,05781	2,46718	2,14617	0,23090	1,42903	1,38978	3,56929	1,45988	0,21566	0,25053	5,54028	0,48256	2,20823	2,36774	0,05196	0,01465	1,49260	2,05722
área	0,12658	1,43737	2,07201	0,07797	2,46511	0,88023	4,56725	1,72202	0,23546	0,14542	5,09619	0,70148	1,99022	2,06392	0,05039	0,00852	1,47751	1,93373
validação	0,02646	1,99367	2,21641	0,14299	2,17910	1,39278	5,48397	1,20225	0,18179	0,05161	4,59301	0,24342	1,98206	2,67043	0,08901	0,02939	1,52990	2,81226
ambiente	0,08368	1,51455	1,85247	0,10416	2,09070	0,36380	3,35152	0,73328	0,32957	0,23481	5,27361	0,29521	2,47004	4,56122	0,20320	0,01144	1,46708	2,20847

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
capacidade	0,24387	1,54171	1,95415	0,27314	2,13055	0,34952	7,38781	0,98257	0,14537	0,11770	3,90326	0,29149	1,94082	1,82784	0,15168	0,00964	1,45319	2,32347
fase	0,23596	2,10404	1,18424	0,07658	1,94735	1,96548	4,46210	0,82873	0,34867	0,04884	5,62511	0,35813	2,13310	2,31462	0,03754	0,01542	1,48037	2,75853
aumento	0,17583	1,90514	1,82192	0,10572	1,05706	1,17947	5,29310	1,72963	0,24598	0,13701	4,90551	0,49014	2,92361	0,95918	0,06968	0,00964	1,43804	2,31352
criação	0,14391	1,37311	2,18203	0,09907	1,70041	0,25900	3,94006	0,53515	0,10788	0,13397	5,86615	0,25664	1,44774	4,77388	0,28788	0,03581	1,44642	2,36934
nível	0,25148	2,84463	2,25435	0,14077	1,78746	0,83017	4,59597	1,67871	0,20390	0,24123	3,86900	0,50006	1,79759	2,35429	0,12561	0,00000	1,46720	2,01171
adequado	0,09788	2,47603	1,76236	0,10152	1,31117	2,27908	3,39298	1,39995	0,18149	0,16282	5,01359	0,33830	3,26113	1,48354	0,03740	0,01046	1,45685	2,10779
meio	0,11764	2,00804	1,72891	0,09771	1,79192	1,45661	3,22670	1,03877	0,44135	0,38994	5,90336	0,19185	2,55383	1,97305	0,14549	0,02328	1,44303	1,95503
tamanho	0,14247	1,31404	1,37330	0,00000	0,85935	1,49360	7,93577	0,83814	0,20035	3,34572	2,48307	0,31949	1,20887	0,87873	0,08016	0,02187	1,40593	3,10062
dispositivo	0,00000	0,05075	2,93676	0,12445	2,78858	0,22438	6,20374	1,59155	0,04553	0,00000	4,20623	0,00000	0,31971	3,45967	0,17554	0,02108	1,38425	2,33356
problema	0,16723	1,55186	2,59365	0,14507	2,22295	0,80843	4,36901	1,22801	0,14325	0,30538	3,55660	0,53985	2,02352	2,54623	0,20538	0,00000	1,40040	1,90067
existente	0,06514	2,11064	2,16246	0,18608	1,74461	0,71141	4,53087	1,11308	0,16725	0,41510	4,27125	0,20670	2,28974	2,48062	0,04330	0,01181	1,40688	1,87517
parâmetro	0,10390	2,83869	1,81346	0,07371	1,53389	0,76641	3,99735	2,65931	0,30076	0,07409	5,08335	0,29536	2,62695	0,86307	0,14302	0,03728	1,45066	2,13999
alteração	0,05159	2,12522	1,83862	0,09691	1,03325	0,89436	5,50151	1,73828	0,15085	0,34880	3,89326	0,60563	2,29095	1,82244	0,05020	0,00000	1,40262	2,04527
montagem	0,00000	0,53268	4,09837	0,12866	2,55051	0,04683	8,95704	1,33017	0,02977	0,46527	2,76174	0,02347	0,36582	0,67853	0,01430	0,00000	1,37395	2,61494
plataforma	0,00000	0,23801	1,55539	0,17104	2,39338	0,47413	3,33293	0,06520	0,06769	0,00000	6,73808	0,00000	0,85117	5,71500	0,25219	0,00000	1,36589	2,64796
especificação	0,02150	1,75812	2,24461	0,09265	1,77311	0,51383	4,12596	2,53585	0,12468	0,08900	4,75071	0,46078	2,00988	1,96335	0,06034	0,01463	1,40869	2,06354
capaz	0,08055	1,30704	2,11253	0,14151	1,97703	0,87307	5,02000	1,04714	0,35296	0,07141	4,50176	0,22671	1,79351	2,24096	0,12594	0,00000	1,36701	2,26754
propriedade	0,03848	1,98737	1,53940	0,11920	0,55193	0,85129	2,18937	3,00574	0,41318	0,09114	4,98312	0,44978	5,46146	0,62934	0,02303	0,04546	1,39871	2,10618
produtivo	0,30598	3,24476	2,07535	0,09776	0,97123	2,13247	4,61836	1,68411	0,15112	0,11532	3,52133	0,81157	2,15488	0,65966	0,00000	0,00964	1,40960	1,94584
óleo	0,00000	3,10969	1,17563	0,05100	1,10171	0,27962	3,12998	0,68740	0,58468	0,00000	9,20459	0,14687	2,48424	0,00000	0,00000	0,00000	1,37221	3,26446
interno	0,04651	1,33854	2,70719	0,11477	1,78978	0,39989	4,66125	2,76390	0,07120	0,47783	3,55841	0,13294	1,63433	1,45701	0,18928	0,00000	1,33393	1,97325
motor	0,00000	0,13370	6,16106	0,06217	1,36403	0,00000	9,47888	0,76122	0,00000	0,02106	1,85667	0,00000	0,28814	0,62205	0,00000	0,00000	1,29681	3,12814
integração	0,01720	0,30677	1,44892	0,24212	2,65972	0,08524	3,21746	0,20654	0,03225	0,00000	5,55683	0,13681	0,41497	6,15287	0,19770	0,00000	1,29221	2,48948
laboratório	0,11558	2,47000	1,88823	0,26113	1,70335	0,67020	3,31968	1,42783	0,40233	0,37170	4,52000	0,65956	3,56450	0,67991	0,12002	0,01046	1,38653	2,30881
dificuldade	0,04903	1,96845	2,12687	0,14597	1,53179	1,34327	3,38073	0,87229	0,13888	0,03370	4,52453	0,31613	2,51883	1,61746	0,07467	0,00000	1,29016	3,15081
carga	0,00000	0,63142	2,09130	0,35881	1,65886	0,33328	7,17683	1,23515	0,07005	0,15119	3,33714	0,33338	1,69009	1,53489	0,12015	0,00000	1,29516	2,16944
viabilidade	0,04018	2,14603	1,96767	0,11860	1,36900	1,02131	4,08519	1,03388	0,35126	0,14038	4,93402	0,43970	1,96298	1,31268	0,15685	0,00000	1,31748	2,20097
matéria	0,03424	2,96713	1,74476	0,07089	0,77817	0,43156	2,71195	1,44563	0,02867	0,24197	5,72566	0,42174	4,64528	0,03941	0,00000	0,02646	1,33210	2,13126
ponto	0,00000	1,26274	1,97812	0,12691	1,94449	0,82750	4,15044	1,02701	0,13897	0,25821	3,53051	0,38032	2,04862	2,62422	0,01863	0,01633	1,27081	1,71058
atividade	0,07117	2,61594	1,42985	0,08635	2,05256	1,08288	3,09424	1,05948	0,12557	0,09698	4,91013	0,40907	1,43717	2,90125	0,13544	0,03324	1,34633	2,00716
inovação	0,04306	1,63010	1,82283	0,05028	1,94493	0,77967	2,94876	0,94992	0,18155	0,13043	4,51402	0,19010	2,40266	2,43379	0,14019	0,00000	1,26014	2,12118
fim	0,09894	1,79606	1,45603	0,17083	1,59819	0,89825	3,80593	1,13173	0,16174	0,10484	4,83097	0,22413	2,08873	2,05661	0,01708	0,01486	1,27843	1,82005

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
relação	0,06883	1,95602	1,95464	0,17466	1,47092	0,96506	3,56422	1,44834	0,30061	0,10287	4,07656	0,38447	2,40572	1,28686	0,03361	0,00000	1,26209	1,66790
padrão	0,01678	2,62601	1,72349	0,19957	1,75684	1,24730	2,73676	0,64393	0,17065	0,37879	3,38061	0,26235	1,88971	3,45029	0,13640	0,00852	1,28925	1,66205
impacto	0,15230	2,30796	0,96248	0,13846	1,56934	0,50889	2,91002	1,00304	0,30596	0,05483	5,52673	0,57181	2,61844	1,37453	0,04232	0,01110	1,25364	1,84695
ativo	0,05590	0,87189	1,22758	0,09963	1,04499	4,68898	0,19716	0,09016	0,06708	0,00000	7,86931	0,18164	2,23920	1,19200	0,00000	0,00000	1,23910	2,98950
primo	0,03355	2,81945	1,65453	0,05234	0,73930	0,42079	2,55511	1,40333	0,00000	0,23663	5,60466	0,38703	4,38641	0,06714	0,00000	0,02646	1,27417	2,07317
embalagem	0,00000	4,36154	0,48206	0,00000	0,28628	2,32664	1,33541	0,87883	0,00000	0,16258	7,33587	1,05197	2,68208	0,00000	0,00000	0,00000	1,30645	2,49043
atual	0,06027	2,14343	2,83888	0,09224	1,46432	0,27345	3,41172	1,30208	0,14552	0,15898	3,54732	0,38101	2,17192	1,79777	0,12295	0,00000	1,24449	1,58396
energia	0,02372	1,40796	2,58597	0,10737	4,45710	0,12217	3,44707	0,88680	0,09993	0,63761	3,16427	0,26355	1,20021	0,77607	0,13974	0,00000	1,20747	2,02703
item	0,00000	1,28645	0,62072	0,01324	1,37612	0,09507	9,25285	0,59977	0,00000	3,09780	1,56604	0,11265	0,29914	0,99845	0,04117	0,00990	1,21059	3,15825
experimental	0,24916	1,84723	1,58309	0,08354	1,13904	0,49962	3,37600	3,45046	0,36392	0,10320	2,88427	0,41893	1,58296	1,82361	0,11658	0,00000	1,22010	2,04930
completo	0,01859	4,23594	0,96910	0,06279	1,05118	0,74705	5,31043	0,46611	0,16447	0,06816	1,25209	0,06065	0,46003	4,71668	0,02196	0,00000	1,22533	2,11335
identificação	0,10237	2,22941	1,35771	0,03616	1,34833	0,60773	3,36267	0,57136	0,18677	0,04876	4,94599	0,33081	1,67584	2,34936	0,15576	0,00000	1,20681	1,75700
função	0,07028	1,01767	2,27343	0,09822	2,14132	0,59730	3,80179	1,43557	0,14711	0,13512	3,01746	0,20735	1,54047	1,81325	0,05279	0,00000	1,14682	1,64510
composição	0,02219	2,22358	1,66976	0,16599	0,34740	0,52103	2,22084	2,67568	0,33738	0,11925	5,46704	0,38688	2,54621	0,13020	0,01757	0,00000	1,17819	1,87958
comunicação	0,00000	0,20434	1,73610	0,01468	3,21990	0,08515	3,14818	0,30673	0,06390	0,00000	4,06453	0,03249	0,27506	4,77671	0,23237	0,00000	1,13501	2,15221
ar	0,08055	1,40072	1,81337	0,11059	0,92965	0,82640	4,96037	0,56010	0,14218	0,03839	3,53042	0,36163	3,15540	0,53496	0,01662	0,00852	1,15437	1,70109
quot	0,10318	2,40854	0,60165	0,02175	0,89853	0,28746	1,97189	1,91202	0,12144	0,00000	8,06704	0,14263	1,70422	1,07822	0,04392	0,00000	1,21016	2,75035
térmico	0,00000	1,20585	3,81528	0,16349	1,49226	0,31104	3,95281	2,67438	0,34576	0,04492	2,14609	0,29526	1,34751	0,11010	0,03418	0,01397	1,12206	1,83581
técnica	0,09787	1,00624	2,09084	0,08802	2,22397	0,92683	1,64337	0,95891	0,15297	0,16514	4,83181	0,18637	0,83776	2,90091	0,05622	0,03719	1,13778	1,82096
físico	0,12888	1,90162	1,49701	0,07537	1,97967	0,84944	2,95338	0,70591	0,21835	0,44462	3,63069	0,50018	2,45934	0,92468	0,07451	0,00990	1,14710	1,48404
equipe	0,06297	0,71882	1,88927	0,20488	2,18882	0,41905	4,56054	0,46940	0,07066	0,40859	3,07197	0,17693	0,97024	2,63378	0,15234	0,00990	1,12551	1,89773
simulação	0,01911	0,46518	2,42634	0,06662	2,37207	0,00000	4,87523	1,86306	0,17426	0,22582	2,93260	0,12737	1,24908	1,13596	0,05495	0,00000	1,12423	2,38901
trabalho	0,02150	1,28443	1,95838	0,12945	1,70726	0,16619	5,10006	0,80449	0,04946	0,12510	3,23043	0,25307	0,95893	2,20772	0,06372	0,00964	1,12936	1,78434
implementação	0,02372	2,08854	1,93087	0,16485	2,17753	0,11534	2,53198	0,63374	0,16399	0,10352	3,90058	0,17831	0,62602	3,48329	0,16755	0,00000	1,14311	1,75136
perda	0,16583	1,85009	2,15045	0,10720	1,57936	0,88668	2,46182	1,10969	0,22675	0,05908	3,34126	0,56772	1,91031	1,21927	0,07354	0,02454	1,10835	1,67532
elaboração	0,00000	1,24275	2,29309	0,18466	1,46489	0,85825	4,88868	0,78045	0,00000	0,40054	3,02204	0,15082	1,31879	1,34085	0,09348	0,01673	1,12850	2,05334
geração	0,26795	2,05685	1,62336	0,06106	3,04672	0,54625	2,39023	0,63809	0,18053	0,04084	3,22007	0,39073	1,40190	2,32835	0,10120	0,00000	1,14338	1,57596
concepção	0,06961	1,59270	2,57274	0,24385	1,57338	0,64340	3,77020	0,70621	0,07886	0,12480	3,16117	0,06162	1,09764	1,58222	0,16455	0,00000	1,09019	1,64811
efeito	0,25687	2,35796	1,58594	0,06222	1,17879	1,72941	1,54889	0,94171	0,52672	0,01872	3,85141	0,50664	2,48903	0,54863	0,01662	0,05422	1,10461	1,51014
vez	0,05056	1,72014	1,40355	0,09416	1,01477	1,47114	2,98589	1,02648	0,20561	0,14619	3,41489	0,41945	1,47927	1,88537	0,03344	0,01665	1,08547	1,63791
parte	0,06591	0,95674	2,36514	0,09681	1,54782	0,99800	3,68530	0,74912	0,00000	0,05485	3,04251	0,59242	1,20439	1,88223	0,07430	0,01263	1,08301	1,55437
algoritmo	0,00000	0,10036	1,01609	0,02172	2,70028	0,07146	1,94196	0,29315	0,06870	0,00000	4,91658	0,02485	0,27169	5,45526	0,24174	0,00000	1,07024	2,34823

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
conjunto	0,05246	0,63343	2,14702	0,21355	1,37493	0,39273	4,43175	1,39242	0,10704	0,19654	3,19219	0,16472	0,97627	1,85748	0,07283	0,00000	1,07534	1,60416
funcionalidade	0,00000	0,64337	1,24719	0,02011	2,32629	0,09792	2,85758	0,37279	0,00654	0,05880	4,08511	0,06480	0,92264	4,18526	0,18301	0,01568	1,06794	1,75137
diferente	0,16566	2,07459	1,64682	0,09106	0,96648	0,81282	2,59003	0,89722	0,14272	0,13251	3,27082	0,34810	2,39126	1,79197	0,06814	0,00000	1,08689	1,56239
quantidade	0,08935	2,64649	2,05096	0,12413	0,93195	0,61534	3,17491	0,94986	0,17653	0,09096	2,79790	0,34370	2,13107	1,37347	0,04510	0,02358	1,09783	1,53127
eletrônico	0,00000	0,15945	2,26286	0,13973	3,30986	0,14038	5,66376	0,77880	0,05902	0,00000	2,30912	0,15018	0,28071	1,52027	0,03362	0,01144	1,05120	1,83188
piloto	0,01859	2,25830	1,04478	0,08787	1,66727	1,55740	1,68860	1,59143	0,35125	0,11614	4,09910	0,25301	2,85008	1,07051	0,10034	0,00000	1,17217	2,32558
processamento	0,08078	1,46084	0,67625	0,08038	1,72566	0,10789	1,71097	1,00615	0,15862	0,02106	3,98094	0,20410	1,53176	4,08581	0,16186	0,00000	1,06207	1,70624
modo	0,07712	1,18486	1,84493	0,12796	1,55156	0,75150	4,33525	1,14222	0,16710	0,02174	2,74919	0,03520	1,22185	1,44341	0,12515	0,00916	1,04926	1,51670
industrial	0,11050	3,14974	1,38603	0,05947	1,10184	1,52941	1,29714	1,27350	0,42169	0,20093	3,37799	0,47823	3,60973	0,20510	0,01708	0,01463	1,13956	1,63801
usuário	0,00000	0,19981	1,35298	0,07985	2,14828	0,07456	1,59465	0,18573	0,00000	0,10131	4,18196	0,03249	0,59132	5,31441	0,20860	0,00000	1,00412	2,02342
fórmula	0,01911	2,84480	0,24913	0,00000	0,00000	1,22693	0,80124	0,10630	0,03877	0,00000	7,83314	0,02112	3,27611	0,47939	0,00000	0,00000	1,05600	2,38941
diverso	0,12502	1,56855	1,33419	0,19046	1,36345	0,42852	3,43058	0,82929	0,14743	0,06366	2,99756	0,29813	1,29386	2,34185	0,05866	0,00000	1,02945	1,50404
norma	0,00000	0,32609	2,92537	0,09158	1,56406	0,39180	2,77631	1,91664	0,02419	0,23826	2,75674	0,10067	2,08756	0,57618	0,03169	0,00000	0,98795	1,50721
pressão	0,09798	1,12437	2,04587	0,12341	0,48245	0,37034	4,77663	0,99056	0,56886	0,07013	3,17343	0,33686	1,83689	0,03559	0,00000	0,00939	1,00267	1,73205
veículo	0,00000	0,42055	0,52889	0,01802	1,09864	0,80540	8,44328	0,16017	0,02212	0,00000	2,59485	0,00000	0,37819	1,10403	0,11044	0,01110	0,98098	2,16482
consumo	0,06561	2,46647	1,41819	0,08071	1,86337	0,15767	3,22535	0,83500	0,08032	0,01982	3,25582	0,33193	0,80180	1,65679	0,08959	0,01465	1,02269	1,55906
rede	0,00000	0,43571	1,58262	0,02102	4,53045	0,06985	1,91834	0,05523	0,05442	0,00000	3,21968	0,16728	0,30901	2,86822	0,41468	0,00000	0,97791	1,92243
escala	0,06769	2,06717	0,69905	0,04769	0,80481	1,92768	1,30646	1,44877	0,37853	0,05118	4,08397	0,27893	3,11532	0,59156	0,00000	0,00000	1,05430	1,89873
aço	0,00000	0,14191	1,77457	0,09409	0,37260	0,02773	3,92224	4,39286	0,07115	0,15619	3,68685	0,10587	0,80658	0,03448	0,00000	0,00000	0,97419	2,10958
químico	0,16739	1,76543	0,82636	0,12051	0,78100	0,70554	1,24176	1,19001	0,26609	0,10627	4,90006	0,73339	3,27301	0,06923	0,00000	0,01394	1,01000	1,78090
manutenção	0,07542	1,30612	1,47128	0,01358	1,49822	0,62557	2,91719	0,45082	0,04304	0,01982	2,31554	0,17648	3,04224	1,43694	0,08934	0,00893	0,96816	1,41146
fluxo	0,04299	0,83242	1,89533	0,07685	1,63217	0,54224	3,35292	0,69625	0,05189	0,02246	3,23292	0,33370	0,59815	2,35170	0,08638	0,00916	0,98485	1,44463
estrutural	0,00000	0,38695	1,28333	0,18953	0,51583	0,29639	8,49321	0,55588	0,02419	0,80563	2,02431	0,02640	0,53378	0,28650	0,02675	0,00000	0,96554	2,40252
cálculo	0,00000	0,41885	2,39832	0,20688	1,58726	0,05546	3,93894	1,07252	0,04497	0,10652	3,09919	0,14931	0,47141	1,86604	0,08824	0,00000	0,96899	1,90763
falha	0,00000	0,30648	1,94999	0,13829	2,29109	0,43930	3,93744	0,56854	0,00000	0,04574	3,15677	0,06408	0,65503	1,73259	0,08100	0,01077	0,96107	1,65427
corte	0,08845	2,56968	2,79770	0,00000	0,32283	0,34773	5,05837	1,43807	0,05529	0,18014	1,64601	0,28881	1,07628	0,09591	0,01576	0,00000	0,99881	1,68472
construção	0,00000	0,31132	1,31384	0,17858	1,40298	0,26466	3,23445	0,83057	0,13535	0,05722	3,45799	0,18685	0,71360	3,31484	0,12169	0,00000	0,97025	1,73054
fonte	0,07502	2,42778	1,71492	0,01597	1,47374	0,15383	0,80904	0,11352	0,00000	0,00000	4,79789	0,15002	2,01425	1,60677	0,19739	0,00000	0,97188	1,63772
resina	0,00000	0,17226	1,35101	0,00000	0,37752	0,07903	3,35919	0,49843	0,37741	0,08008	1,62689	0,16548	7,04584	0,00000	0,00000	0,01993	0,94707	2,05880
disponível	0,02751	1,37476	2,08403	0,06326	1,34299	0,68387	1,72348	0,64638	0,16644	0,09709	2,66217	0,24382	2,58839	1,54306	0,08680	0,00000	0,95838	1,23715
interface	0,00000	0,06437	1,09143	0,18660	2,33013	0,03011	3,49476	0,25377	0,06118	0,00000	2,99586	0,00000	0,34510	3,92311	0,30613	0,00000	0,94266	1,63510
instalação	0,00000	0,80336	1,14770	0,05160	2,20895	0,28216	5,83544	0,42557	0,10598	0,02407	2,10276	0,23673	1,10882	0,90777	0,07908	0,00000	0,95750	1,63776

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
planta	0,46592	2,74524	0,54609	0,10855	0,74237	0,92139	1,42591	0,33527	0,39167	0,11890	3,42946	0,53499	4,61650	0,07426	0,03377	0,00000	1,03064	1,59588
medição	0,05291	0,62097	1,89788	0,03130	2,96851	0,42666	4,04638	0,83135	0,14532	0,00000	1,82324	0,32920	1,15024	0,76557	0,04993	0,00000	0,94622	1,60418
único	0,07849	0,88076	1,12213	0,10542	1,61760	1,01402	2,59698	0,53028	0,04435	0,10722	2,73564	0,10256	1,04955	2,84198	0,15458	0,01110	0,93704	1,48974
comportamento	0,14984	0,86560	1,54789	0,17120	1,24043	0,50953	3,71969	1,07317	0,32956	0,04067	2,79632	0,32797	1,32538	1,06266	0,05930	0,00000	0,95120	1,44310
desenho	0,02150	0,81145	1,98919	0,07349	0,64689	0,56316	4,37034	0,55938	0,10786	0,19324	3,70493	0,37427	0,60700	1,27059	0,07202	0,00990	0,96095	1,97854
velocidade	0,00000	0,96043	1,95028	0,01392	1,18460	0,28377	4,33374	1,33788	0,10727	0,00000	2,34195	0,32905	1,07995	1,10025	0,03516	0,00000	0,94114	1,45892
ajuste	0,03127	1,46796	1,55214	0,02035	0,99717	1,03149	2,84599	1,07467	0,10188	0,04347	2,30481	0,50150	2,51713	0,83413	0,05259	0,00000	0,96103	1,38759
possibilidade	0,06253	1,05361	1,35808	0,12639	0,94555	0,28333	2,82366	0,78145	0,05981	0,33527	3,59565	0,30853	1,22591	1,57098	0,06665	0,00000	0,91234	1,33177
módulo	0,00000	0,49448	1,45621	0,11447	2,45046	0,00000	3,48159	0,22591	0,07506	0,11905	2,46320	0,08010	0,42143	2,91459	0,13977	0,00000	0,90227	1,59563
fornecedor	0,00000	2,73655	1,84156	0,10740	0,85658	0,82988	3,10125	0,65891	0,06349	0,14485	2,74634	0,18120	1,58177	1,14424	0,02196	0,01144	1,00171	1,55327
obtenção	0,29308	1,40662	1,00443	0,03985	0,55680	1,65221	1,36002	0,84975	0,38059	0,04765	3,91788	0,40574	2,21365	0,50948	0,01863	0,04272	0,91869	1,44076
peso	0,10248	2,69871	1,31226	0,10050	0,58775	0,36425	4,76891	0,59642	0,19183	0,16352	2,30588	0,06348	2,10923	0,25920	0,00000	0,01263	0,97731	1,62014
vida	0,05758	1,65401	2,04252	0,11275	1,40296	0,50215	2,24030	0,97655	0,09748	0,04478	3,26992	0,04080	1,49425	0,79160	0,00000	0,01465	0,92139	1,26807
atendimento	0,00000	0,70161	0,91999	0,06970	1,02507	0,26836	2,32963	1,99261	0,03874	0,15853	3,30188	0,00000	1,07234	2,30159	0,06645	0,00000	0,89041	1,42706
combinação	0,11261	2,28585	0,67583	0,02252	0,46511	2,41321	1,27459	0,37278	0,15774	0,06105	4,10022	0,19011	1,86117	0,70381	0,00000	0,00000	0,91854	1,66932
arquitetura	0,08413	0,15201	0,43370	0,00000	1,74812	0,00000	3,03426	0,07090	0,06130	0,02324	3,49117	0,00000	0,11547	4,67189	0,23770	0,00000	0,88274	1,79422
automático	0,02219	0,99851	1,80223	0,00000	1,26502	0,40925	3,78437	0,73662	0,09865	0,02106	2,55172	0,08869	0,24492	1,89831	0,10763	0,00964	0,87743	1,47784
operacional	0,09439	0,43137	1,20897	0,03153	1,87259	0,20415	2,43174	0,69987	0,25730	0,00000	2,92148	0,51586	1,26385	2,03889	0,02049	0,01046	0,87518	1,24671
brasil	0,15892	1,37932	1,03995	0,02628	1,37832	0,69710	2,81071	0,38677	0,02277	0,03746	3,13089	0,32190	1,67438	0,93339	0,01808	0,01521	0,87697	1,26697
adequação	0,00000	1,64998	1,34466	0,02175	0,91237	1,32657	3,28491	1,00813	0,09607	0,06197	1,81259	0,36017	1,21518	0,96815	0,04867	0,00000	0,88195	1,22743
real	0,06518	0,62804	1,29277	0,07285	1,76454	0,28704	2,40868	0,36406	0,15874	0,05675	3,20019	0,05922	0,66031	2,59940	0,13627	0,00000	0,85963	1,38606
proteção	0,04544	0,32942	2,39901	0,16388	2,15950	0,56559	2,35860	0,32423	0,04994	0,06672	3,00075	0,26961	1,27326	0,44203	0,04099	0,04385	0,84580	1,44262
longo	0,19235	1,48845	1,07008	0,06914	1,28433	1,25406	2,53744	0,64817	0,20478	0,03797	2,17419	0,21817	2,06860	0,52335	0,10280	0,00000	0,86712	1,31176
amostra	0,08777	2,00272	1,00146	0,06110	0,46228	0,37182	1,49735	0,98991	0,39485	0,03744	4,50296	0,42090	2,56314	0,46178	0,00000	0,00000	0,92847	2,04804
primeiro	0,10715	1,43287	1,32504	0,02628	1,54581	1,48594	2,59963	0,70002	0,14041	0,07590	2,03954	0,22157	0,98636	1,40053	0,01808	0,00000	0,88157	1,19774
distribuição	0,03141	1,26700	1,54288	0,16602	2,26941	0,61790	2,42194	0,61628	0,19269	0,03797	2,12831	0,11294	0,84200	1,32850	0,00000	0,00000	0,84845	1,21543
volume	0,00000	1,39072	0,99135	0,18547	0,43233	0,38589	1,86517	0,40136	0,07155	0,05085	3,75781	0,09499	1,27095	2,64977	0,05455	0,00000	0,85017	1,57496
variação	0,10238	1,99513	1,89493	0,22309	0,91221	0,57210	2,42047	1,06596	0,17133	0,11899	2,08529	0,27188	1,34186	0,30643	0,09882	0,01144	0,84952	1,19882
funcional	0,00000	0,70590	1,09572	0,10531	1,17914	0,30087	3,73488	0,42625	0,05323	0,15946	2,99516	0,04526	0,65802	1,77674	0,12865	0,00990	0,83591	1,59937
execução	0,00000	0,58413	1,41942	0,25306	1,31066	0,35764	2,02942	1,02231	0,04511	0,03243	2,51884	0,00000	0,41444	3,22787	0,11018	0,01915	0,83404	1,33845
líquido	0,00000	2,19959	0,44280	0,00000	0,39287	0,76563	1,94348	0,80569	0,19132	0,00000	3,98150	0,05957	2,70672	0,20017	0,00000	0,00000	0,85558	1,54239
química	0,02372	0,72125	0,44291	0,03297	0,15420	0,84724	1,54682	2,39179	0,24248	0,02042	3,82869	0,24948	3,10263	0,00000	0,00000	0,00852	0,85082	1,53385

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
ação	0,12620	1,62931	0,72900	0,00000	1,19660	1,41247	1,11487	0,72890	0,03518	0,04317	3,54778	0,18665	1,87852	1,28298	0,00000	0,00000	0,86948	1,30067
placa	0,00000	0,51328	2,49142	0,11308	2,69398	0,02773	2,08060	1,56047	0,00000	0,03642	2,22333	0,02725	0,56703	0,74354	0,19394	0,00000	0,82951	1,35333
serviço	0,00000	0,08881	0,27939	0,20303	1,24684	0,18922	0,93903	0,06911	0,00000	0,02106	4,19978	0,00000	0,51124	4,91986	0,35682	0,00000	0,81401	1,84598
funcionamento	0,00000	0,66077	1,83954	0,13024	1,04466	0,35705	4,10398	0,58348	0,13364	0,04249	1,86101	0,12853	0,48473	1,65541	0,12855	0,01263	0,82292	1,33655
acordo	0,15353	0,87652	0,93096	0,07290	0,82176	1,52728	2,32862	0,47665	0,07686	0,03914	2,45345	0,27945	1,56969	1,62358	0,04160	0,02089	0,83081	1,12780
massa	0,00000	3,66469	1,24183	0,20142	0,22568	0,28396	1,70015	0,69665	0,22433	0,00000	2,42165	0,25961	2,30036	0,52158	0,00000	0,00000	0,85887	1,39384
demanda	0,04316	0,74222	0,66549	0,07182	1,09025	0,09396	3,43931	0,68924	0,02419	0,04492	3,66418	0,19801	0,82101	1,16769	0,07400	0,00000	0,80184	1,36332
ganho	0,07198	1,61527	1,45786	0,05568	0,86589	0,15707	3,27138	0,99124	0,00654	0,05103	1,55895	0,57852	1,61639	0,91941	0,09470	0,01181	0,83273	1,24496
ideal	0,04136	2,26702	1,18261	0,13028	0,15913	0,66204	1,74195	0,79428	0,14080	0,18055	3,30142	0,31995	1,95661	0,23008	0,00000	0,03488	0,82144	1,38258
partir	0,04175	1,16511	0,69407	0,04857	1,61062	0,68531	1,82030	1,16141	0,17685	0,05291	3,04001	0,19501	1,39609	0,97374	0,05202	0,00893	0,82017	1,26448
agente	0,03424	0,67634	0,33927	0,03531	1,02474	1,03657	0,33136	0,30602	0,00000	0,00000	6,06894	0,44386	2,04987	0,29602	0,00000	0,00000	0,79016	1,75299
variável	0,11021	0,91975	1,79210	0,05364	1,20198	0,38214	1,98303	0,74260	0,11530	0,05394	2,75508	0,38337	0,99984	1,31645	0,01808	0,00000	0,80172	1,20878
perfil	0,02293	2,17941	0,64765	0,03945	0,51752	1,83635	2,29183	1,33498	0,00000	0,06585	2,52884	0,15493	0,85521	0,58556	0,00000	0,00000	0,81628	1,28149
mecanismo	0,10751	0,26652	0,97529	0,01429	0,90489	0,26382	2,59396	0,60295	0,08277	0,13633	3,26183	0,00000	0,56534	2,51155	0,11305	0,00964	0,77561	1,37057
eficácia	0,10205	0,68883	0,45425	0,00000	0,31365	4,24143	0,95221	0,07302	0,00000	0,00000	4,03920	0,18069	1,33098	0,19563	0,01479	0,00000	0,78667	1,98331
seleção	0,23877	1,15176	0,65965	0,04951	0,65356	0,84487	2,69033	0,17342	0,24938	0,01872	3,50469	0,44551	1,24484	0,66106	0,01576	0,00893	0,78817	1,62526
recurso	0,00000	0,41828	0,99561	0,11021	1,43094	0,42638	1,65103	0,51007	0,02497	0,12643	3,05842	0,25676	0,38402	2,94104	0,13365	0,00000	0,77924	1,30427
ciclo	0,23443	0,34158	1,36263	0,08457	0,84459	0,14520	2,30053	1,70064	0,07050	0,03151	1,86825	0,10413	1,14766	1,98309	0,03892	0,00000	0,76614	1,22371
lote	0,00000	1,21418	1,12654	0,04852	0,79954	1,92862	1,01901	1,39420	0,05458	0,12176	2,75814	0,32528	1,55081	0,60205	0,01500	0,00000	0,80989	1,95742
camada	0,00000	0,86002	0,83947	0,11736	0,53986	0,31369	1,87153	0,61473	0,11898	0,01872	2,92999	0,07160	1,28798	2,55533	0,04480	0,00000	0,76150	1,31205
injeção	0,00000	0,54624	2,02044	0,08066	0,79270	0,13950	3,13578	1,33124	0,25976	0,21351	2,91700	0,00000	0,59099	0,06076	0,00000	0,00000	0,75554	1,41047
ingrediente	0,03876	5,37562	0,00000	0,00000	0,00000	0,42101	0,03691	0,00000	0,00000	0,00000	5,65747	0,00000	2,10891	0,00000	0,00000	0,00000	0,85242	2,17784
rápido	0,00000	0,88117	1,00545	0,03030	0,79029	0,63442	2,51058	0,50359	0,09479	0,01821	2,26004	0,00000	1,50315	1,73034	0,03479	0,01110	0,75051	1,22940
mistura	0,00000	3,75968	0,70220	0,21437	0,19571	0,97293	0,85749	0,65075	0,54153	0,00000	2,95822	0,27716	1,97212	0,07284	0,00000	0,00000	0,82344	1,50035
experimento	0,26436	2,00614	1,58919	0,01697	0,18806	0,75228	2,24372	1,07512	0,17405	0,00000	2,12614	0,42620	1,36514	0,64037	0,00000	0,00000	0,80423	1,60579
eficiente	0,09788	1,10101	1,14195	0,13193	1,04079	0,30851	2,12614	0,47740	0,27358	0,03254	2,23513	0,25569	1,45915	1,12643	0,06585	0,01110	0,74282	1,11359
implantação	0,03609	0,64721	1,52636	0,13407	1,99657	0,25626	1,53636	0,18635	0,02277	0,11592	2,54560	0,25344	0,94844	1,76061	0,15053	0,00964	0,75789	1,17862
ano	0,17579	1,06256	0,76166	0,03504	0,97847	1,26423	1,66045	0,59089	0,15280	0,12292	3,07621	0,19757	0,98482	0,97207	0,01863	0,00000	0,75338	1,40037
externo	0,02150	0,82522	1,92468	0,08244	0,72569	0,30151	2,44695	0,81274	0,06130	0,10460	2,12503	0,25876	1,16142	0,92560	0,09475	0,00000	0,74201	1,10363
otimização	0,00000	1,19573	1,04132	0,06004	1,24086	0,69865	1,70811	0,78188	0,12860	0,05417	2,11131	0,28884	1,11558	1,44289	0,14637	0,00964	0,75150	1,00226
tensão	0,00000	0,09936	3,33648	0,12339	2,79932	0,07152	2,39364	0,62447	0,04470	0,02174	1,16972	0,02640	0,63170	0,25363	0,03051	0,01415	0,72755	1,51896
bancada	0,00000	1,01891	0,87023	0,01358	0,48686	1,22721	3,13527	0,83159	0,14059	0,01925	2,75282	0,08147	1,76686	0,12408	0,01500	0,00000	0,78023	1,79092

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
cor	0,06610	2,97205	1,69912	0,00000	0,03385	0,15619	0,76312	0,31300	0,00000	0,07155	3,73367	0,39734	1,82748	0,31812	0,00000	0,02241	0,77338	1,39378
formação	0,01859	1,91482	0,76655	0,05378	0,56014	0,54540	1,04757	1,47426	0,31056	0,12448	2,67208	0,26120	2,04600	0,19296	0,00000	0,01929	0,75048	1,24362
fixação	0,00000	0,25875	2,27272	0,08925	0,36113	0,28687	3,45937	1,11857	0,02580	0,30332	2,64454	0,17489	0,61893	0,00000	0,00000	0,01046	0,72654	1,53847
teor	0,00000	2,55123	0,34987	0,03912	0,18934	0,75236	0,06407	1,02224	0,59449	0,05725	3,67552	0,15759	2,51418	0,04086	0,00000	0,00000	0,75051	1,35342
nacional	0,04333	1,00308	1,24814	0,05587	1,71128	0,78421	1,63679	0,54720	0,13901	0,08242	1,83937	0,39197	1,32964	0,83103	0,04783	0,00852	0,73123	1,04726
banco	0,00000	0,46723	0,54741	0,01551	1,28741	0,00000	1,43796	0,22530	0,09343	0,00000	2,92119	0,04701	0,45966	3,95495	0,25347	0,01110	0,73260	1,35982
região	0,36537	0,82569	1,06279	0,23681	1,07103	0,04215	1,19018	0,85227	0,07898	0,02496	4,21778	0,20571	1,00310	0,45407	0,02673	0,00000	0,72860	1,38172
painel	0,00000	0,26405	1,64016	0,00000	1,19200	0,16080	3,83962	0,07025	0,02580	0,59372	2,81261	0,05120	0,40264	0,60812	0,01757	0,00000	0,72991	1,38972
matériasprima	0,00000	1,72442	0,64045	0,00000	0,00000	0,42149	2,77534	0,13831	0,15157	0,14995	2,88717	0,19566	3,13592	0,00000	0,00000	0,02099	0,76508	1,46510
valor	0,00000	1,52931	1,36177	0,08706	1,19502	0,23077	2,00694	0,50054	0,03686	0,06195	2,25700	0,28798	1,07820	1,03290	0,08232	0,00000	0,73429	0,98814
sensor	0,00000	0,54659	0,76742	0,00000	2,40416	0,00000	3,55322	0,20413	0,28538	0,00000	2,52039	0,05045	0,51508	0,59743	0,07426	0,01144	0,72062	1,34201
unidade	0,07167	0,67343	1,03384	0,05731	1,18547	0,23110	2,09782	0,57175	0,16479	0,05762	2,73383	0,41619	2,04657	0,44809	0,07645	0,00000	0,74162	1,10059
durabilidade	0,00000	0,63813	1,49354	0,02909	0,29932	0,06199	5,12597	0,77713	0,05669	0,28250	1,88571	0,07312	0,77482	0,11575	0,00000	0,00000	0,72586	1,48513
monitoramento	0,07703	0,49393	0,80818	0,06110	2,59210	0,07844	1,31932	0,38821	0,07372	0,05912	2,85404	0,27378	0,85755	1,62720	0,06371	0,02358	0,72819	1,25505
ambiental	0,20257	0,84938	0,53865	0,03818	1,66138	0,23493	1,16654	0,35526	0,10293	0,02174	4,35508	0,67500	1,13052	0,09519	0,00000	0,01110	0,71490	1,43398
configuração	0,00000	0,33639	1,73131	0,04261	1,10097	0,03011	2,65255	0,25755	0,06193	0,19750	2,04012	0,07610	0,50920	2,04603	0,24296	0,00000	0,70783	1,13263
programa	0,10038	1,40495	0,67059	0,08234	1,12997	0,36448	1,77937	1,18131	0,00000	0,00000	3,02514	0,02485	0,18714	1,47677	0,02685	0,00000	0,71588	1,16900
procedimento	0,09169	0,69652	1,25876	0,08920	1,33775	0,68823	2,00196	0,49774	0,09285	0,00000	2,69022	0,04229	0,80422	1,06236	0,05894	0,00000	0,71330	1,08778
determinação	0,00000	0,99029	0,90510	0,06469	1,05089	0,66282	2,02501	0,58819	0,18349	0,04413	2,59635	0,20156	2,36937	0,25070	0,00000	0,00000	0,74579	1,32012
revestimento	0,00000	0,13192	0,65039	0,22926	0,06571	1,42134	2,50923	2,41824	0,00000	0,17778	1,34632	0,14802	2,15227	0,00000	0,00000	0,00000	0,70316	1,31265
adaptação	0,15445	0,76947	0,74601	0,17764	0,79701	0,12138	3,62129	0,07731	0,11888	1,02751	1,48200	0,22420	1,13539	0,83192	0,07629	0,00000	0,71005	1,39459
índice	0,00000	1,00824	0,33313	0,24997	1,56810	0,30795	1,69137	0,32090	0,11798	0,08532	3,01441	0,10618	1,56198	0,98727	0,00000	0,00000	0,70955	1,12452
molde	0,00000	0,70382	1,64890	0,00000	0,51949	0,00000	3,59104	0,97217	0,15589	0,11751	3,08817	0,07345	0,48979	0,02758	0,00000	0,00000	0,71174	1,40908
alternativa	0,04586	1,24139	1,92152	0,05189	1,03658	0,26087	1,88778	0,49624	0,11627	0,04284	2,13235	0,28045	1,54631	0,40325	0,00000	0,00000	0,71648	1,08525
maneira	0,05238	0,93891	1,12703	0,03756	0,67029	0,30095	2,49426	0,65668	0,05954	0,09661	1,85522	0,23666	0,85776	1,63244	0,07446	0,00000	0,69317	1,00968
natural	0,09654	2,17053	0,65296	0,04207	0,82048	0,40028	1,10238	0,38097	0,22733	0,00000	2,26441	0,48009	2,05993	0,51665	0,03757	0,02567	0,70487	1,14283
barra	0,00000	1,64178	1,10015	0,09771	0,29653	0,08936	1,95870	3,18722	0,16768	0,00000	1,64707	0,00000	0,55261	0,52719	0,00000	0,00000	0,70412	1,38814
caso	0,04586	0,76626	1,27304	0,04639	1,08413	0,56730	1,62748	0,47647	0,14898	0,04008	2,13289	0,24082	1,08999	1,49381	0,08149	0,00000	0,69469	0,94768
gestão	0,02023	0,13276	0,65101	0,00000	1,58779	0,04053	0,79968	0,00000	0,00000	0,00000	3,53015	0,11650	0,15106	3,85120	0,09970	0,00000	0,68629	1,44763
substituição	0,04586	1,90634	1,04674	0,04627	0,88321	0,23622	2,16044	0,74649	0,05515	0,20914	1,98499	0,32004	1,16911	0,47167	0,04009	0,00000	0,70761	1,04025
fio	0,00000	0,04339	4,14538	0,00000	1,16007	0,05269	1,58499	0,36703	0,02346	0,02407	2,21939	0,22869	0,52768	0,46796	0,00000	0,05331	0,68113	1,60037
busca	0,20472	0,96114	1,21157	0,05973	0,59186	0,27295	2,13744	0,27345	0,06034	0,05776	2,06784	0,24928	1,20778	1,75499	0,06089	0,00000	0,69823	1,01719

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
tinta	0,00000	0,12363	1,66162	0,00000	0,06361	0,03011	1,48211	0,35910	0,11734	0,00000	1,60718	0,13916	5,13471	0,13740	0,00000	0,00852	0,67903	1,74003
protocolo	0,00000	0,22892	0,95599	0,03153	2,03276	0,78458	1,05404	0,13633	0,01985	0,00000	1,97272	0,18347	0,56976	2,63195	0,34109	0,00000	0,68394	1,15370
engenharia	0,01859	0,44963	1,97581	0,06133	0,69311	0,16020	3,57467	0,84349	0,05795	0,20401	1,69248	0,00000	0,52880	0,74532	0,06474	0,02089	0,69319	1,32221
mudança	0,06961	0,85402	0,46853	0,00000	0,68704	0,06602	3,01603	0,58658	0,11889	0,20863	1,76735	0,21139	0,94171	1,74317	0,05613	0,00000	0,67469	1,00945
concentração	0,03943	1,94602	0,40072	0,01971	0,21527	1,68069	0,64760	0,56070	0,13904	0,00000	2,98013	0,30310	2,30748	0,15768	0,00000	0,00000	0,71235	1,42220
aquisição	0,02978	1,80069	1,03753	0,04755	0,97385	0,39105	2,51132	0,34889	0,05338	0,15961	1,61765	0,02347	1,09842	0,54437	0,01863	0,00000	0,66601	1,24125
crítico	0,02219	0,75384	0,91441	0,02175	1,02999	1,00700	2,21253	0,61816	0,06451	0,04352	1,81741	0,12804	0,83128	1,27779	0,03856	0,01181	0,67455	0,99541
grupo	0,12882	0,73969	0,95309	0,01971	0,99140	1,35562	1,60202	0,50389	0,08665	0,03656	2,35976	0,18267	1,05426	1,06520	0,00000	0,00000	0,69246	1,01260
verificação	0,00000	1,21169	0,67059	0,08239	1,26049	0,11313	2,21931	1,10150	0,02346	0,08955	2,84512	0,15898	0,98865	0,78645	0,00000	0,01181	0,72269	1,44454
resíduo	0,04050	1,31864	1,04887	0,14151	0,47556	0,77267	0,78509	0,85003	0,22200	0,01925	3,59566	0,15184	1,25903	0,07118	0,03757	0,00000	0,67434	1,14847
brasileiro	0,18005	0,86369	0,36708	0,00000	0,77534	0,48556	1,74119	0,09252	0,00000	0,02407	3,94114	0,11209	1,16764	0,93809	0,04704	0,00000	0,67097	1,41219
presente	0,09069	1,13136	0,74991	0,02336	0,98035	0,61322	1,68209	0,40514	0,19745	0,07364	2,47331	0,36174	1,13855	0,77282	0,00000	0,01046	0,66901	0,97714
momento	0,01965	0,50566	0,98889	0,10907	0,69470	0,72973	1,65823	0,37928	0,10263	0,00000	2,73067	0,11415	1,38393	1,10244	0,01983	0,00000	0,65868	1,17751
número	0,13662	0,65920	0,96752	0,04418	0,79364	0,59237	1,54486	0,90973	0,05364	0,01773	1,92180	0,22940	1,39852	1,20646	0,12838	0,00000	0,66275	1,02556
transporte	0,01678	1,02417	0,53859	0,13292	0,67684	0,16454	2,99654	0,46790	0,14051	0,04352	2,60969	0,41230	0,88097	0,70323	0,02967	0,00000	0,67738	1,01446
modelagem	0,00000	0,22677	1,19845	0,01468	1,50588	0,09580	1,95533	0,27866	0,10314	0,06418	2,19478	0,10898	0,78853	1,87942	0,06003	0,00000	0,65466	1,16593
geometria	0,00000	0,08080	1,52154	0,09245	0,57253	0,00000	3,94945	2,22973	0,02346	0,10318	1,41906	0,06378	0,27154	0,07206	0,00000	0,00000	0,64997	1,44989
fibra	0,00000	1,89216	1,31342	0,09292	0,72821	0,12094	1,42815	0,10449	0,17716	0,06774	2,07198	0,68412	1,67809	0,27198	0,05803	0,05718	0,67166	1,02125
superfície	0,00000	0,54307	1,37347	0,12838	0,28152	0,17900	1,94449	1,17227	0,24094	0,05291	1,66074	0,28095	2,52349	0,08965	0,00000	0,00000	0,65443	1,15459
programação	0,00000	0,95144	1,36339	0,08803	0,97052	0,00000	2,20010	0,62107	0,00000	0,05394	2,15197	0,00000	0,14559	2,22954	0,05930	0,00000	0,67718	1,16464
manual	0,02150	0,47233	1,65984	0,04705	1,27842	0,27640	2,35030	0,61351	0,00000	0,09278	2,22657	0,08251	0,47785	0,83100	0,06486	0,00000	0,65593	1,08029
secagem	0,04744	2,19398	1,15314	0,02577	0,08770	1,84937	0,89256	0,10951	0,21755	0,04347	1,04107	0,35392	2,89407	0,00000	0,00000	0,00000	0,68185	1,39752
compatibilidade	0,00000	0,20497	0,69098	0,03288	0,59703	1,23015	1,02448	0,06963	0,17560	0,02246	3,28528	0,19647	1,66680	1,12033	0,07403	0,00000	0,64944	1,41693
fator	0,04817	1,65546	0,96367	0,15338	1,26186	0,66848	1,94419	0,19212	0,07221	0,04352	1,75124	0,23537	1,14495	0,39565	0,04911	0,01308	0,66203	1,04084
limpeza	0,00000	0,60801	0,92448	0,00000	0,40880	0,21271	1,05952	0,91686	0,06130	0,00000	3,30492	0,14115	2,67729	0,00000	0,00000	0,00000	0,64469	1,29961
imagem	0,00000	0,04018	0,95981	0,00000	1,73942	0,16423	1,22720	0,10795	0,00000	0,03850	3,23256	0,38398	0,25783	2,06292	0,01381	0,01046	0,63993	1,33521
metálico	0,00000	0,56351	1,27215	0,23839	0,45627	0,10626	2,53839	1,21909	0,08684	0,15774	2,30089	0,11314	1,06513	0,14739	0,04815	0,00000	0,64458	1,12332
reação	0,00000	1,41593	0,39796	0,08203	0,25385	0,77495	0,51094	0,42000	0,08743	0,04634	2,44462	0,11667	4,16359	0,03559	0,00000	0,00000	0,67187	1,45045
útil	0,02372	1,06290	1,56269	0,06798	1,01182	0,05741	1,67484	0,81778	0,07936	0,02496	2,63163	0,02166	1,13627	0,25598	0,01618	0,01465	0,65374	1,02953
transmissão	0,00000	0,07236	0,78968	0,00000	2,59864	0,01437	2,93860	0,20001	0,00000	0,00000	1,51331	0,07920	0,36441	1,30332	0,23963	0,00000	0,63210	1,24570
insumo	0,07954	1,38640	0,82321	0,00000	0,51952	1,84549	0,41629	0,17156	0,13525	0,13705	3,60945	0,08921	1,42750	0,08304	0,01218	0,00000	0,67098	1,30500
formato	0,05421	1,50165	1,25984	0,05119	0,34011	0,10273	1,20373	0,37366	0,11329	0,22066	2,67046	0,21558	0,71649	1,56526	0,07358	0,00000	0,65390	0,96887

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
usinagem	0,00000	0,13848	2,11813	0,03604	0,05611	0,00000	3,86565	2,79441	0,00000	0,23248	0,89324	0,00000	0,02819	0,00000	0,00000	0,00000	0,63517	1,53992
sensorial	0,01965	4,95653	0,07781	0,00000	0,14358	0,37778	0,03433	0,00000	0,07501	0,00000	4,70656	0,11963	1,91858	0,00000	0,00000	0,00000	0,77684	2,08160
local	0,02023	0,47590	1,02859	0,22472	1,31692	0,82262	1,33721	0,30665	0,07618	0,00000	2,41661	0,19813	0,87579	0,99113	0,01863	0,00000	0,63183	0,89568
dimensional	0,00000	0,20215	1,72801	0,06397	0,47779	0,03099	3,37162	2,52106	0,00000	0,01925	1,16673	0,05065	0,24811	0,13426	0,00000	0,00000	0,62591	1,38994
contato	0,00000	0,53145	1,49009	0,05802	1,26034	0,30697	1,88185	0,40000	0,14445	0,02592	1,92402	0,45333	1,16632	0,39573	0,04655	0,00000	0,63031	0,99593
matriz	0,11214	1,06283	1,39438	0,00000	0,67321	0,37001	1,32473	1,50639	0,02277	0,66583	1,38405	0,32007	1,09149	0,26511	0,00000	0,01786	0,63818	0,87844
aplicativo	0,00000	0,04350	0,76205	0,02172	1,23562	0,00000	0,85090	0,00000	0,00000	0,00000	3,30361	0,03379	0,06857	3,48802	0,12450	0,00000	0,62077	1,45067
especial	0,01911	0,62297	1,68223	0,06553	0,77233	0,26088	2,30350	0,85971	0,09562	0,05272	1,58028	0,24953	1,17015	0,26600	0,00000	0,02506	0,62660	0,94562
corrente	0,00000	0,15015	2,13652	0,11977	2,17422	0,05496	2,79218	0,39107	0,00000	0,02407	1,19356	0,00000	0,66036	0,22486	0,01537	0,00000	0,62107	1,25318
pó	0,00000	2,59517	0,33774	0,03748	0,16196	0,69423	0,86507	0,75811	0,06641	0,00000	2,49878	0,26528	2,25113	0,00000	0,00000	0,00000	0,65821	1,35809
interação	0,06930	1,11615	0,44996	0,06791	0,87391	1,11321	1,25995	0,04536	0,08323	0,03520	2,29375	0,11331	1,26849	1,21501	0,05600	0,00000	0,62880	0,91204
polímero	0,00000	0,40428	0,79628	0,01802	0,20799	0,99196	0,50451	0,58840	0,20032	0,13232	2,81251	0,19450	3,07532	0,00000	0,00000	0,00000	0,62040	1,20091
acesso	0,00000	0,15757	0,57568	0,13628	1,52946	0,31685	1,31132	0,16135	0,02212	0,02324	2,43218	0,08402	0,03927	2,91181	0,12110	0,00000	0,61389	1,24735
plástico	0,00000	0,43302	1,23323	0,01810	0,81168	0,15256	3,15880	0,83071	0,00000	0,25017	1,95126	0,13524	0,87528	0,07118	0,00000	0,00000	0,62008	1,15563
gás	0,00000	0,31182	0,74762	0,04190	1,01125	0,23285	2,72028	0,89710	0,28958	0,02407	1,96865	0,00000	1,44900	0,07007	0,00000	0,00000	0,61026	1,11135
circuito	0,00000	0,16386	2,50670	0,01646	2,74267	0,00000	2,70957	0,37803	0,11081	0,00000	0,45857	0,19346	0,07922	0,34534	0,05647	0,00000	0,61007	1,35443
adição	0,00000	2,31520	0,30862	0,09635	0,39504	0,84881	0,58166	1,47830	0,24466	0,05085	1,88016	0,19102	1,57877	0,29164	0,00000	0,00000	0,64132	1,05689
grau	0,10908	0,86072	1,30845	0,07779	0,91706	0,49698	1,53473	0,96677	0,17975	0,04574	1,41024	0,12377	1,13145	0,66061	0,00000	0,00000	0,61395	0,92195
modificação	0,03439	1,21164	0,92881	0,01911	0,36251	0,32778	2,28138	0,89380	0,02497	0,03018	1,45117	0,09841	1,66587	0,50907	0,00000	0,00000	0,61494	0,88057
dimensão	0,00000	0,39199	1,66732	0,21046	0,83251	0,00000	3,39668	0,83532	0,00000	0,26186	1,53968	0,00000	0,25312	0,28098	0,02478	0,00000	0,60592	1,17027
planejamento	0,00000	0,26173	0,52652	0,04004	0,98864	0,42048	1,38076	1,07210	0,13233	0,06161	1,82719	0,09837	0,75654	2,10922	0,03942	0,02207	0,60856	1,15266
seguinte	0,08261	0,96224	0,86895	0,07850	1,00416	0,30404	1,80461	0,34450	0,21495	0,03060	1,82769	0,11113	0,99985	1,49190	0,05557	0,00893	0,63689	1,07852
design	0,00000	0,07916	1,24029	0,00000	0,96118	0,12911	1,92285	0,47402	0,00000	0,07940	3,38253	0,03379	0,19526	0,96952	0,00000	0,00000	0,59169	1,15181
comprovação	0,00000	1,74364	1,04955	0,09191	0,48175	1,40888	1,40543	0,32350	0,12485	0,15260	1,73828	0,02485	0,84957	0,36068	0,03349	0,00000	0,61181	1,12008
comercial	0,10721	0,86074	0,70680	0,00000	1,27254	0,01077	2,22263	0,03597	0,09803	0,02324	1,96865	0,23417	1,26332	1,00066	0,00000	0,02003	0,61405	1,01183
alumínio	0,00000	0,09793	1,80396	0,00000	0,31812	0,22291	2,01776	1,19861	0,11030	0,15029	2,45376	0,10091	0,99830	0,00000	0,00000	0,00000	0,59205	1,16961
dia	0,07765	2,23824	0,60383	0,06453	0,95995	1,62852	0,64030	0,14324	0,00000	0,00000	1,59194	0,00000	1,92362	1,14718	0,00000	0,00000	0,68869	1,27145
levantamento	0,00000	0,86782	1,05274	0,09952	1,14915	0,13562	1,76106	0,34018	0,00654	0,00000	2,24882	0,11382	0,88400	1,45765	0,01537	0,01263	0,63406	1,58672
gerenciamento	0,00000	0,21369	1,42732	0,05829	1,09270	0,03011	1,33875	0,12194	0,06966	0,02106	2,15356	0,01628	0,08680	2,71675	0,12440	0,00000	0,59196	1,15838
potência	0,00000	0,04520	2,52601	0,03194	2,41128	0,05629	2,72126	0,13604	0,03225	0,00000	0,87651	0,03129	0,19504	0,24137	0,05006	0,00000	0,58466	1,26036
inicial	0,02219	1,19315	0,94684	0,10117	0,79289	1,01024	1,63057	0,77731	0,06278	0,00000	1,39596	0,17824	1,16241	0,91654	0,03063	0,00000	0,63881	0,99893
aspecto	0,03620	1,26585	1,12538	0,05250	0,99888	0,33611	1,10737	0,27233	0,10863	0,00000	1,88973	0,32216	1,54812	0,65997	0,01983	0,02247	0,61035	0,90459

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
caixa	0,00000	0,83394	1,67740	0,00000	0,77738	0,00000	2,57862	0,79528	0,02765	0,04998	1,17444	0,16686	0,54628	0,64125	0,04392	0,00000	0,58206	0,97182
hardware	0,00000	0,04350	0,63342	0,04713	3,00266	0,00000	2,07158	0,00000	0,00000	0,00000	1,34113	0,00000	0,18877	1,76728	0,12173	0,00000	0,57608	1,19052
suporte	0,00000	0,16464	0,85269	0,04222	0,97947	0,14808	2,34587	0,55443	0,09175	0,85604	1,27299	0,03017	0,71297	1,20771	0,06528	0,00000	0,58277	0,84983
terceiro	0,15169	0,62015	0,55163	0,00000	0,62401	0,37284	1,53268	0,96152	0,02346	0,02496	2,62251	0,19395	0,56688	1,29631	0,00000	0,00000	0,59641	1,27842
potencial	0,29807	0,99541	0,58186	0,03949	0,84004	0,53042	1,09683	0,36909	0,20024	0,02246	2,02417	0,48905	1,63457	0,38239	0,00000	0,00000	0,59401	0,87159
caracterização	0,11320	0,44209	0,36383	0,04444	1,22884	0,38227	1,37047	0,84313	0,34960	0,01821	2,35888	0,12442	1,55028	0,14920	0,01119	0,00000	0,58438	1,07546
consumidor	0,10560	2,53080	0,24680	0,00000	1,12505	0,33058	0,34777	0,24893	0,00000	0,05800	2,87001	0,04722	1,30193	0,60824	0,00000	0,00000	0,61381	1,08244
seguro	0,00000	0,67452	0,58634	0,12743	0,54425	0,71458	1,78509	0,12277	0,04730	0,03340	2,34646	0,13117	0,35348	1,62179	0,04059	0,00000	0,57057	1,00977
aditivo	0,00000	2,61355	0,27369	0,10241	0,06372	0,30817	0,38367	0,11352	0,77548	0,03476	1,55234	0,15650	3,39665	0,00000	0,00000	0,00990	0,61152	1,19796
coleta	0,10927	0,73878	0,83195	0,05012	1,14448	0,29869	1,55157	0,58517	0,07072	0,00000	2,13391	0,26249	0,40719	1,14336	0,02562	0,00000	0,58458	0,95227
automação	0,04901	1,17634	0,97990	0,02628	0,85166	0,12507	1,92275	0,43722	0,00000	0,00000	1,67448	0,07464	0,18185	1,66299	0,04029	0,00000	0,57515	0,88077
linguagem	0,00000	0,40971	0,42960	0,03563	0,92184	0,00000	0,70755	0,39784	0,02419	0,00000	2,18786	0,00000	0,09965	3,65985	0,08993	0,00000	0,56023	1,29129
solo	0,39257	0,85678	0,28633	0,45937	0,81546	0,00000	2,27632	0,38155	0,11367	0,00000	1,59303	0,64651	1,10685	0,12303	0,00000	0,00000	0,56572	0,89577
analítico	0,00000	0,58974	0,44237	0,00000	0,66569	3,10473	0,87390	0,04536	0,00000	0,04653	2,13350	0,04023	0,74644	0,54137	0,00000	0,00000	0,57687	1,40395
sabor	0,05516	7,45369	0,00000	0,00000	0,00000	0,70006	0,00000	0,00000	0,00000	0,00000	1,08748	0,07586	1,20183	0,00000	0,00000	0,00000	0,66088	2,18353
partícula	0,00000	0,72122	0,56319	0,01752	0,13260	1,72267	1,17807	0,35675	0,33605	0,00000	2,87571	0,10165	1,04910	0,00000	0,00000	0,00000	0,56591	1,25281
relatório	0,05411	0,20617	0,53794	0,00000	0,82690	0,18849	1,72983	0,28677	0,05795	0,02496	1,43400	0,15514	0,57918	2,88628	0,03941	0,00000	0,56294	1,09024
setor	0,04289	0,63688	0,81650	0,00000	1,21004	0,03293	2,47814	0,39277	0,06181	0,02246	1,58818	0,12726	0,59220	0,96762	0,02154	0,00000	0,56195	0,85981
computacional	0,00000	0,07615	0,79186	0,02263	1,92881	0,00000	1,94398	0,58768	0,05017	0,02246	1,98832	0,08192	0,16812	1,14797	0,04285	0,00990	0,55393	1,17184
móvel	0,00000	0,00000	0,51217	0,03663	0,99442	0,00000	1,40455	0,14943	0,17347	0,23933	2,18643	0,00000	0,45410	2,39649	0,25995	0,00000	0,55044	1,07910
fragrância	0,00000	0,06193	0,00000	0,00000	0,00000	0,10484	0,00000	0,00000	0,00000	0,00000	7,97454	0,00000	0,66226	0,00000	0,00000	0,00000	0,55022	2,35388
tecido	0,04046	0,15610	3,07432	0,00000	0,00000	0,31582	0,53234	0,05721	0,14356	0,00000	3,11336	0,12704	1,08775	0,12259	0,00000	0,09428	0,55405	1,28015
fato	0,02372	1,04192	0,86290	0,02089	0,53058	0,37731	1,28428	1,07508	0,04634	0,00000	2,25994	0,09660	0,56886	0,59363	0,05522	0,00000	0,55233	1,17015
sólido	0,00000	1,46209	0,70091	0,01263	0,43968	1,06912	0,57673	0,46943	0,24728	0,03854	1,65720	0,03379	2,32764	0,06886	0,03757	0,00000	0,57134	1,01071
viscosidade	0,01678	2,13134	0,52958	0,01872	0,15811	0,38344	0,27151	0,56683	0,07076	0,00000	2,15020	0,13039	3,12834	0,00000	0,00000	0,00000	0,59725	1,27311
pele	0,01638	0,10390	0,21865	0,00000	0,00000	0,47333	0,03345	0,00000	0,04261	0,00000	5,72575	0,07708	2,01788	0,00000	0,00000	0,00939	0,54490	1,98727
bibliográfico	0,00000	0,96511	1,20609	0,06440	0,56126	0,88015	1,00067	0,77193	0,11448	0,02407	2,09455	0,27060	1,15919	0,32971	0,00000	0,00000	0,59014	1,43181
laboratorial	0,08873	1,70759	0,49490	0,04767	0,58426	0,47576	1,08759	0,74880	0,24622	0,11032	2,52909	0,31670	1,24682	0,31168	0,05575	0,01046	0,62890	1,35906
benefício	0,02862	1,29224	0,31830	0,05240	0,26178	0,87895	0,88672	0,18404	0,16329	0,02496	2,98813	0,06336	1,17954	0,43776	0,00000	0,00000	0,54751	0,96620
resistente	0,27477	0,70156	0,88162	0,05201	0,30952	0,37775	1,94520	0,37514	0,11919	0,19632	2,09321	0,21620	1,07817	0,00000	0,00000	0,02219	0,54018	1,06274
acompanhamento	0,04757	0,61797	1,02301	0,01802	1,01711	0,69519	0,96987	0,97212	0,00000	0,05183	1,60836	0,08337	0,79665	1,11456	0,01808	0,00000	0,56461	1,01923
solda	0,00000	0,16168	1,33380	0,05661	0,88427	0,00000	3,20697	1,09106	0,00000	0,35377	1,10169	0,03379	0,34633	0,02982	0,01576	0,00000	0,53847	1,02876

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
emissão	0,06743	0,17736	0,47294	0,02336	0,80944	0,16203	3,26490	0,16946	0,15686	0,04478	1,94036	0,04446	0,54448	0,73233	0,01863	0,00000	0,53930	0,96693
próprio	0,00000	0,41950	0,88716	0,00000	0,78797	0,28091	1,53720	0,40989	0,05844	0,12459	1,51687	0,15708	0,59977	1,90713	0,07357	0,00000	0,54750	0,85300
documento	0,00000	0,20010	0,76053	0,10683	1,79407	0,03763	0,95615	0,59756	0,00539	0,00000	1,70223	0,00000	0,17001	2,37577	0,00000	0,00000	0,54414	1,05137
complexo	0,03692	0,69028	0,70148	0,00000	0,75783	0,28687	1,53623	0,23968	0,07067	0,08300	2,27399	0,07681	0,79198	1,11997	0,07201	0,00000	0,54611	1,06822
esforço	0,02150	0,32501	1,10005	0,15777	0,41512	0,22079	2,98778	0,48754	0,02580	0,02592	1,52527	0,00000	0,56517	0,71327	0,01593	0,00939	0,53727	0,99803
conexão	0,00000	0,06082	1,05897	0,01597	1,22548	0,00000	2,55669	0,37078	0,00000	0,02246	1,63427	0,00000	0,22743	1,25290	0,12170	0,00000	0,53422	1,04150
tanque	0,02991	0,52730	0,85991	0,03319	0,10783	0,16742	3,69506	0,50733	0,14013	0,02246	1,03912	0,13747	1,15794	0,21773	0,00000	0,00000	0,54018	1,05585
digital	0,02150	0,05746	0,71975	0,00000	1,34088	0,01437	2,26744	0,04347	0,03171	0,00000	1,36950	0,00000	0,15752	2,27500	0,24666	0,01046	0,53473	1,08971
cabo	0,00000	0,03191	1,32388	0,09553	1,95729	0,00000	2,11803	1,23514	0,12037	0,01604	0,78425	0,04722	0,46681	0,12113	0,14486	0,00000	0,52890	1,08282
cartão	0,00000	0,08580	0,00000	0,05380	2,86388	0,00000	0,24076	0,08694	0,00000	0,00000	2,63731	0,00000	0,00000	2,42578	0,03780	0,00000	0,52700	1,30042
precisão	0,02023	0,45608	1,14603	0,01810	1,07321	0,15557	2,32963	0,62227	0,04553	0,08773	1,61360	0,14999	0,34178	0,43763	0,01983	0,00964	0,53293	0,93935
anterior	0,00000	0,52040	0,69405	0,05899	0,69531	0,35264	1,93808	0,64753	0,11165	0,02930	1,50761	0,23708	0,46395	1,19571	0,05113	0,02089	0,53277	0,77840
filtro	0,00000	0,23116	0,86966	0,09834	0,58383	0,24375	2,15887	0,33895	0,00000	0,00000	2,48503	0,33695	0,53233	0,56059	0,00000	0,00000	0,52747	1,03678
doença	0,31881	0,61341	0,00000	0,00000	0,00000	2,15163	0,26757	0,03478	0,00000	0,00000	3,16435	0,47919	1,31416	0,04247	0,00000	0,00000	0,52415	1,45649
negócio	0,03439	0,40977	0,47855	0,00000	0,53728	0,00000	0,30361	0,07745	0,00591	0,00000	2,71020	0,12924	0,09336	3,53505	0,09653	0,00000	0,52571	1,20184
corrosão	0,00000	0,05825	1,03562	0,04861	0,40033	0,00000	2,15728	1,34953	0,19866	0,00000	1,47808	0,09021	1,50030	0,03941	0,00000	0,00000	0,52227	0,97919
tolerância	0,94377	0,26784	0,96653	0,01597	0,35051	0,06410	2,13431	1,73024	0,00000	0,00000	1,19633	0,04828	0,27694	0,30017	0,01576	0,00000	0,51942	1,05415
inclusão	0,00000	1,78242	0,47985	0,01752	0,59048	0,39178	1,22991	0,90848	0,02092	0,03456	2,80838	0,04324	0,09470	0,50990	0,08981	0,00000	0,56262	0,98263
rendimento	0,20182	1,40344	0,86396	0,00000	0,16887	0,67931	0,86599	0,79611	0,16213	0,00000	1,66167	0,17510	1,70085	0,06951	0,00000	0,02073	0,54809	0,85623
aderência	0,00000	0,51253	0,62994	0,01802	0,12908	0,35457	1,16456	0,33445	0,00000	0,00000	1,44112	0,11166	2,75749	1,00574	0,00000	0,02156	0,53005	1,01990
adesivo	0,00000	0,02712	1,20478	0,00000	0,17161	0,21057	0,42489	0,66200	0,00000	0,05664	1,07739	0,34036	4,08624	0,00000	0,00000	0,00000	0,51635	1,28858
paciente	0,00000	0,00000	0,07996	0,00000	0,23500	4,79065	0,00000	0,00000	0,00000	0,00000	2,27047	0,00000	0,30107	0,57952	0,00000	0,00000	0,51604	1,82408
resposta	0,02023	0,56815	0,70476	0,01429	0,40263	1,55447	0,94279	0,26744	0,08873	0,00000	1,49498	0,10181	0,93202	1,29722	0,04845	0,00000	0,52737	0,92809
confiabilidade	0,00000	0,22429	0,89532	0,03130	1,64842	0,28571	2,46125	0,23624	0,02212	0,00000	1,08689	0,10681	0,40572	0,79954	0,01921	0,00000	0,51393	0,90891
segmento	0,02866	0,39756	0,67820	0,15174	0,55942	0,45388	1,29566	0,59408	0,03343	0,49431	1,44887	0,14953	0,90159	1,04401	0,02459	0,00000	0,51597	0,79366
compatível	0,02372	0,28742	0,42712	0,03003	0,64997	0,46508	1,88784	0,09800	0,04749	0,00000	1,66764	0,10556	1,36018	1,12577	0,01183	0,00000	0,51173	0,94935
liberação	0,00000	0,57152	0,97010	0,00000	0,29344	2,25379	1,46634	0,22559	0,02580	0,00000	1,64342	0,09901	0,53599	0,28095	0,00000	0,00000	0,52287	1,15761
regra	0,00000	0,11055	1,40665	0,00000	0,49361	0,03099	0,06144	0,05039	0,00000	0,00000	2,64146	0,00000	0,16323	3,17529	0,04198	0,00000	0,51098	1,20549
convencional	0,04392	0,51559	0,89061	0,14464	0,69607	0,22275	1,33012	0,80230	0,18654	0,09763	1,57840	0,16855	1,19903	0,36969	0,00000	0,00000	0,51537	0,74601
opção	0,00000	0,42085	0,58325	0,06321	0,66107	0,48925	1,11210	0,15866	0,00000	0,07141	2,60039	0,13548	1,01957	1,02196	0,03302	0,00000	0,52314	0,89682
acabamento	0,00000	0,18700	1,54211	0,01971	0,16019	0,00000	1,97752	0,77849	0,02580	0,32619	0,81767	0,21332	2,03351	0,00000	0,00000	0,05148	0,50831	0,98268
versão	0,00000	0,44312	0,56812	0,00000	1,00098	0,00000	1,26777	0,06648	0,00000	0,00000	1,87595	0,00000	0,24327	2,62455	0,12141	0,00000	0,51323	0,98104

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
flexível	0,00000	0,17233	0,98810	0,01810	0,43126	0,48041	0,94812	0,11373	0,02497	0,09715	2,67966	0,03379	0,90118	1,19737	0,02365	0,00000	0,50686	1,07028
alimentação	0,02219	1,32857	1,64128	0,00000	1,02902	0,08891	1,88427	0,81444	0,14567	0,02324	0,83998	0,00000	0,14288	0,27641	0,14967	0,00000	0,52416	0,92150
tubo	0,00000	0,20225	0,64099	0,03826	0,07418	0,00000	2,48286	1,75015	0,02977	0,14624	1,88667	0,02816	0,81877	0,00000	0,00000	0,00000	0,50614	1,03812
escória	0,00000	0,00000	0,22013	0,00000	0,00000	0,00000	0,05732	1,17510	0,04926	0,00000	6,56061	0,00000	0,00000	0,00000	0,00000	0,00000	0,50390	2,59066
extração	0,01965	0,98212	0,46690	0,04111	0,20616	0,08151	0,84450	0,55879	0,10824	0,03503	3,03393	0,31214	0,51263	1,32347	0,03194	0,00000	0,53488	1,07722
similar	0,02457	0,96341	0,94448	0,06137	0,37451	0,55710	1,37081	0,56658	0,09752	0,05912	1,58041	0,18427	1,15480	0,25636	0,00000	0,00000	0,51221	0,74467
eixo	0,00000	0,07031	1,28738	0,03607	0,19654	0,08106	4,03506	0,72046	0,00000	0,05615	1,48991	0,00000	0,00000	0,07239	0,00000	0,00000	0,50283	1,13742
prática	0,00000	0,68716	0,75160	0,16819	0,63106	0,38232	0,39248	0,35403	0,00000	0,00000	2,65495	0,35360	0,26149	1,51674	0,04087	0,01402	0,51303	0,96994
troca	0,02150	0,54134	0,96269	0,00000	0,50572	0,10820	2,43241	0,34351	0,08416	0,00000	1,34556	0,13381	0,45441	1,17577	0,08575	0,00000	0,51218	0,83607
corpo	0,00000	0,50192	1,14100	0,10813	0,52024	0,02702	1,42127	0,61924	0,11659	0,10821	2,35159	0,00000	1,11040	0,07253	0,01883	0,00939	0,50790	0,98311
visual	0,00000	0,72073	0,96145	0,03459	0,54652	0,08189	2,38826	0,38629	0,06711	0,01872	1,33393	0,00000	0,40421	1,26873	0,00000	0,03035	0,51517	1,05968
certificação	0,00000	0,15598	0,74945	0,00000	1,01601	0,17947	3,63049	0,17282	0,00000	0,00000	1,03505	0,03379	0,08437	0,94286	0,03590	0,00000	0,50226	1,30703
ferramental	0,00000	0,05389	1,47893	0,00000	0,12253	0,09580	3,70404	1,40272	0,00000	0,16758	0,74219	0,04693	0,00000	0,19662	0,00000	0,00000	0,50070	1,20655
interferência	0,04525	0,45428	0,84051	0,11965	0,98732	0,29241	2,42113	0,17217	0,00000	0,00000	1,27423	0,02347	0,61136	0,65831	0,08880	0,00000	0,49930	0,86540
ii	0,00000	0,16275	0,12613	0,01752	1,53516	1,10119	1,26725	0,22031	0,04380	0,00000	2,26687	0,00000	0,26903	1,03357	0,01430	0,00000	0,50362	0,97864
contínuo	0,01965	0,84146	1,10663	0,00000	0,68789	0,21856	1,07475	1,43418	0,05110	0,07049	1,11550	0,09641	0,61550	0,76217	0,02196	0,00000	0,50727	0,77451
operador	0,00000	0,17485	1,34452	0,05720	0,62796	0,14159	3,31992	0,76007	0,00000	0,00000	0,67858	0,01816	0,25101	0,61143	0,00000	0,00000	0,49908	1,02544
liga	0,00000	0,00000	0,62206	0,00000	0,76793	0,00000	2,45244	3,10453	0,00000	0,00000	0,70720	0,00000	0,16774	0,06227	0,00000	0,00000	0,49276	1,29814
literatura	0,00000	0,78116	0,59893	0,03668	0,67749	0,84860	0,76518	0,63203	0,16942	0,00000	2,05107	0,14875	0,83740	0,59094	0,06509	0,00000	0,51267	0,90475
umidade	0,06735	2,03457	0,78934	0,09038	0,38767	0,76935	0,80277	0,14045	0,14910	0,09742	1,63694	0,34611	1,25163	0,00000	0,00000	0,00000	0,53519	0,92112
célula	0,00000	0,29095	0,59593	0,15875	0,60322	0,91692	2,58589	0,25454	0,02977	0,00000	1,44022	0,04970	0,81960	0,09244	0,12293	0,00000	0,49755	0,96748
distinto	0,01911	0,72070	0,48661	0,00000	0,68621	0,38650	0,91694	0,15897	0,08536	0,01872	1,60812	0,25494	0,98333	1,61028	0,10298	0,02711	0,50412	0,83250
plano	0,00000	0,20340	0,90226	0,01358	0,95235	0,06798	1,59866	0,86131	0,03991	0,02042	1,41559	0,19017	0,52423	1,09931	0,04994	0,00000	0,49620	0,91054
confecção	0,07455	0,33228	1,22766	0,02969	0,52989	0,08042	1,98146	0,87438	0,02669	0,22087	1,28900	0,02607	1,20231	0,11423	0,02619	0,00990	0,50285	0,97457
calor	0,04986	0,73970	1,26210	0,07691	0,74986	0,36381	2,07285	0,33716	0,00000	0,01821	1,41556	0,04149	0,69107	0,18744	0,03854	0,00000	0,50278	0,85312
rotina	0,06847	0,06619	0,84462	0,00000	0,61936	0,15729	2,05609	0,07167	0,00000	0,00000	1,46550	0,00000	0,13877	2,34553	0,01593	0,00000	0,49059	0,91166
responsável	0,02219	1,16259	0,60509	0,01752	0,71749	0,26036	1,25768	0,14714	0,01935	0,03850	1,46888	0,10824	0,73920	1,24026	0,07347	0,00000	0,49237	0,73931
homologação	0,00000	0,09958	0,27107	0,00000	0,64420	0,05692	1,70734	0,34063	0,00654	0,04813	1,82437	0,04023	0,59006	2,06636	0,14531	0,00000	0,49005	1,20847
densidade	0,02219	1,33108	0,80210	0,05810	0,65212	0,64997	1,11317	0,23449	0,31564	0,08358	1,09286	0,25372	1,55880	0,03152	0,00000	0,00000	0,51246	0,73248
fábrica	0,04299	1,29838	1,28477	0,01358	0,69995	0,03903	2,73622	0,17923	0,00000	0,01925	0,85098	0,29713	0,73850	0,21962	0,00000	0,00000	0,52623	0,89935
período	0,09226	1,62555	0,51545	0,02674	0,60834	1,78681	0,79947	0,11439	0,02497	0,00000	0,97238	0,19677	1,17945	0,64125	0,02196	0,01110	0,53856	0,94825
frequência	0,07974	0,31554	1,46816	0,05336	1,57287	0,11932	2,43736	0,20855	0,03366	0,00000	0,94791	0,02414	0,03436	0,32525	0,13448	0,00000	0,48467	0,93807

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
limite	0,02457	0,51794	0,95271	0,13111	0,80989	0,19722	1,95573	0,75949	0,00000	0,00000	1,29317	0,09751	0,48679	0,53238	0,00000	0,00000	0,48491	0,73003
espessura	0,00000	0,54716	1,36905	0,12536	0,16779	0,03399	2,41319	0,95515	0,08566	0,19088	0,86130	0,13968	0,99027	0,03448	0,00000	0,00893	0,49518	0,82857
papel	0,00000	0,65779	0,68984	0,00000	0,34399	0,10073	0,28461	0,00000	0,00000	0,00000	1,82519	2,23198	1,16232	0,53213	0,01757	0,00000	0,49038	0,92229
essencial	0,01859	1,21068	0,23623	0,02523	0,35115	0,26568	0,32930	0,00000	0,06230	0,00000	4,61094	0,08197	0,26773	0,41040	0,00000	0,00000	0,49189	1,81616
prova	0,00000	0,09604	0,66097	0,09554	0,42631	0,07667	1,62898	0,38873	0,02277	0,00000	1,45401	0,04294	1,09073	1,62175	0,09235	0,02093	0,48242	1,02944
ácido	0,00000	2,29453	0,54048	0,00000	0,09027	0,92035	0,38631	0,15482	0,22337	0,00000	1,63734	0,11689	1,91426	0,00000	0,00000	0,00000	0,51741	1,03994
sinal	0,00000	0,03874	0,79896	0,00000	1,98148	0,19062	1,40480	0,10639	0,02765	0,00000	1,39586	0,04806	0,32891	1,04324	0,28268	0,00000	0,47796	0,83440
armazenamento	0,00000	0,97054	0,56780	0,01468	0,70150	0,35276	1,07709	0,20896	0,00000	0,02246	1,73425	0,04722	0,87585	1,43874	0,03498	0,00000	0,50293	0,76677
frio	0,02646	0,59803	0,14537	0,00000	0,06134	0,03513	1,59536	3,89603	0,22503	0,08022	0,40405	0,15655	0,54203	0,00000	0,00000	0,00000	0,48535	1,40296
transferência	0,01859	0,40304	0,82796	0,03393	0,83938	0,30287	1,00616	0,32946	0,12858	0,00000	1,76009	0,10966	1,05055	0,86086	0,03416	0,00000	0,48158	0,71693
combustível	0,00000	0,09707	0,26723	0,00000	0,25084	0,03586	4,56524	0,43605	0,15608	0,00000	1,11764	0,14868	0,34631	0,16763	0,00000	0,00000	0,47429	1,23974
hora	0,00000	0,66120	1,10474	0,00000	0,24557	0,45245	1,58751	0,41800	0,02867	0,00000	1,36789	0,07943	1,04011	0,70710	0,02365	0,00000	0,48227	0,77301
força	0,00000	0,25846	1,25300	0,09282	0,27501	0,06234	3,05170	0,47594	0,01935	0,01821	1,27293	0,03997	0,54004	0,14233	0,03617	0,00000	0,47114	0,97404
aprovação	0,00000	0,57997	1,21026	0,02593	0,22350	0,93101	1,15238	0,52707	0,00000	0,07966	1,51268	0,08932	0,83660	0,78831	0,00000	0,00990	0,49791	1,00182
falta	0,01810	0,62486	0,99521	0,09253	1,02998	0,28553	1,67092	0,21408	0,03932	0,00000	1,29015	0,17197	0,34209	0,64374	0,00000	0,01144	0,46437	0,89125
código	0,00000	0,03806	0,43983	0,00000	0,97281	0,20001	0,78461	0,11799	0,00000	0,00000	1,93328	0,00000	0,03927	2,72029	0,17896	0,00000	0,46407	1,00119
estável	0,03824	1,01016	0,34438	0,00000	0,19158	1,38307	0,83153	0,23378	0,15903	0,00000	1,06532	0,07836	1,77352	0,43785	0,00000	0,00000	0,47168	0,99261
taxa	0,00000	0,54836	0,46271	0,00000	0,64560	1,11028	0,60960	0,62606	0,22187	0,00000	1,83310	0,12458	0,73351	0,53032	0,05132	0,00000	0,46858	0,80398
pigmento	0,00000	0,09637	0,40079	0,00000	0,00000	0,02702	0,17377	0,00000	0,05160	0,00000	5,41770	0,17592	1,03333	0,00000	0,00000	0,00000	0,46103	2,10980
entrada	0,02751	0,43381	1,24515	0,08805	1,20982	0,16984	1,34580	0,19957	0,03970	0,00000	1,32219	0,09315	0,30671	0,94810	0,05206	0,00990	0,46821	0,74097
estado	0,12544	0,25938	0,77443	0,00000	1,27486	0,38413	0,74824	0,19917	0,06888	0,00000	1,68782	0,08479	0,51258	1,11005	0,11013	0,00000	0,45874	0,71180
questão	0,01965	0,75288	0,50935	0,08432	0,61656	0,19170	1,32428	0,22369	0,05946	0,09894	1,25798	0,08995	1,37843	0,78480	0,01972	0,00000	0,46323	0,77165
enzima	0,00000	3,23105	0,03842	0,00000	0,00000	0,22923	0,02993	0,00000	0,00000	0,00000	1,39491	0,40256	2,62361	0,00000	0,00000	0,00000	0,49686	1,34018
revisão	0,00000	0,24255	0,65401	0,00000	0,93990	0,42022	1,29255	0,69422	0,08450	0,00000	1,79676	0,13619	0,65978	0,60221	0,03860	0,00000	0,47259	0,89163
porta	0,08081	0,07439	0,74390	0,15633	0,52699	0,04811	3,83372	0,11503	0,00000	0,85240	0,22694	0,00000	0,07776	0,51071	0,03626	0,00000	0,45521	1,06077
separação	0,07428	0,51805	0,48787	0,04917	0,10749	0,15304	2,25502	0,27843	0,21915	0,00000	1,74468	0,03129	1,10762	0,37808	0,03740	0,00000	0,46510	0,95897
total	0,02646	1,28286	0,83926	0,07901	0,64907	0,31484	1,32905	0,39088	0,02669	0,00000	1,47981	0,06824	0,57377	0,59838	0,01119	0,00000	0,47934	0,69081
3d	0,00000	0,00000	1,08103	0,04059	0,57698	0,12507	2,46938	0,30350	0,04547	0,00000	1,85178	0,00000	0,66675	0,10287	0,00000	0,01402	0,45484	1,36841
limitação	0,00000	0,88216	0,92297	0,02866	0,59477	0,28040	1,34772	0,63572	0,00000	0,04998	1,55380	0,09267	0,51006	0,93564	0,05846	0,00000	0,49331	0,86099
animal	0,00000	5,23574	0,09240	0,00000	0,03611	1,17323	0,68455	0,04968	0,04691	0,00000	1,10853	0,00000	0,30599	0,00000	0,00000	0,00000	0,54582	1,60293
laminação	0,00000	0,36975	0,32024	0,00000	0,10833	0,00000	0,07360	4,40295	0,05999	0,00000	1,31104	0,05638	0,58413	0,00000	0,00000	0,00000	0,45540	1,51745
cilindro	0,16287	0,04567	0,11629	0,08345	0,00000	0,00000	4,05619	1,36623	0,10321	0,00000	0,82345	0,12548	0,24676	0,00000	0,00000	0,01953	0,44682	1,10748

(Continues...)

Table 58 – ... Continuation

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
canal	0,00000	0,09589	0,68586	0,07049	0,91558	0,13231	0,95690	0,85997	0,05442	0,00000	1,92034	0,00000	0,00000	1,43200	0,09952	0,00000	0,45145	0,78934
forno	0,04586	0,82108	0,89253	0,00000	0,08552	0,00000	0,86988	2,13252	0,21465	0,02106	1,86088	0,05844	0,36253	0,00000	0,00000	0,00000	0,46031	0,90281
mapeamento	0,02372	0,64876	0,64305	0,03271	0,85786	0,05645	1,10695	0,22036	0,06095	0,04492	2,02911	0,07207	0,70460	1,06502	0,07588	0,00000	0,47765	0,82407
movimentação	0,00000	0,21861	1,05338	0,09072	0,44995	0,07415	3,14793	0,39280	0,00000	0,04765	0,96069	0,06746	0,16650	0,49979	0,00000	0,00000	0,44810	0,88991
lógico	0,00000	0,09850	0,69721	0,01468	0,95149	0,01744	1,28668	1,05857	0,05742	0,03789	1,44909	0,00000	0,18544	1,22415	0,11721	0,00000	0,44974	0,76650
posterior	0,04231	0,55509	0,80248	0,04470	0,34548	0,40551	1,66504	1,21199	0,07762	0,09266	1,19049	0,16968	0,56087	0,29468	0,03456	0,00852	0,46886	0,80262
layout	0,00000	0,33833	0,84742	0,03522	0,66164	0,02927	2,74764	0,22150	0,00000	0,33426	0,78677	0,04970	0,22201	0,94248	0,02049	0,00000	0,45230	0,92816
solvente	0,00000	0,11709	0,47422	0,00000	0,00000	1,36216	0,39043	0,35752	0,28966	0,00000	1,28661	0,02414	2,86298	0,00000	0,00000	0,00000	0,44780	1,02516
dinâmico	0,00000	0,13425	0,89655	0,10747	0,59521	0,00000	1,73132	0,08129	0,12073	0,02407	1,56450	0,29464	0,19542	1,42064	0,00000	0,00000	0,44788	0,81533
cenário	0,04927	0,28697	0,26872	0,11915	1,00217	0,18559	0,70688	0,29976	0,04427	0,02246	1,54861	0,15226	0,65118	1,84722	0,09634	0,00000	0,45505	0,83599
mineral	0,00000	2,83232	0,21440	0,00000	0,37847	0,67260	0,26831	0,80064	0,25686	0,00000	1,24601	0,18310	0,92802	0,03343	0,00000	0,00000	0,48839	0,94365
estatístico	0,07605	0,72776	0,40461	0,01207	0,71238	0,33914	0,98294	1,15787	0,02867	0,00000	1,78910	0,02485	0,54216	0,60613	0,02793	0,00000	0,46448	1,02315
bateria	0,00000	0,00000	2,57866	0,00000	1,34591	0,00000	1,08713	0,18867	0,13269	0,00000	1,11929	0,00000	0,03141	0,51584	0,04103	0,00000	0,44004	1,05863
cadeia	0,00000	1,09319	0,22337	0,00000	0,27953	0,15363	0,50756	0,06158	0,00000	0,00000	3,29462	0,00000	1,22096	0,41041	0,00000	0,00000	0,45280	1,34456
gráfico	0,00000	0,04716	0,49865	0,04022	1,16122	0,11554	0,63143	0,06375	0,00000	0,02407	2,14247	0,08448	0,14256	2,07050	0,03513	0,00000	0,44107	0,89280
vegetal	0,14985	2,07967	0,09721	0,00000	0,39775	0,17583	0,07623	0,15230	0,06557	0,00000	3,05458	0,13909	1,02460	0,00000	0,00000	0,00000	0,46329	1,33589
emulsão	0,00000	0,29259	0,05859	0,00000	0,00000	0,18346	0,08714	0,41773	0,14519	0,04669	3,54929	0,00000	2,24511	0,00000	0,00000	0,00000	0,43911	1,21013
compressão	0,00000	0,29926	0,54985	0,14467	0,40891	1,64060	0,67650	0,03068	0,23466	0,04228	0,82003	0,10237	1,62110	0,50443	0,05058	0,00000	0,44537	1,06169
ph	0,02293	1,79442	0,30320	0,00000	0,46234	0,84243	0,12192	0,45342	0,07473	0,00000	1,81771	0,27873	1,62830	0,09422	0,00000	0,00000	0,49340	0,91172
complexidade	0,00000	0,16959	0,49558	0,05722	0,73924	0,15624	0,88404	0,13668	0,12459	0,00000	2,01098	0,06587	0,44864	1,65711	0,02855	0,00000	0,43590	1,03151
filme	0,00000	0,59545	0,41453	0,00000	0,16870	0,24986	0,32011	0,25916	0,14713	0,06618	2,34740	0,33058	2,18476	0,00000	0,00000	0,00000	0,44274	0,94927
sintético	0,01720	0,21945	0,24919	0,01697	0,31618	0,35751	0,22522	0,14398	0,11981	0,00000	4,77980	0,04828	0,45690	0,00000	0,00000	0,01308	0,43522	2,06404
dosagem	0,04586	1,99719	0,33303	0,01429	0,06655	0,81810	1,10735	0,16993	0,13548	0,00000	1,14773	0,28979	1,36031	0,11876	0,00000	0,00000	0,47527	0,78016
padronização	0,00000	0,94595	1,36930	0,04004	0,53144	0,11292	0,85661	0,34004	0,03686	0,01821	1,37664	0,11929	0,28703	1,06181	0,04931	0,00000	0,44659	0,74325
parcela	0,14948	0,05023	0,03183	0,00000	0,03186	0,00000	0,08200	0,00000	0,00000	0,00000	6,34624	0,00000	0,02893	0,13002	0,00000	0,00000	0,42816	3,68620
gase	0,16923	0,36780	0,71090	0,02089	1,02025	0,02289	1,82255	0,51448	0,19116	0,00000	1,05697	0,00000	1,14257	0,00000	0,00000	0,00000	0,43998	0,73225
conteúdo	0,00000	0,38386	0,20282	0,00000	0,57873	0,40699	0,45860	0,11375	0,02092	0,02407	2,56505	0,00000	0,15147	1,96248	0,20202	0,00000	0,44192	0,96919
pintura	0,00000	0,00000	1,39778	0,01502	0,07342	0,00000	2,62682	0,55527	0,00000	0,19912	0,26040	0,00000	1,79215	0,00000	0,00000	0,00000	0,43250	0,94899
vibração	0,00000	0,00000	1,60417	0,18582	0,79965	0,00000	3,47171	0,06429	0,02419	0,00000	0,56568	0,00000	0,14077	0,00000	0,05281	0,00000	0,43182	1,13480
superficial	0,00000	0,17120	0,64724	0,00000	0,17787	0,17568	1,03183	2,36221	0,09833	0,00000	0,88258	0,44130	0,92065	0,00000	0,00000	0,00000	0,43181	0,88933
espaço	0,01965	0,52415	0,97339	0,03850	0,88244	0,00000	2,06158	0,06527	0,03366	0,16219	1,37746	0,20981	0,18162	0,46910	0,00000	0,00000	0,43743	0,75775
válvula	0,02751	0,17797	0,72367	0,00000	0,05035	0,05546	3,13446	0,73549	0,05557	0,00000	1,29647	0,07240	0,62455	0,00000	0,00000	0,00000	0,43462	1,08017

(Continues...)

Table 58 – Conclusion

Entities	Agroindustry						Food						Consumer good					
	AGR-BAR-AP	AGR-BAR-RP	AGR-ELE-AP	AGR-ELE-RP	AGR-MET-AP	AGR-MET-RP	FOD-BAR-AP	FOD-BAR-RP	FOD-ELE-AP	FOD-ELE-RP	FOD-MET-AP	FOD-MET-RP	CSG-BAR-AP	CSG-BAR-RP	CSG-ELE-AP	CSG-ELE-RP	CSG-MET-AP	CSG-MET-RP
fácil	0,00000	0,59656	0,58614	0,00000	0,68358	0,34387	1,11967	0,23681	0,04300	0,00000	1,63008	0,17468	0,76015	0,67735	0,00000	0,00000	0,42824	0,77645

Table 58 – Source: Produced by the author in August, 2022

Apendix F – Step 2 - Semantic sets comparision

Table 59 – Results of Comparision on the semantic set [método (method), metodologia (methodology)

Entity	Método (method)	Metodologia (methodology)
abordagem	x	
água	x	
alterar	x	
analisar		x
análise	x	
animal	x	
através	x	
automação		x
bete		x
carta		x
ciclo	**x**	**x**
consistir		x
corrente	x	
definição		x
desafio	x	
desenvolver	x	
desenvolvimento	**x**	**x**
dividir	x	
eficiência		x
eficiente	x	
eletromagnético	x	
estudo		x
experimental	x	
físico	x	
fornada		x
graduação	x	
graf	x	
ideia		x

(Continues...)

Table 59 – Conclusion

Entity	Método (method)	Metodologia (methodology)
identificação		x
implementação		x
insumo	x	
laboratório		x
lançamento		x
lote	x	
manejo	x	
manutenção		x
medição	x	
metodologia		x
parte		x
planejamento	x	
prefixação		x
processo	x	
radial	x	
rastreabilidade		x
rota		x
sempre	x	
ser		x
setor	x	
solução		x
submerso	x	
superfície		x
técnico		x
tecnológico	x	
teórico		x
teste	x	
universidade	x	
utilização	x	

Table 59 – Source: Produced by the author in August, 2022

Table 60 – Results on Comparision on the semantic set [Fabricação (manufacturing), Produção (production)

Entity	Fabricação (manufacturing)	Produção (production)
abatimento	x	
acabamento		x
aplicação	x	
automático	x	
banco		x
barra	x	
basáltico		x
bloco	x	
calço	x	
cancã		x
carbonatar		x
conjunto	x	
conta	x	
decisão		x
desenho	**x**	**x**
encômio	x	
ensaio		x
estar	x	
eu	x	
executar	x	
fabricação	x	
ferramenta		x
físico		x
focar		x
folha		x
forma		x
grande	x	
inversor		x
linha	x	
macho		x
mais	x	
nível		x
operação	x	
pão	x	

(Continues...)

Table 60 – Conclusion

Entity	Fabricação (manufacturing)	Produção (production)
pé		x
pequeno	x	
pesquisa	x	
procedimento		x
produção		x
project		x
protótipo	x	
qualidade	x	
realização		x
realizar		x
ser	x	
setor	x	
solar		x
sulafricano		x
tecnológico		x
teste		x
tolerância	x	
utilizar		x
veterinário		x

Table 60 – Source: Produced by the author in August, 2022

Table 61 – Results on Comparision on the semantic set [Necessário (necessary), Need (necessidade)

Entity	Necessário (necessary)	Need (necessidade)
adaptação	x	
adequar		x
amido	x	
antes		x
areia		x
atender		x
atingir	x	
automaticamente	x	
binário	x	
cliente		x

(Continues...)

Table 61 – ... Continuation

Entity	Necessário (necessary)	Need (necessidade)
concepção	x	
constituir		x
dado		x
desenvolver		x
domínio		x
equipe	x	
estar	x	
experimental	x	
fórmula		x
formulação		x
gás	x	
genético	x	
gramatura	x	
impressão		x
lactose	x	
linha	x	
melhoria		x
método	x	
necessidade		x
novo	x	
otimizar	x	
parametrização		x
passar		x
pneumático	x	
possível	x	
principal		x
processo		x
produção	**x**	**x**
programa	x	
projeto	**x**	**x**
qualidade	x	
quê	x	
realizar	x	
saúde		x
segurança	x	

(Continues...)

Table 61 – Conclusion

Entity	Necessário (necessary)	Need (necessidade)
ser	**x**	**x**
sistema		x
solução		x
ter		x
termosselantar	x	
teste	**x**	**x**
uso	x	
utilizar	x	

Table 61 – Source: Produced by the author in August, 2022

Table 62 – Results on Comparision on the semantic set [Produtivo (productive), Produtividade (productivity)

Entity	Produtivo (productive)	Produtividade (productivity)
algoritmo		x
alto		x
análise	**x**	**x**
aplicação	x	
asnico		x
através		x
aumento	x	
avaliar		x
comparar		x
confluência	x	
conhecimento	x	
cor		x
deformação	x	
demarcar		x
desenvolver		**x**
desenvolvimento	**x**	
elemento		x
equipamento	x	
especializar		x
especialmente	x	
exigir	x	

(Continues...)

Table 62 – ... Continuation

Entity	Produtivo (productive)	Produtividade (productivity)
fadigo	x	
fruto		x
ganho		x
geométrico		x
interior		x
laboratório		x
lançar	x	
lote	x	
mar	x	
método	x	
milho	x	
mole		x
necessário	x	
novo	x	x
novo	x	x
poder		x
polimerização		x
ponto		x
produtividade	**x**	**x**
produtivo	x	
prova		x
raiz		x
realizar	**x**	**x**
região	x	
semente		x
simulação		x
sistema	x	
somente	x	
tecnologia	x	
tecnológico		x
testar		**x**
teste	**x**	
trabalho	x	
variedade		x
verificar	x	

(Continues...)

Table 62 – Conclusion

Entity	Produtivo (productive)	Produtividade (productivity)

Table 62 – Source: Produced by the author in August, 2022

Table 63 – Results on Comparision on the semantic set [End (Fim), Final (final), Result (resultado)

Entity	End (Fim)	Final (final)	Result (resultado))
acrítico			x
ajuste			x
além	x		
amsterdam		x	
analise		**x**	**x**
aplicação		x	
atividade	x		
automático	x		
avaliação	**x**		**x**
bibliográfico	x		
bloco	x		
cálculo	x		
carbonização			x
carretel			x
clone			x
concreto		x	
condição		x	
configuração		x	
controlar		x	
coxa	x		
criação		x	
definição			x
definir	x		
demão		x	
desafio			x
desenvolvimento			x
empate	x		
ensaio			x
entidade	x		

(Continues...)

Table 63 – ... Continuation

Entity	End (Fim)	Final (final)	Result (resultado))
estabilidade		x	
fim	x		
flexível		x	
formulação	x		
identificar			x
imobiliário			x
implantar		x	
indicador	x		
influência			x
informação	x		
integração			x
limitante			x
literatura		x	
mapeamento			x
melhor		x	
metodologia		x	
montagem		x	
não		x	
necessário	x		
observar	x		
país			x
permeabilidade	x		
piloto		x	
poço			x
posição	x		
preparação	x		
processo		x	
produto		**x**	**x**
propriedade		x	
protótipo			x
químico	x		
raspagem		x	
realizar		x	
redução		x	
referir	x		

(Continues...)

Table 63 – Conclusion

Entity	End (Fim)	Final (final)	Result (resultado))
requerer			x
resultado			x
retornar			x
revalidar		x	
século		x	
ser	**x**	**x**	
simulação			x
sistema	x		
suprimento		x	
técnico		x	
temperatura			x
térmico			x
teste			x
tratamento	x		
vazão	x		

Table 63 – Source: Produced by the author in August, 2022

Apendix G – Step 3 - Domain distinction

Table 64 – 2015 recognized entities

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
produto	0,34184	10,24679	8,14527	0,44989	6,23895	7,00296	18,32348	6,13287	0,96621	7,53080	19,88180	1,78877	10,71777	5,21807	0,16186	0,03797	6,44908	7,75996
novo	0,59388	8,84481	8,08089	0,51486	6,12484	3,96149	21,40812	5,39427	0,86203	2,20597	19,71628	1,52453	9,17346	8,07478	0,48140	0,05905	6,06379	7,78034
processo	0,34960	8,44510	9,30615	0,45670	4,99695	5,76632	16,60599	6,87767	1,00866	0,61658	17,34182	1,58766	8,89643	7,40267	0,37218	0,07903	5,63184	6,93674
desenvolvimento	0,45379	7,64748	7,80815	0,49250	6,52649	5,48402	17,27029	5,34976	0,72716	1,13845	17,18458	1,52526	8,27004	7,70495	0,43417	0,04443	5,50384	6,73888
projeto	0,21635	6,65927	8,19005	0,59903	6,35712	3,70845	18,34480	6,33585	0,93452	1,69403	14,49100	1,18160	6,62444	7,29613	0,36869	0,03847	5,18999	6,51819
sistema	0,23601	3,76440	7,58568	0,55863	6,99969	1,24343	17,50002	3,43819	0,65947	0,47941	15,79372	0,86384	5,34629	8,63933	0,37660	0,03907	4,59524	6,42679
anexo	0,06604	4,59360	2,04789	0,00000	1,43280	0,26053	25,05176	0,15270	0,27465	0,15987	7,45546	0,00000	0,55771	18,50409	0,00000	0,00000	3,78482	9,23035
teste	0,32568	5,29704	5,42272	0,35221	4,46750	2,86433	11,30995	3,90311	0,57374	0,47919	12,59983	1,02011	6,43298	4,68618	0,27606	0,00990	3,75128	5,59869
aplicação	0,25902	4,27495	5,14841	0,37837	3,56395	1,29498	10,17503	3,13383	0,41303	0,14874	11,02968	1,09412	7,86189	5,72760	0,19163	0,01308	3,41927	4,44995
grande	0,38907	4,78526	5,11049	0,42604	3,84318	2,62744	10,30808	3,25128	0,45738	0,76681	10,48403	1,10914	5,16748	4,99467	0,18403	0,03102	3,37096	4,68818
estudo	0,39271	6,01164	4,20955	0,44577	3,91971	3,83931	10,21468	3,57384	0,59745	0,88053	10,91361	1,02591	4,59914	3,12394	0,25803	0,01953	3,37659	4,65089
tecnologia	0,25771	3,74093	3,89808	0,16310	4,82887	1,88245	8,39811	1,40691	0,53449	0,79236	12,05078	0,94842	5,50632	6,25534	0,27992	0,00000	3,18399	4,42946
material	0,38590	2,57096	5,49995	0,44127	2,95885	1,01469	11,11879	4,74128	0,65811	0,90402	11,24140	0,95139	5,03368	0,60292	0,03422	0,01465	3,01076	4,35713
pesquisa	0,35672	5,63971	4,59292	0,21923	3,26644	2,57778	7,81261	3,46385	0,62856	0,47858	11,12813	0,99414	4,39393	4,12913	0,24094	0,03345	3,12226	4,47453
produção	0,35402	5,61829	3,89115	0,18131	2,40148	2,80165	8,50527	3,71808	0,57352	0,38667	10,57527	1,34063	6,27161	2,00545	0,09098	0,02414	3,04622	3,82485
alto	0,43911	3,63257	4,04944	0,30434	3,30661	2,91676	7,48035	2,60766	0,69001	0,33107	8,16128	0,84054	5,77194	3,43279	0,20528	0,03532	2,76282	3,70696
equipamento	0,12806	5,09665	4,83820	0,42257	4,10385	1,65963	13,00827	2,22617	0,34160	0,37609	7,45458	0,48662	3,01943	1,73064	0,30763	0,06144	2,82884	4,02494
necessário	0,21776	3,53043	4,29246	0,39381	2,81747	3,09755	8,13375	2,96689	0,44909	0,25870	9,41858	0,73512	3,55635	4,15767	0,14476	0,00000	2,76065	3,79259
mercado	0,15625	4,41009	3,87946	0,14094	2,94974	1,90895	8,93793	2,96280	0,39032	0,44507	7,07772	0,89206	5,90288	3,62589	0,22362	0,00939	2,74457	3,55539
forma	0,14679	3,39525	4,31776	0,24404	3,41887	2,87262	7,55647	1,83140	0,37659	0,26407	8,57574	0,91678	3,89110	5,12468	0,21857	0,01110	2,69761	3,61438
solução	0,06445	2,47506	3,68448	0,28677	3,99383	1,62717	7,54094	1,77595	0,31381	0,17991	9,64080	0,37193	3,52642	6,90270	0,40009	0,01263	2,67481	3,85715
análise	0,27108	4,67981	3,54965	0,23386	3,61021	2,06862	9,20221	2,18953	0,50100	0,16149	9,04549	0,83252	3,26255	4,29387	0,14699	0,00964	2,75366	4,47116
qualidade	0,24934	5,05890	3,71174	0,31107	3,12065	1,74018	7,14752	2,79772	0,37922	0,35678	8,62074	1,34261	3,78996	2,54176	0,16656	0,03219	2,58543	3,22537
bom	0,43458	4,32589	3,14123	0,21394	2,60058	1,81340	6,26906	2,19025	0,46033	0,26830	8,74437	1,03771	6,20366	3,39317	0,11950	0,02624	2,57764	3,24585
linha	0,08889	5,10728	4,20156	0,10487	3,47341	1,91533	9,53459	2,40661	0,11211	1,01360	7,79150	0,58871	3,49865	1,17139	0,11977	0,03575	2,57275	3,72736
técnico	0,16171	3,54429	4,31646	0,39784	3,62054	1,92335	8,73559	2,33208	0,37485	0,27692	7,90812	0,59182	3,39591	3,24542	0,25438	0,02705	2,56915	3,68049
utilização	0,13608	3,63837	3,42972	0,34495	2,35759	1,51215	6,23250	2,15399	0,36383	0,23910	11,57443	0,83539	3,54269	3,97852	0,14202	0,00000	2,53008	3,62168
dado	0,04534	1,91766	2,97821	0,20349	4,49548	0,92911	4,89312	2,10648	0,28842	0,04806	10,52774	0,42937	1,60105	8,70409	0,43954	0,00000	2,47545	4,04196
tecnológico	0,16062	4,56468	2,76811	0,10273	2,75708	1,84853	6,49763	2,48937	0,58527	0,27513	7,44030	0,66811	4,15840	4,22079	0,18402	0,03709	2,42237	3,59252
informação	0,06879	4,39462	2,74783	0,10058	2,78506	0,59759	6,63776	0,38502	0,12871	0,16981	11,16515	0,14812	1,41442	7,80528	0,28647	0,01144	2,42791	4,31447
metodologia	0,17280	2,94114	2,90254	0,49420	4,30894	2,54439	5,86261	1,46241	0,46590	0,46060	8,59293	0,66019	3,92836	4,11325	0,20650	0,00000	2,44480	3,98747
controle	0,40830	2,90878	3,66387	0,20723	3,76622	1,69822	7,20777	1,86549	0,34856	0,08053	6,35718	0,88365	3,35644	3,98783	0,17834	0,03322	2,30948	2,98681
tempo	0,02751	3,48711	4,03343	0,18193	2,57262	1,25666	7,50571	2,08023	0,33821	0,21255	6,32613	0,44928	4,38532	4,12951	0,15347	0,01110	2,32192	3,09552
ferramenta	0,00000	0,51639	3,76777	0,20351	2,44173	0,08993	9,21676	7,03961	0,24909	0,85586	5,11584	0,30158	0,83349	5,49611	0,13675	0,00000	2,26653	3,95448
característica	0,37883	5,64372	3,77673	0,27891	2,34235	1,79530	5,37901	1,85906	0,43123	0,20786	7,64686	0,86334	4,95243	1,53478	0,02928	0,02474	2,32153	3,14351

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
necessidade	0,04231	2,71532	3,29894	0,18637	3,32362	1,00259	6,37182	2,20973	0,19428	0,33408	7,40234	0,66627	3,71274	4,32292	0,17552	0,01915	2,24863	3,03958
formulação	0,18388	6,47269	1,04781	0,03509	0,58818	5,00172	0,82675	0,57147	0,13400	0,00000	12,56167	0,81419	8,76144	0,14591	0,00000	0,00852	2,32208	4,85310
base	0,09049	3,76090	2,76853	0,30183	2,19633	1,43873	4,42224	1,50631	0,43942	0,50234	8,59758	0,59580	4,56032	3,99820	0,14359	0,04312	2,21036	3,03723
principal	0,22610	3,24305	3,24766	0,21828	2,97959	1,80825	6,42482	1,86737	0,58543	0,11009	6,45491	0,45290	3,52629	3,42686	0,10404	0,00000	2,16723	3,30699
empresa	0,03439	4,25633	3,13567	0,31713	2,03666	1,73413	6,76784	2,18012	0,29474	0,64773	6,09661	0,51283	3,21486	3,73494	0,21605	0,03009	2,20063	2,81761
custo	0,13765	3,80173	3,77045	0,11655	3,29828	0,69473	6,77938	1,96001	0,22492	0,46822	6,34511	0,69502	4,05608	2,16646	0,14503	0,00000	2,16623	2,86724
componente	0,04545	1,37204	5,58890	0,24099	3,02546	1,14103	10,30434	1,44492	0,23470	0,23174	4,30336	0,15617	2,21468	3,08926	0,13908	0,00000	2,09576	3,30435
avaliação	0,39416	3,19860	3,08339	0,14397	2,47226	2,70278	5,61793	1,95883	0,26157	0,07543	8,84024	0,71260	4,24597	1,49865	0,18259	0,01948	2,21303	3,81174
elemento	0,11877	2,04752	3,80082	0,20166	2,45779	1,43080	8,12307	2,52712	0,34515	0,22827	5,91791	0,65877	2,84124	2,37013	0,13558	0,01371	2,07614	3,59549
desafio	0,14553	3,26579	2,81941	0,20724	2,23329	1,91150	5,88838	1,92700	0,26909	0,30432	6,69857	0,48627	3,27995	3,58665	0,15518	0,01900	2,07482	4,34732
uso	0,12240	2,89807	3,03330	0,07739	3,10817	1,93308	4,32960	1,57248	0,32002	0,39980	6,97336	0,67262	3,98761	3,50237	0,12953	0,02544	2,06783	2,70351
tipo	0,11801	4,31677	3,62960	0,22485	2,02845	1,40999	5,64239	1,79681	0,55608	0,58447	6,86461	0,73180	3,73771	1,74441	0,13055	0,01898	2,09597	2,70602
redução	0,29456	3,38394	3,67641	0,15933	1,98076	1,29292	7,15672	2,37328	0,23776	0,15211	6,88673	0,68034	3,35710	1,24567	0,00000	0,00000	2,05485	3,06704
cliente	0,00000	3,33161	2,03520	0,07664	1,80955	0,05018	4,48710	2,31185	0,20957	0,38324	7,28500	0,53659	5,18962	5,70950	0,28352	0,00000	2,10620	2,95142
método	0,15066	2,80396	4,01641	0,15673	3,37591	1,86129	4,54382	1,16294	0,41757	0,13658	7,59979	0,51434	3,38544	2,83790	0,13411	0,00000	2,06859	3,01927
resistência	0,48115	2,01353	3,28865	0,41619	1,03347	0,23405	6,33210	2,86827	0,54117	0,37330	5,75785	1,06894	7,21281	0,19874	0,08305	0,01342	1,99479	2,99376
nome	0,00000	0,12553	8,41748	0,00000	8,29465	0,04053	4,39640	1,04326	0,00000	0,00000	1,35605	4,12196	3,54063	0,06149	0,00000	0,00000	1,96237	6,06080
software	0,00000	0,94004	3,32066	0,15571	4,93470	0,00000	7,75295	2,29480	0,00000	0,16467	5,56780	0,03249	0,84792	5,18404	0,16197	0,03216	1,96187	3,28504
fabricação	0,00000	2,84582	5,13757	0,12154	1,95900	3,71873	7,07832	3,66908	0,15326	0,52756	3,53403	0,32860	1,83647	0,27318	0,01757	0,02200	1,95142	2,87316
modelo	0,01720	1,05230	3,28121	0,06795	3,50058	0,26036	7,48334	0,51772	0,50202	0,25954	6,84339	0,95229	1,21350	4,56880	0,13101	0,00000	1,91570	3,00929
resultado	0,29946	2,93522	2,97033	0,17544	2,19141	1,58431	4,38740	2,72704	0,29428	0,14645	6,68048	0,63357	3,66555	2,43799	0,08086	0,02054	1,95189	2,75622
melhoria	0,15613	2,74869	3,37931	0,08747	2,34746	0,77682	6,40345	1,59019	0,22770	0,81631	5,62358	0,69655	2,63756	2,47451	0,10682	0,01854	1,88069	2,39464
desempenho	0,16372	4,04347	2,29613	0,10913	2,58304	1,01133	4,98804	1,81303	0,20503	0,06470	4,76164	0,27521	4,29188	3,03746	0,16438	0,03706	1,86533	2,45886
possível	0,15042	2,91133	2,94642	0,25741	1,88313	0,80051	5,50037	1,60946	0,26661	0,26774	6,00162	0,49669	2,77547	3,06275	0,17784	0,01144	1,81995	2,43726
etapa	0,20669	2,82694	2,67161	0,16664	2,29184	3,44740	4,08366	2,61493	0,24190	0,11116	5,23613	0,38245	2,62349	2,21125	0,10285	0,03303	1,82825	2,62695
baixo	0,10658	1,83910	2,31329	0,17800	2,37989	1,54312	4,34300	1,26516	0,57764	0,12057	6,60808	0,69116	3,91240	2,01739	0,13948	0,01998	1,75343	2,70475
final	0,10744	4,52134	3,51241	0,17790	1,38659	0,91145	3,95512	1,96650	0,24733	0,12848	5,06864	0,66146	4,34392	1,87040	0,10459	0,02067	1,81152	2,40675
segurança	0,12686	1,20303	2,21065	0,22536	2,17656	3,51185	5,20470	1,05058	0,02277	0,05889	5,91110	0,35412	1,43970	4,15145	0,16513	0,00964	1,73890	2,53517
temperatura	0,12208	3,48974	3,85208	0,10993	2,15210	0,62085	4,89614	2,41493	0,51038	0,12690	5,21519	0,43720	3,88108	0,39961	0,07725	0,02108	1,77041	2,49294
condição	0,43177	2,15069	2,06699	0,45125	2,09510	0,83116	4,74667	1,40622	0,46352	0,02592	6,84771	0,83397	4,59688	0,84828	0,01479	0,00964	1,73878	2,40899
conhecimento	0,13682	3,44479	2,53434	0,24123	1,91231	0,50147	5,00325	2,27600	0,25329	0,16475	5,92481	0,52278	2,47399	2,43394	0,07302	0,00000	1,74355	2,35913
risco	0,09156	1,61591	2,84541	0,17529	1,80002	1,29568	4,70525	2,07399	0,26719	0,16320	6,41846	0,52127	2,52137	2,60235	0,02196	0,02702	1,69662	2,80538
conceito	0,01859	1,36590	3,19578	0,11833	1,95546	0,25646	8,33492	1,18519	0,12752	0,04813	5,56850	0,33629	1,10051	3,63356	0,23294	0,00000	1,71738	2,88604
estrutura	0,03827	2,05040	2,74746	0,46909	1,87092	0,51531	5,06499	1,07122	0,19287	0,25441	7,21608	0,68563	2,42592	2,67417	0,15784	0,05332	1,71799	2,58119
operação	0,09429	1,40928	2,81773	0,20588	3,04147	0,30260	6,93469	2,07526	0,24828	0,05336	4,54651	0,26266	1,93774	3,19820	0,12824	0,00000	1,70351	2,50379
mecânico	0,00000	0,62512	4,17248	0,19402	2,25087	0,15774	7,78264	3,82584	0,24910	0,18954	4,32030	0,37221	2,50865	0,09795	0,08028	0,02652	1,67833	2,96465

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
pequeno	0,23447	2,48858	2,75708	0,16171	1,55060	1,15089	5,09394	2,01384	0,24316	0,56949	5,99636	0,49907	2,74533	1,50921	0,06940	0,01144	1,69341	2,45719
requisito	0,04744	0,92940	2,19383	0,11325	2,09219	0,29997	7,86676	1,65208	0,06072	0,11594	4,35554	0,16442	3,14700	3,44251	0,12536	0,00000	1,66290	2,84287
inovador	0,11582	2,07357	2,75250	0,09301	2,13748	1,73319	5,09559	1,33458	0,27682	0,22244	5,04877	0,52281	2,81533	2,04563	0,04314	0,01254	1,64520	3,37888
peça	0,00000	0,41114	3,55323	0,22996	0,78306	0,02927	11,32331	3,06068	0,03870	0,92560	4,04921	0,28953	1,35747	0,27698	0,00000	0,01077	1,64618	3,19295
realização	0,21867	2,48614	2,48999	0,31322	1,96368	1,40011	4,77848	1,61434	0,27170	0,12236	5,96035	0,46314	3,45154	1,86528	0,13761	0,00990	1,72166	3,01857
eficiência	0,17485	2,54348	2,63144	0,18756	1,84021	0,71066	5,62462	1,02260	0,30448	0,18250	5,24991	0,49230	3,09673	1,58461	0,03761	0,02226	1,60661	2,15072
máquina	0,07268	1,31238	4,36525	0,23984	1,01018	0,12428	9,24447	2,53023	0,05990	0,14240	3,41280	0,97881	0,92077	1,27305	0,00000	0,00000	1,60544	2,83673
protótipo	0,06191	2,17374	4,17382	0,10894	2,67957	0,21728	5,75530	1,48323	0,15326	0,11886	5,17140	0,30702	2,36979	1,95066	0,11564	0,02409	1,67903	3,45396
tratamento	0,23104	2,29810	1,30169	0,17537	0,98846	4,29877	3,06832	2,06258	0,32823	0,12007	5,84404	0,41517	2,87323	1,84219	0,11452	0,00000	1,62261	2,52917
ensaio	0,39651	1,23590	4,48785	0,37652	2,58324	0,69430	4,91744	2,35802	0,26183	0,17392	4,41258	0,12142	3,37464	0,09316	0,07877	0,00000	1,59788	3,03748
descritivo	0,00000	0,08328	0,11533	0,00000	0,02321	0,00000	21,79991	0,04013	0,00000	0,00000	0,19980	0,02640	0,77232	1,58497	0,00000	0,00000	1,54033	7,55341
objetivo	0,12165	3,53424	2,30697	0,09182	1,73160	1,04759	4,10829	1,69422	0,36763	0,14102	4,78673	0,46134	2,49608	2,21432	0,12827	0,00000	1,57699	2,15469
água	0,19068	3,78258	2,18253	0,40674	1,67023	1,20488	2,79878	0,80027	0,47050	0,10680	6,80800	0,81270	4,07558	0,04413	0,08212	0,00000	1,58978	2,58270
performance	0,03762	2,76164	1,20238	0,10944	0,86496	0,31133	4,30254	0,72623	0,20085	0,04279	6,61292	0,36183	3,98016	3,31406	0,21227	0,01665	1,56610	2,19898
barreira	0,08074	2,42742	2,42360	0,06876	1,59672	1,31795	4,23438	1,25740	0,32599	0,14535	4,97252	0,58269	2,67859	2,19362	0,08912	0,00000	1,52468	3,31789
produtividade	0,33908	1,82780	2,55270	0,25522	0,88874	0,66915	5,48406	1,38290	0,27207	0,18861	4,60715	0,60901	3,61670	1,50617	0,01657	0,03280	1,51555	2,43411
elétrico	0,00000	0,49323	5,32288	0,18648	4,60614	0,00000	8,47286	0,81161	0,06530	0,05561	2,64296	0,14085	0,50208	0,59098	0,09310	0,00000	1,49901	2,94653
definição	0,07272	1,63533	2,12115	0,20943	2,15053	0,78679	5,15553	1,11041	0,24080	0,10157	5,72154	0,30476	2,19733	2,87340	0,19234	0,04095	1,55716	2,67157
estabilidade	0,22568	3,14355	1,23657	0,12631	0,67739	4,08517	2,05425	0,49987	0,23706	0,04171	5,57771	0,15274	5,80659	0,55860	0,03456	0,00000	1,52861	2,59147
campo	0,24994	2,91551	1,96839	0,21752	2,28602	0,06055	8,19081	0,78622	0,33928	0,08879	3,77053	0,48791	2,65297	1,42060	0,22432	0,00000	1,60371	3,44888
específico	0,05781	2,46718	2,14617	0,23090	1,42903	1,38978	3,56929	1,45988	0,21566	0,25053	5,54028	0,48256	2,20823	2,36774	0,05196	0,01465	1,49260	2,05722
área	0,12658	1,43737	2,07201	0,07797	2,46511	0,88023	4,56725	1,72202	0,23546	0,14542	5,09619	0,70148	1,99022	2,06392	0,05039	0,00852	1,47751	1,93373
validação	0,02646	1,99367	2,21641	0,14299	2,17910	1,39278	5,48397	1,20225	0,18179	0,05161	4,59301	0,24342	1,98206	2,67043	0,08901	0,02939	1,52990	2,81226
ambiente	0,08368	1,51455	1,85247	0,10416	2,09070	0,36380	3,35152	0,73328	0,32957	0,23481	5,27361	0,29521	2,47004	4,56122	0,20320	0,01144	1,46708	2,20847
capacidade	0,24387	1,54171	1,95415	0,27314	2,13055	0,34952	7,38781	0,98257	0,14537	0,11770	3,90326	0,29149	1,94082	1,82784	0,15168	0,00964	1,45319	2,32347
fase	0,23596	2,10404	1,18424	0,07658	1,94735	1,96548	4,46210	0,82873	0,34867	0,04884	5,62511	0,35813	2,13310	2,31462	0,03754	0,01542	1,48037	2,75853
aumento	0,17583	1,90514	1,82192	0,10572	1,05706	1,17947	5,29310	1,72963	0,24598	0,13701	4,90551	0,49014	2,92361	0,95918	0,06968	0,00964	1,43804	2,31352
criação	0,14391	1,37311	2,18203	0,09907	1,70041	0,25900	3,94006	0,53515	0,10788	0,13397	5,86615	0,25664	1,44774	4,77388	0,28788	0,03581	1,44642	2,36934
nível	0,25148	2,84463	2,25435	0,14077	1,78746	0,83017	4,59597	1,67871	0,20390	0,24123	3,86900	0,50006	1,79759	2,35429	0,12561	0,00000	1,46720	2,01171
adequado	0,09788	2,47603	1,76236	0,10152	1,31117	2,27908	3,39298	1,39995	0,18149	0,16282	5,01359	0,33830	3,26113	1,48354	0,03740	0,01046	1,45685	2,10779
meio	0,11764	2,00804	1,72891	0,09771	1,79192	1,45661	3,22670	1,03877	0,44135	0,38994	5,90336	0,19185	2,55383	1,97305	0,14549	0,02328	1,44303	1,95503
tamanho	0,14247	1,31404	1,37330	0,00000	0,85935	1,49360	7,93577	0,83814	0,20035	3,34572	2,48307	0,31949	1,20887	0,87873	0,08016	0,02187	1,40593	3,10062
dispositivo	0,00000	0,05075	2,93676	0,12445	2,78858	0,22438	6,20374	1,59155	0,04553	0,00000	4,20623	0,00000	0,31971	3,45967	0,17554	0,02108	1,38425	2,33356
problema	0,16723	1,55186	2,59365	0,14507	2,22295	0,80843	4,36901	1,22801	0,14325	0,30538	3,55660	0,53985	2,02352	2,54623	0,20538	0,00000	1,40040	1,90067
existente	0,06514	2,11064	2,16246	0,18608	1,74461	0,71141	4,53087	1,11308	0,16725	0,41510	4,27125	0,20670	2,28974	2,48062	0,04330	0,01181	1,40688	1,87517
parâmetro	0,10390	2,83869	1,81346	0,07371	1,53389	0,76641	3,99735	2,65931	0,30076	0,07409	5,08335	0,29536	2,62695	0,86307	0,14302	0,03728	1,45066	2,13999

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
alteração	0,05159	2,12522	1,83862	0,09691	1,03325	0,89436	5,50151	1,73828	0,15085	0,34880	3,89326	0,60563	2,29095	1,82244	0,05020	0,00000	1,40262	2,04527
montagem	0,00000	0,53268	4,09837	0,12866	2,55051	0,04683	8,95704	1,33017	0,02977	0,46527	2,76174	0,02347	0,36582	0,67853	0,01430	0,00000	1,37395	2,61494
plataforma	0,00000	0,23801	1,55539	0,17104	2,39338	0,47413	3,33293	0,06520	0,06769	0,00000	6,73808	0,00000	0,85117	5,71500	0,25219	0,00000	1,36589	2,64796
especificação	0,02150	1,75812	2,24461	0,09265	1,77311	0,51383	4,12596	2,53585	0,12468	0,08900	4,75071	0,46078	2,00988	1,96335	0,06034	0,01463	1,40869	2,06354
capaz	0,08055	1,30704	2,11253	0,14151	1,97703	0,87307	5,02000	1,04714	0,35296	0,07141	4,50176	0,22671	1,79351	2,24096	0,12594	0,00000	1,36701	2,26754
propriedade	0,03848	1,98737	1,53940	0,11920	0,55193	0,85129	2,18937	3,00574	0,41318	0,09114	4,98312	0,44978	5,46146	0,62934	0,02303	0,04546	1,39871	2,10618
produtivo	0,30598	3,24476	2,07535	0,09776	0,97123	2,13247	4,61836	1,68411	0,15112	0,11532	3,52133	0,81157	2,15488	0,65966	0,00000	0,00964	1,40960	1,94584
óleo	0,00000	3,10969	1,17563	0,05100	1,10171	0,27962	3,12998	0,68740	0,58468	0,00000	9,20459	0,14687	2,48424	0,00000	0,00000	0,00000	1,37221	3,26446
interno	0,04651	1,33854	2,70719	0,11477	1,78978	0,39989	4,66125	2,76390	0,07120	0,47783	3,55841	0,13294	1,63433	1,45701	0,18928	0,00000	1,33393	1,97325
motor	0,00000	0,13370	6,16106	0,06217	1,36403	0,00000	9,47888	0,76122	0,00000	0,02106	1,85667	0,00000	0,28814	0,62205	0,00000	0,00000	1,29681	3,12814
integração	0,01720	0,30677	1,44892	0,24212	2,65972	0,08524	3,21746	0,20654	0,03225	0,00000	5,55683	0,13681	0,41497	6,15287	0,19770	0,00000	1,29221	2,48948
laboratório	0,11558	2,47000	1,88823	0,26113	1,70335	0,67020	3,31968	1,42783	0,40233	0,37170	4,52000	0,65956	3,56450	0,67991	0,12002	0,01046	1,38653	2,30881
dificuldade	0,04903	1,96845	2,12687	0,14597	1,53179	1,34327	3,38073	0,87229	0,13888	0,03370	4,52453	0,31613	2,51883	1,61746	0,07467	0,00000	1,29016	3,15081
carga	0,00000	0,63142	2,09130	0,35881	1,65886	0,33328	7,17683	1,23515	0,07005	0,15119	3,33714	0,33338	1,69009	1,53489	0,12015	0,00000	1,29516	2,16944
viabilidade	0,04018	2,14603	1,96767	0,11860	1,36900	1,02131	4,08519	1,03388	0,35126	0,14038	4,93402	0,43970	1,96298	1,31268	0,15685	0,00000	1,31748	2,20097
matéria	0,03424	2,96713	1,74476	0,07089	0,77817	0,43156	2,71195	1,44563	0,02867	0,24197	5,72566	0,42174	4,64528	0,03941	0,00000	0,02646	1,33210	2,13126
ponto	0,00000	1,26274	1,97812	0,12691	1,94449	0,82750	4,15044	1,02701	0,13897	0,25821	3,53051	0,38032	2,04862	2,62422	0,01863	0,01633	1,27081	1,71058
atividade	0,07117	2,61594	1,42985	0,08635	2,05256	1,08288	3,09424	1,05948	0,12557	0,09698	4,91013	0,40907	1,43717	2,90125	0,13544	0,03324	1,34633	2,00716
inovação	0,04306	1,63010	1,82283	0,05028	1,94493	0,77967	2,94876	0,94992	0,18155	0,13043	4,51402	0,19010	2,40266	2,43379	0,14019	0,00000	1,26014	2,12118
fim	0,09894	1,79606	1,45603	0,17083	1,59819	0,89825	3,80593	1,13173	0,16174	0,10484	4,83097	0,22413	2,08873	2,05661	0,01708	0,01486	1,27843	1,82005
relação	0,06883	1,95602	1,95464	0,17466	1,47092	0,96506	3,56422	1,44834	0,30061	0,10287	4,07656	0,38447	2,40572	1,28686	0,03361	0,00000	1,26209	1,66790
padrão	0,01678	2,62601	1,72349	0,19957	1,75684	1,24730	2,73676	0,64393	0,17065	0,37879	3,38061	0,26235	1,88971	3,45029	0,13640	0,00852	1,28925	1,66205
impacto	0,15230	2,30796	0,96248	0,13846	1,56934	0,50889	2,91002	1,00304	0,30596	0,05483	5,52673	0,57181	2,61844	1,37453	0,04232	0,01110	1,25364	1,84695
ativo	0,05590	0,87189	1,22758	0,09963	1,04499	4,68898	0,19716	0,09016	0,06708	0,00000	7,86931	0,18164	2,23920	1,19200	0,00000	0,00000	1,23910	2,98950
primo	0,03355	2,81945	1,65453	0,05234	0,73930	0,42079	2,55511	1,40333	0,00000	0,23663	5,60466	0,38703	4,38641	0,06714	0,00000	0,02646	1,27417	2,07317
embalagem	0,00000	4,36154	0,48206	0,00000	0,28628	2,32664	1,33541	0,87883	0,00000	0,16258	7,33587	1,05197	2,68208	0,00000	0,00000	0,00000	1,30645	2,49043
atual	0,06027	2,14343	2,83888	0,09224	1,46432	0,27345	3,41172	1,30208	0,14552	0,15898	3,54732	0,38101	2,17192	1,79777	0,12295	0,00000	1,24449	1,58396
energia	0,02372	1,40796	2,58597	0,10737	4,45710	0,12217	3,44707	0,88680	0,09993	0,63761	3,16427	0,26355	1,20021	0,77607	0,13974	0,00000	1,20747	2,02703
item	0,00000	1,28645	0,62072	0,01324	1,37612	0,09507	9,25285	0,59977	0,00000	3,09780	1,56604	0,11265	0,29914	0,99845	0,04117	0,00990	1,21059	3,15825
experimental	0,24916	1,84723	1,58309	0,08354	1,13904	0,49962	3,37600	3,45046	0,36392	0,10320	2,88427	0,41893	1,58296	1,82361	0,11658	0,00000	1,22010	2,04930
completo	0,01859	4,23594	0,96910	0,06279	1,05118	0,74705	5,31043	0,46611	0,16447	0,06816	1,25209	0,06065	0,46003	4,71668	0,02196	0,00000	1,22533	2,11335
identificação	0,10237	2,22941	1,35771	0,03616	1,34833	0,60773	3,36267	0,57136	0,18677	0,04876	4,94599	0,33081	1,67584	2,34936	0,15576	0,00000	1,20681	1,75700
função	0,07028	1,01767	2,27343	0,09822	2,14132	0,59730	3,80179	1,43557	0,14711	0,13512	3,01746	0,20735	1,54047	1,81325	0,05279	0,00000	1,14682	1,64510
composição	0,02219	2,22358	1,66976	0,16599	0,34740	0,52103	2,22084	2,67568	0,33738	0,11925	5,46704	0,38688	2,54621	0,13020	0,01757	0,00000	1,17819	1,87958
comunicação	0,00000	0,20434	1,73610	0,01468	3,21990	0,08515	3,14818	0,30673	0,06390	0,00000	4,06453	0,03249	0,27506	4,77671	0,23237	0,00000	1,13501	2,15221
ar	0,08055	1,40072	1,81337	0,11059	0,92965	0,82640	4,96037	0,56010	0,14218	0,03839	3,53042	0,36163	3,15540	0,53496	0,01662	0,00852	1,15437	1,70109

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
quot	0,10318	2,40854	0,60165	0,02175	0,89853	0,28746	1,97189	1,91202	0,12144	0,00000	8,06704	0,14263	1,70422	1,07822	0,04392	0,00000	1,21016	2,75035
térmico	0,00000	1,20585	3,81528	0,16349	1,49226	0,31104	3,95281	2,67438	0,34576	0,04492	2,14609	0,29526	1,34751	0,11010	0,03418	0,01397	1,12206	1,83581
técnica	0,09787	1,00624	2,09084	0,08802	2,22397	0,92683	1,64337	0,95891	0,15297	0,16514	4,83181	0,18637	0,83776	2,90091	0,05622	0,03719	1,13778	1,82096
físico	0,12888	1,90162	1,49701	0,07537	1,97967	0,84944	2,95338	0,70591	0,21835	0,44462	3,63069	0,50018	2,45934	0,92468	0,07451	0,00990	1,14710	1,48404
equipe	0,06297	0,71882	1,88927	0,20488	2,18882	0,41905	4,56054	0,46940	0,07066	0,40859	3,07197	0,17693	0,97024	2,63378	0,15234	0,00990	1,12551	1,89773
simulação	0,01911	0,46518	2,42634	0,06662	2,37207	0,00000	4,87523	1,86306	0,17426	0,22582	2,93260	0,12737	1,24908	1,13596	0,05495	0,00000	1,12423	2,38901
trabalho	0,02150	1,28443	1,95838	0,12945	1,70726	0,16619	5,10006	0,80449	0,04946	0,12510	3,23043	0,25307	0,95893	2,20772	0,06372	0,00964	1,12936	1,78434
implementação	0,02372	2,08854	1,93087	0,16485	2,17753	0,11534	2,53198	0,63374	0,16399	0,10352	3,90058	0,17831	0,62602	3,48329	0,16755	0,00000	1,14311	1,75136
perda	0,16583	1,85009	2,15045	0,10720	1,57936	0,88668	2,46182	1,10969	0,22675	0,05908	3,34126	0,56772	1,91031	1,21927	0,07354	0,02454	1,10835	1,67532
elaboração	0,00000	1,24275	2,29309	0,18466	1,46489	0,85825	4,88868	0,78045	0,00000	0,40054	3,02204	0,15082	1,31879	1,34085	0,09348	0,01673	1,12850	2,05334
geração	0,26795	2,05685	1,62336	0,06106	3,04672	0,54625	2,39023	0,63809	0,18053	0,04084	3,22007	0,39073	1,40190	2,32835	0,10120	0,00000	1,14338	1,57596
concepção	0,06961	1,59270	2,57274	0,24385	1,57338	0,64340	3,77020	0,70621	0,07886	0,12480	3,16117	0,06162	1,09764	1,58222	0,16455	0,00000	1,09019	1,64811
efeito	0,25687	2,35796	1,58594	0,06222	1,17879	1,72941	1,54889	0,94171	0,52672	0,01872	3,85141	0,50664	2,48903	0,54863	0,01662	0,05422	1,10461	1,51014
vez	0,05056	1,72014	1,40355	0,09416	1,01477	1,47114	2,98589	1,02648	0,20561	0,14619	3,41489	0,41945	1,47927	1,88537	0,03344	0,01665	1,08547	1,63791
parte	0,06591	0,95674	2,36514	0,09681	1,54782	0,99800	3,68530	0,74912	0,00000	0,05485	3,04251	0,59242	1,20439	1,88223	0,07430	0,01263	1,08301	1,55437
algoritmo	0,00000	0,10036	1,01609	0,02172	2,70028	0,07146	1,94196	0,29315	0,06870	0,00000	4,91658	0,02485	0,27169	5,45526	0,24174	0,00000	1,07024	2,34823
conjunto	0,05246	0,63343	2,14702	0,21355	1,37493	0,39273	4,43175	1,39242	0,10704	0,19654	3,19219	0,16472	0,97627	1,85748	0,07283	0,00000	1,07534	1,60416
funcionalidade	0,00000	0,64337	1,24719	0,02011	2,32629	0,09792	2,85758	0,37279	0,00654	0,05880	4,08511	0,06480	0,92264	4,18526	0,18301	0,01568	1,06794	1,75137
diferente	0,16566	2,07459	1,64682	0,09106	0,96648	0,81282	2,59003	0,89722	0,14272	0,13251	3,27082	0,34810	2,39126	1,79197	0,06814	0,00000	1,08689	1,56239
quantidade	0,08935	2,64649	2,05096	0,12413	0,93195	0,61534	3,17491	0,94986	0,17653	0,09096	2,79790	0,34370	2,13107	1,37347	0,04510	0,02358	1,09783	1,53127
eletrônico	0,00000	0,15945	2,26286	0,13973	3,30986	0,14038	5,66376	0,77880	0,05902	0,00000	2,30912	0,15018	0,28071	1,52027	0,03362	0,01144	1,05120	1,83188
piloto	0,01859	2,25830	1,04478	0,08787	1,66727	1,55740	1,68860	1,59143	0,35125	0,11614	4,09910	0,25301	2,85008	1,07051	0,10034	0,00000	1,17217	2,32558
processamento	0,08078	1,46084	0,67625	0,08038	1,72566	0,10789	1,71097	1,00615	0,15862	0,02106	3,98094	0,20410	1,53176	4,08581	0,16186	0,00000	1,06207	1,70624
modo	0,07712	1,18486	1,84493	0,12796	1,55156	0,75150	4,33525	1,14222	0,16710	0,02174	2,74919	0,03520	1,22185	1,44341	0,12515	0,00916	1,04926	1,51670
industrial	0,11050	3,14974	1,38603	0,05947	1,10184	1,52941	1,29714	1,27350	0,42169	0,20093	3,37799	0,47823	3,60973	0,20510	0,01708	0,01463	1,13956	1,63801
usuário	0,00000	0,19981	1,35298	0,07985	2,14828	0,07456	1,59465	0,18573	0,00000	0,10131	4,18196	0,03249	0,59132	5,31441	0,20860	0,00000	1,00412	2,02342
fórmula	0,01911	2,84480	0,24913	0,00000	0,00000	1,22693	0,80124	0,10630	0,03877	0,00000	7,83314	0,02112	3,27611	0,47939	0,00000	0,00000	1,05600	2,38941
diverso	0,12502	1,56855	1,33419	0,19046	1,36345	0,42852	3,43058	0,82929	0,14743	0,06366	2,99756	0,29813	1,29386	2,34185	0,05866	0,00000	1,02945	1,50404
norma	0,00000	0,32609	2,92537	0,09158	1,56406	0,39180	2,77631	1,91664	0,02419	0,23826	2,75674	0,10067	2,08756	0,57618	0,03169	0,00000	0,98795	1,50721
pressão	0,09798	1,12437	2,04587	0,12341	0,48245	0,37034	4,77663	0,99056	0,56886	0,07013	3,17343	0,33686	1,83689	0,03559	0,00000	0,00939	1,00267	1,73205
veículo	0,00000	0,42055	0,52889	0,01802	1,09864	0,80540	8,44328	0,16017	0,02212	0,00000	2,59485	0,00000	0,37819	1,10403	0,11044	0,01110	0,98098	2,16482
consumo	0,06561	2,46647	1,41819	0,08071	1,86337	0,15767	3,22535	0,83500	0,08032	0,01982	3,25582	0,33193	0,80180	1,65679	0,08959	0,01465	1,02269	1,55906
rede	0,00000	0,43571	1,58262	0,02102	4,53045	0,06985	1,91834	0,05523	0,05442	0,00000	3,21968	0,16728	0,30901	2,86822	0,41468	0,00000	0,97791	1,92243
escala	0,06769	2,06717	0,69905	0,04769	0,80481	1,92768	1,30646	1,44877	0,37853	0,05118	4,08397	0,27893	3,11532	0,59156	0,00000	0,00000	1,05430	1,89873
aço	0,00000	0,14191	1,77457	0,09409	0,37260	0,02773	3,92224	4,39286	0,07115	0,15619	3,68685	0,10587	0,80658	0,03448	0,00000	0,00000	0,97419	2,10958
químico	0,16739	1,76543	0,82636	0,12051	0,78100	0,70554	1,24176	1,19001	0,26609	0,10627	4,90006	0,73339	3,27301	0,06923	0,00000	0,01394	1,01000	1,78090

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
manutenção	0,07542	1,30612	1,47128	0,01358	1,49822	0,62557	2,91719	0,45082	0,04304	0,01982	2,31554	0,17648	3,04224	1,43694	0,08934	0,00893	0,96816	1,41146
fluxo	0,04299	0,83242	1,89533	0,07685	1,63217	0,54224	3,35292	0,69625	0,05189	0,02246	3,23292	0,33370	0,59815	2,35170	0,08638	0,00916	0,98485	1,44463
estrutural	0,00000	0,38695	1,28333	0,18953	0,51583	0,29639	8,49321	0,55588	0,02419	0,80563	2,02431	0,02640	0,53378	0,28650	0,02675	0,00000	0,96554	2,40252
cálculo	0,00000	0,41885	2,39832	0,20688	1,58726	0,05546	3,93894	1,07252	0,04497	0,10652	3,09919	0,14931	0,47141	1,86604	0,08824	0,00000	0,96899	1,90763
falha	0,00000	0,30648	1,94999	0,13829	2,29109	0,43930	3,93744	0,56854	0,00000	0,04574	3,15677	0,06408	0,65503	1,73259	0,08100	0,01077	0,96107	1,65427
corte	0,08845	2,56968	2,79770	0,00000	0,32283	0,34773	5,05837	1,43807	0,05529	0,18014	1,64601	0,28881	1,07628	0,09591	0,01576	0,00000	0,99881	1,68472
construção	0,00000	0,31132	1,31384	0,17858	1,40298	0,26466	3,23445	0,83057	0,13535	0,05722	3,45799	0,18685	0,71360	3,31484	0,12169	0,00000	0,97025	1,73054
fonte	0,07502	2,42778	1,71492	0,01597	1,47374	0,15383	0,80904	0,11352	0,00000	0,00000	4,79789	0,15002	2,01425	1,60677	0,19739	0,00000	0,97188	1,63772
resina	0,00000	0,17226	1,35101	0,00000	0,37752	0,07903	3,35919	0,49843	0,37741	0,08008	1,62689	0,16548	7,04584	0,00000	0,00000	0,01993	0,94707	2,05880
disponível	0,02751	1,37476	2,08403	0,06326	1,34299	0,68387	1,72348	0,64638	0,16644	0,09709	2,66217	0,24382	2,58839	1,54306	0,08680	0,00000	0,95838	1,23715
interface	0,00000	0,06437	1,09143	0,18660	2,33013	0,03011	3,49476	0,25377	0,06118	0,00000	2,99586	0,00000	0,34510	3,92311	0,30613	0,00000	0,94266	1,63510
instalação	0,00000	0,80336	1,14770	0,05160	2,20895	0,28216	5,83544	0,42557	0,10598	0,02407	2,10276	0,23673	1,10882	0,90777	0,07908	0,00000	0,95750	1,63776
planta	0,46592	2,74524	0,54609	0,10855	0,74237	0,92139	1,42591	0,33527	0,39167	0,11890	3,42946	0,53499	4,61650	0,07426	0,03377	0,00000	1,03064	1,59588
medição	0,05291	0,62097	1,89788	0,03130	2,96851	0,42666	4,04638	0,83135	0,14532	0,00000	1,82324	0,32920	1,15024	0,76557	0,04993	0,00000	0,94622	1,60418
único	0,07849	0,88076	1,12213	0,10542	1,61760	1,01402	2,59698	0,53028	0,04435	0,10722	2,73564	0,10256	1,04955	2,84198	0,15458	0,01110	0,93704	1,48974
comportamento	0,14984	0,86560	1,54789	0,17120	1,24043	0,50953	3,71969	1,07317	0,32956	0,04067	2,79632	0,32797	1,32538	1,06266	0,05930	0,00000	0,95120	1,44310
desenho	0,02150	0,81145	1,98919	0,07349	0,64689	0,56316	4,37034	0,55938	0,10786	0,19324	3,70493	0,37427	0,60700	1,27059	0,07202	0,00990	0,96095	1,97854
velocidade	0,00000	0,96043	1,95028	0,01392	1,18460	0,28377	4,33374	1,33788	0,10727	0,00000	2,34195	0,32905	1,07995	1,10025	0,03516	0,00000	0,94114	1,45892
ajuste	0,03127	1,46796	1,55214	0,02035	0,99717	1,03149	2,84599	1,07467	0,10188	0,04347	2,30481	0,50150	2,51713	0,83413	0,05259	0,00000	0,96103	1,38759
possibilidade	0,06253	1,05361	1,35808	0,12639	0,94555	0,28333	2,82366	0,78145	0,05981	0,33527	3,59565	0,30853	1,22591	1,57098	0,06665	0,00000	0,91234	1,33177
módulo	0,00000	0,49448	1,45621	0,11447	2,45046	0,00000	3,48159	0,22591	0,07506	0,11905	2,46320	0,08010	0,42143	2,91459	0,13977	0,00000	0,90227	1,59563
fornecedor	0,00000	2,73655	1,84156	0,10740	0,85658	0,82988	3,10125	0,65891	0,06349	0,14485	2,74634	0,18120	1,58177	1,14424	0,02196	0,01144	1,00171	1,55327
obtenção	0,29308	1,40662	1,00443	0,03985	0,55680	1,65221	1,36002	0,84975	0,38059	0,04765	3,91788	0,40574	2,21365	0,50948	0,01863	0,04272	0,91869	1,44076
peso	0,10248	2,69871	1,31226	0,10050	0,58775	0,36425	4,76891	0,59642	0,19183	0,16352	2,30588	0,06348	2,10923	0,25920	0,00000	0,01263	0,97731	1,62014
vida	0,05758	1,65401	2,04252	0,11275	1,40296	0,50215	2,24030	0,97655	0,09748	0,04478	3,26992	0,04080	1,49425	0,79160	0,00000	0,01465	0,92139	1,26807
atendimento	0,00000	0,70161	0,91999	0,06970	1,02507	0,26836	2,32963	1,99261	0,03874	0,15853	3,30188	0,00000	1,07234	2,30159	0,06645	0,00000	0,89041	1,42706
combinação	0,11261	2,28585	0,67583	0,02252	0,46511	2,41321	1,27459	0,37278	0,15774	0,06105	4,10022	0,19011	1,86117	0,70381	0,00000	0,00000	0,91854	1,66932
arquitetura	0,08413	0,15201	0,43370	0,00000	1,74812	0,00000	3,03426	0,07090	0,06130	0,02324	3,49117	0,00000	0,11547	4,67189	0,23770	0,00000	0,88274	1,79422
automático	0,02219	0,99851	1,80223	0,00000	1,26502	0,40925	3,78437	0,73662	0,09865	0,02106	2,55172	0,08869	0,24492	1,89831	0,10763	0,00964	0,87743	1,47784
operacional	0,09439	0,43137	1,20897	0,03153	1,87259	0,20415	2,43174	0,69987	0,25730	0,00000	2,92148	0,51586	1,26385	2,03889	0,02049	0,01046	0,87518	1,24671
brasil	0,15892	1,37932	1,03995	0,02628	1,37832	0,69710	2,81071	0,38677	0,02277	0,03746	3,13089	0,32190	1,67438	0,93339	0,01808	0,01521	0,87697	1,26697
adequação	0,00000	1,64998	1,34466	0,02175	0,91237	1,32657	3,28491	1,00813	0,09607	0,06197	1,81259	0,36017	1,21518	0,96815	0,04867	0,00000	0,88195	1,22743
real	0,06518	0,62804	1,29277	0,07285	1,76454	0,28704	2,40868	0,36406	0,15874	0,05675	3,20019	0,05922	0,66031	2,59940	0,13627	0,00000	0,85963	1,38606
proteção	0,04544	0,32942	2,39901	0,16388	2,15950	0,56559	2,35860	0,32423	0,04994	0,06672	3,00075	0,26961	1,27326	0,44203	0,04099	0,04385	0,84580	1,44262
longo	0,19235	1,48845	1,07008	0,06914	1,28433	1,25406	2,53744	0,64817	0,20478	0,03797	2,17419	0,21817	2,06860	0,52335	0,10280	0,00000	0,86712	1,31176
amostra	0,08777	2,00272	1,00146	0,06110	0,46228	0,37182	1,49735	0,98991	0,39485	0,03744	4,50296	0,42090	2,56314	0,46178	0,00000	0,00000	0,92847	2,04804

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
primeiro	0,10715	1,43287	1,32504	0,02628	1,54581	1,48594	2,59963	0,70002	0,14041	0,07590	2,03954	0,22157	0,98636	1,40053	0,01808	0,00000	0,88157	1,19774
distribuição	0,03141	1,26700	1,54288	0,16602	2,26941	0,61790	2,42194	0,61628	0,19269	0,03797	2,12831	0,11294	0,84200	1,32850	0,00000	0,00000	0,84845	1,21543
volume	0,00000	1,39072	0,99135	0,18547	0,43233	0,38589	1,86517	0,40136	0,07155	0,05085	3,75781	0,09499	1,27095	2,64977	0,05455	0,00000	0,85017	1,57496
variação	0,10238	1,99513	1,89493	0,22309	0,91221	0,57210	2,42047	1,06596	0,17133	0,11899	2,08529	0,27188	1,34186	0,30643	0,09882	0,01144	0,84952	1,19882
funcional	0,00000	0,70590	1,09572	0,10531	1,17914	0,30087	3,73488	0,42625	0,05323	0,15946	2,99516	0,04526	0,65802	1,77674	0,12865	0,00990	0,83591	1,59937
execução	0,00000	0,58413	1,41942	0,25306	1,31066	0,35764	2,02942	1,02231	0,04511	0,03243	2,51884	0,00000	0,41444	3,22787	0,11018	0,01915	0,83404	1,33845
líquido	0,00000	2,19959	0,44280	0,00000	0,39287	0,76563	1,94348	0,80569	0,19132	0,00000	3,98150	0,05957	2,70672	0,20017	0,00000	0,00000	0,85558	1,54239
química	0,02372	0,72125	0,44291	0,03297	0,15420	0,84724	1,54682	2,39179	0,24248	0,02042	3,82869	0,24948	3,10263	0,00000	0,00000	0,00852	0,85082	1,53385
ação	0,12620	1,62931	0,72900	0,00000	1,19660	1,41247	1,11487	0,72890	0,03518	0,04317	3,54778	0,18665	1,87852	1,28298	0,00000	0,00000	0,86948	1,30067
placa	0,00000	0,51328	2,49142	0,11308	2,69398	0,02773	2,08060	1,56047	0,00000	0,03642	2,22333	0,02725	0,56703	0,74354	0,19394	0,00000	0,82951	1,35333
serviço	0,00000	0,08881	0,27939	0,20303	1,24684	0,18922	0,93903	0,06911	0,00000	0,02106	4,19978	0,00000	0,51124	4,91986	0,35682	0,00000	0,81401	1,84598
funcionamento	0,00000	0,66077	1,83954	0,13024	1,04466	0,35705	4,10398	0,58348	0,13364	0,04249	1,86101	0,12853	0,48473	1,65541	0,12855	0,01263	0,82292	1,33655
acordo	0,15353	0,87652	0,93096	0,07290	0,82176	1,52728	2,32862	0,47665	0,07686	0,03914	2,45345	0,27945	1,56969	1,62358	0,04160	0,02089	0,83081	1,12780
massa	0,00000	3,66469	1,24183	0,20142	0,22568	0,28396	1,70015	0,69665	0,22433	0,00000	2,42165	0,25961	2,30036	0,52158	0,00000	0,00000	0,85887	1,39384
demanda	0,04316	0,74222	0,66549	0,07182	1,09025	0,09396	3,43931	0,68924	0,02419	0,04492	3,66418	0,19801	0,82101	1,16769	0,07400	0,00000	0,80184	1,36332
ganho	0,07198	1,61527	1,45786	0,05568	0,86589	0,15707	3,27138	0,99124	0,00654	0,05103	1,55895	0,57852	1,61639	0,91941	0,09470	0,01181	0,83273	1,24496
ideal	0,04136	2,26702	1,18261	0,13028	0,15913	0,66204	1,74195	0,79428	0,14080	0,18055	3,30142	0,31995	1,95661	0,23008	0,00000	0,03488	0,82144	1,38258
partir	0,04175	1,16511	0,69407	0,04857	1,61062	0,68531	1,82030	1,16141	0,17685	0,05291	3,04001	0,19501	1,39609	0,97374	0,05202	0,00893	0,82017	1,26448
agente	0,03424	0,67634	0,33927	0,03531	1,02474	1,03657	0,33136	0,30602	0,00000	0,00000	6,06894	0,44386	2,04987	0,29602	0,00000	0,00000	0,79016	1,75299
variável	0,11021	0,91975	1,79210	0,05364	1,20198	0,38214	1,98303	0,74260	0,11530	0,05394	2,75508	0,38337	0,99984	1,31645	0,01808	0,00000	0,80172	1,20878
perfil	0,02293	2,17941	0,64765	0,03945	0,51752	1,83635	2,29183	1,33498	0,00000	0,06585	2,52884	0,15493	0,85521	0,58556	0,00000	0,00000	0,81628	1,28149
mecanismo	0,10751	0,26652	0,97529	0,01429	0,90489	0,26382	2,59396	0,60295	0,08277	0,13633	3,26183	0,00000	0,56534	2,51155	0,11305	0,00964	0,77561	1,37057
eficácia	0,10205	0,68883	0,45425	0,00000	0,31365	4,24143	0,95221	0,07302	0,00000	0,00000	4,03920	0,18069	1,33098	0,19563	0,01479	0,00000	0,78667	1,98331
seleção	0,23877	1,15176	0,65965	0,04951	0,65356	0,84487	2,69033	0,17342	0,24938	0,01872	3,50469	0,44551	1,24484	0,66106	0,01576	0,00893	0,78817	1,62526
recurso	0,00000	0,41828	0,99561	0,11021	1,43094	0,42638	1,65103	0,51007	0,02497	0,12643	3,05842	0,25676	0,38402	2,94104	0,13365	0,00000	0,77924	1,30427
ciclo	0,23443	0,34158	1,36263	0,08457	0,84459	0,14520	2,30053	1,70064	0,07050	0,03151	1,86825	0,10413	1,14766	1,98309	0,03892	0,00000	0,76614	1,22371
lote	0,00000	1,21418	1,12654	0,04852	0,79954	1,92862	1,01901	1,39420	0,05458	0,12176	2,75814	0,32528	1,55081	0,60205	0,01500	0,00000	0,80989	1,95742
camada	0,00000	0,86002	0,83947	0,11736	0,53986	0,31369	1,87153	0,61473	0,11898	0,01872	2,92999	0,07160	1,28798	2,55533	0,04480	0,00000	0,76150	1,31205
injeção	0,00000	0,54624	2,02044	0,08066	0,79270	0,13950	3,13578	1,33124	0,25976	0,21351	2,91700	0,00000	0,59099	0,06076	0,00000	0,00000	0,75554	1,41047
ingrediente	0,03876	5,37562	0,00000	0,00000	0,00000	0,42101	0,03691	0,00000	0,00000	0,00000	5,65747	0,00000	2,10891	0,00000	0,00000	0,00000	0,85242	2,17784
rápido	0,00000	0,88117	1,00545	0,03030	0,79029	0,63442	2,51058	0,50359	0,09479	0,01821	2,26004	0,00000	1,50315	1,73034	0,03479	0,01110	0,75051	1,22940
mistura	0,00000	3,75968	0,70220	0,21437	0,19571	0,97293	0,85749	0,65075	0,54153	0,00000	2,95822	0,27716	1,97212	0,07284	0,00000	0,00000	0,82344	1,50035
experimento	0,26436	2,00614	1,58919	0,01697	0,18806	0,75228	2,24372	1,07512	0,17405	0,00000	2,12614	0,42620	1,36514	0,64037	0,00000	0,00000	0,80423	1,60579
eficiente	0,09788	1,10101	1,14195	0,13193	1,04079	0,30851	2,12614	0,47740	0,27358	0,03254	2,23513	0,25569	1,45915	1,12643	0,06585	0,01110	0,74282	1,11359
implantação	0,03609	0,64721	1,52636	0,13407	1,99657	0,25626	1,53636	0,18635	0,02277	0,11592	2,54560	0,25344	0,94844	1,76061	0,15053	0,00964	0,75789	1,17862
ano	0,17579	1,06256	0,76166	0,03504	0,97847	1,26423	1,66045	0,59089	0,15280	0,12292	3,07621	0,19757	0,98482	0,97207	0,01863	0,00000	0,75338	1,40037

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
externo	0,02150	0,82522	1,92468	0,08244	0,72569	0,30151	2,44695	0,81274	0,06130	0,10460	2,12503	0,25876	1,16142	0,92560	0,09475	0,00000	0,74201	1,10363
otimização	0,00000	1,19573	1,04132	0,06004	1,24086	0,69865	1,70811	0,78188	0,12860	0,05417	2,11131	0,28884	1,11558	1,44289	0,14637	0,00964	0,75150	1,00226
tensão	0,00000	0,09936	3,33648	0,12339	2,79932	0,07152	2,39364	0,62447	0,04470	0,02174	1,16972	0,02640	0,63170	0,25363	0,03051	0,01415	0,72755	1,51896
bancada	0,00000	1,01891	0,87023	0,01358	0,48686	1,22721	3,13527	0,83159	0,14059	0,01925	2,75282	0,08147	1,76686	0,12408	0,01500	0,00000	0,78023	1,79092
cor	0,06610	2,97205	1,69912	0,00000	0,03385	0,15619	0,76312	0,31300	0,00000	0,07155	3,73367	0,39734	1,82748	0,31812	0,00000	0,02241	0,77338	1,39378
formação	0,01859	1,91482	0,76655	0,05378	0,56014	0,54540	1,04757	1,47426	0,31056	0,12448	2,67208	0,26120	2,04600	0,19296	0,00000	0,01929	0,75048	1,24362
fixação	0,00000	0,25875	2,27272	0,08925	0,36113	0,28687	3,45937	1,11857	0,02580	0,30332	2,64454	0,17489	0,61893	0,00000	0,00000	0,01046	0,72654	1,53847
teor	0,00000	2,55123	0,34987	0,03912	0,18934	0,75236	0,06407	1,02224	0,59449	0,05725	3,67552	0,15759	2,51418	0,04086	0,00000	0,00000	0,75051	1,35342
nacional	0,04333	1,00308	1,24814	0,05587	1,71128	0,78421	1,63679	0,54720	0,13901	0,08242	1,83937	0,39197	1,32964	0,83103	0,04783	0,00852	0,73123	1,04726
banco	0,00000	0,46723	0,54741	0,01551	1,28741	0,00000	1,43796	0,22530	0,09343	0,00000	2,92119	0,04701	0,45966	3,95495	0,25347	0,01110	0,73260	1,35982
região	0,36537	0,82569	1,06279	0,23681	1,07103	0,04215	1,19018	0,85227	0,07898	0,02496	4,21778	0,20571	1,00310	0,45407	0,02673	0,00000	0,72860	1,38172
painel	0,00000	0,26405	1,64016	0,00000	1,19200	0,16080	3,83962	0,07025	0,02580	0,59372	2,81261	0,05120	0,40264	0,60812	0,01757	0,00000	0,72991	1,38972
matériasprima	0,00000	1,72442	0,64045	0,00000	0,00000	0,42149	2,77534	0,13831	0,15157	0,14995	2,88717	0,19566	3,13592	0,00000	0,00000	0,02099	0,76508	1,46510
valor	0,00000	1,52931	1,36177	0,08706	1,19502	0,23077	2,00694	0,50054	0,03686	0,06195	2,25700	0,28798	1,07820	1,03290	0,08232	0,00000	0,73429	0,98814
sensor	0,00000	0,54659	0,76742	0,00000	2,40416	0,00000	3,55322	0,20413	0,28538	0,00000	2,52039	0,05045	0,51508	0,59743	0,07426	0,01144	0,72062	1,34201
unidade	0,07167	0,67343	1,03384	0,05731	1,18547	0,23110	2,09782	0,57175	0,16479	0,05762	2,73383	0,41619	2,04657	0,44809	0,07645	0,00000	0,74162	1,10059
durabilidade	0,00000	0,63813	1,49354	0,02909	0,29932	0,06199	5,12597	0,77713	0,05669	0,28250	1,88571	0,07312	0,77482	0,11575	0,00000	0,00000	0,72586	1,48513
monitoramento	0,07703	0,49393	0,80818	0,06110	2,59210	0,07844	1,31932	0,38821	0,07372	0,05912	2,85404	0,27378	0,85755	1,62720	0,06371	0,02358	0,72819	1,25505
ambiental	0,20257	0,84938	0,53865	0,03818	1,66138	0,23493	1,16654	0,35526	0,10293	0,02174	4,35508	0,67500	1,13052	0,09519	0,00000	0,01110	0,71490	1,43398
configuração	0,00000	0,33639	1,73131	0,04261	1,10097	0,03011	2,65255	0,25755	0,06193	0,19750	2,04012	0,07610	0,50920	2,04603	0,24296	0,00000	0,70783	1,13263
programa	0,10038	1,40495	0,67059	0,08234	1,12997	0,36448	1,77937	1,18131	0,00000	0,00000	3,02514	0,02485	0,18714	1,47677	0,02685	0,00000	0,71588	1,16900
procedimento	0,09169	0,69652	1,25876	0,08920	1,33775	0,68823	2,00196	0,49774	0,09285	0,00000	2,69022	0,04229	0,80422	1,06236	0,05894	0,00000	0,71330	1,08778
determinação	0,00000	0,99029	0,90510	0,06469	1,05089	0,66282	2,02501	0,58819	0,18349	0,04413	2,59635	0,20156	2,36937	0,25070	0,00000	0,00000	0,74579	1,32012
revestimento	0,00000	0,13192	0,65039	0,22926	0,06571	1,42134	2,50923	2,41824	0,00000	0,17778	1,34632	0,14802	2,15227	0,00000	0,00000	0,00000	0,70316	1,31265
adaptação	0,15445	0,76947	0,74601	0,17764	0,79701	0,12138	3,62129	0,07731	0,11888	1,02751	1,48200	0,22420	1,13539	0,83192	0,07629	0,00000	0,71005	1,39459
índice	0,00000	1,00824	0,33313	0,24997	1,56810	0,30795	1,69137	0,32090	0,11798	0,08532	3,01441	0,10618	1,56198	0,98727	0,00000	0,00000	0,70955	1,12452
molde	0,00000	0,70382	1,64890	0,00000	0,51949	0,00000	3,59104	0,97217	0,15589	0,11751	3,08817	0,07345	0,48979	0,02758	0,00000	0,00000	0,71174	1,40908
alternativa	0,04586	1,24139	1,92152	0,05189	1,03658	0,26087	1,88778	0,49624	0,11627	0,04284	2,13235	0,28045	1,54631	0,40325	0,00000	0,00000	0,71648	1,08525
maneira	0,05238	0,93891	1,12703	0,03756	0,67029	0,30095	2,49426	0,65668	0,05954	0,09661	1,85522	0,23666	0,85776	1,63244	0,07446	0,00000	0,69317	1,00968
natural	0,09654	2,17053	0,65296	0,04207	0,82048	0,40028	1,10238	0,38097	0,22733	0,00000	2,26441	0,48009	2,05993	0,51665	0,03757	0,02567	0,70487	1,14283
barra	0,00000	1,64178	1,10015	0,09771	0,29653	0,08936	1,95870	3,18722	0,16768	0,00000	1,64707	0,00000	0,55261	0,52719	0,00000	0,00000	0,70412	1,38814
caso	0,04586	0,76626	1,27304	0,04639	1,08413	0,56730	1,62748	0,47647	0,14898	0,04008	2,13289	0,24082	1,08999	1,49381	0,08149	0,00000	0,69469	0,94768
gestão	0,02023	0,13276	0,65101	0,00000	1,58779	0,04053	0,79968	0,00000	0,00000	0,00000	3,53015	0,11650	0,15106	3,85120	0,09970	0,00000	0,68629	1,44763
substituição	0,04586	1,90634	1,04674	0,04627	0,88321	0,23622	2,16044	0,74649	0,05515	0,20914	1,98499	0,32004	1,16911	0,47167	0,04009	0,00000	0,70761	1,04025
fio	0,00000	0,04339	4,14538	0,00000	1,16007	0,05269	1,58499	0,36703	0,02346	0,02407	2,21939	0,22869	0,52768	0,46796	0,00000	0,05331	0,68113	1,60037
busca	0,20472	0,96114	1,21157	0,05973	0,59186	0,27295	2,13744	0,27345	0,06034	0,05776	2,06784	0,24928	1,20778	1,75499	0,06089	0,00000	0,69823	1,01719

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
tinta	0,00000	0,12363	1,66162	0,00000	0,06361	0,03011	1,48211	0,35910	0,11734	0,00000	1,60718	0,13916	5,13471	0,13740	0,00000	0,00852	0,67903	1,74003
protocolo	0,00000	0,22892	0,95599	0,03153	2,03276	0,78458	1,05404	0,13633	0,01985	0,00000	1,97272	0,18347	0,56976	2,63195	0,34109	0,00000	0,68394	1,15370
engenharia	0,01859	0,44963	1,97581	0,06133	0,69311	0,16020	3,57467	0,84349	0,05795	0,20401	1,69248	0,00000	0,52880	0,74532	0,06474	0,02089	0,69319	1,32221
mudança	0,06961	0,85402	0,46853	0,00000	0,68704	0,06602	3,01603	0,58658	0,11889	0,20863	1,76735	0,21139	0,94171	1,74317	0,05613	0,00000	0,67469	1,00945
concentração	0,03943	1,94602	0,40072	0,01971	0,21527	1,68069	0,64760	0,56070	0,13904	0,00000	2,98013	0,30310	2,30748	0,15768	0,00000	0,00000	0,71235	1,42220
aquisição	0,02978	1,80069	1,03753	0,04755	0,97385	0,39105	2,51132	0,34889	0,05338	0,15961	1,61765	0,02347	1,09842	0,54437	0,01863	0,00000	0,66601	1,24125
crítico	0,02219	0,75384	0,91441	0,02175	1,02999	1,00700	2,21253	0,61816	0,06451	0,04352	1,81741	0,12804	0,83128	1,27779	0,03856	0,01181	0,67455	0,99541
grupo	0,12882	0,73969	0,95309	0,01971	0,99140	1,35562	1,60202	0,50389	0,08665	0,03656	2,35976	0,18267	1,05426	1,06520	0,00000	0,00000	0,69246	1,01260
verificação	0,00000	1,21169	0,67059	0,08239	1,26049	0,11313	2,21931	1,10150	0,02346	0,08955	2,84512	0,15898	0,98865	0,78645	0,00000	0,01181	0,72269	1,44454
resíduo	0,04050	1,31864	1,04887	0,14151	0,47556	0,77267	0,78509	0,85003	0,22200	0,01925	3,59566	0,15184	1,25903	0,07118	0,03757	0,00000	0,67434	1,14847
brasileiro	0,18005	0,86369	0,36708	0,00000	0,77534	0,48556	1,74119	0,09252	0,00000	0,02407	3,94114	0,11209	1,16764	0,93809	0,04704	0,00000	0,67097	1,41219
presente	0,09069	1,13136	0,74991	0,02336	0,98035	0,61322	1,68209	0,40514	0,19745	0,07364	2,47331	0,36174	1,13855	0,77282	0,00000	0,01046	0,66901	0,97714
momento	0,01965	0,50566	0,98889	0,10907	0,69470	0,72973	1,65823	0,37928	0,10263	0,00000	2,73067	0,11415	1,38393	1,10244	0,01983	0,00000	0,65868	1,17751
número	0,13662	0,65920	0,96752	0,04418	0,79364	0,59237	1,54486	0,90973	0,05364	0,01773	1,92180	0,22940	1,39852	1,20646	0,12838	0,00000	0,66275	1,02556
transporte	0,01678	1,02417	0,53859	0,13292	0,67684	0,16454	2,99654	0,46790	0,14051	0,04352	2,60969	0,41230	0,88097	0,70323	0,02967	0,00000	0,67738	1,01446
modelagem	0,00000	0,22677	1,19845	0,01468	1,50588	0,09580	1,95533	0,27866	0,10314	0,06418	2,19478	0,10898	0,78853	1,87942	0,06003	0,00000	0,65466	1,16593
geometria	0,00000	0,08080	1,52154	0,09245	0,57253	0,00000	3,94945	2,22973	0,02346	0,10318	1,41906	0,06378	0,27154	0,07206	0,00000	0,00000	0,64997	1,44989
fibra	0,00000	1,89216	1,31342	0,09292	0,72821	0,12094	1,42815	0,10449	0,17716	0,06774	2,07198	0,68412	1,67809	0,27198	0,05803	0,05718	0,67166	1,02125
superfície	0,00000	0,54307	1,37347	0,12838	0,28152	0,17900	1,94449	1,17227	0,24094	0,05291	1,66074	0,28095	2,52349	0,08965	0,00000	0,00000	0,65443	1,15459
programação	0,00000	0,95144	1,36339	0,08803	0,97052	0,00000	2,20010	0,62107	0,00000	0,05394	2,15197	0,00000	0,14559	2,22954	0,05930	0,00000	0,67718	1,16464
manual	0,02150	0,47233	1,65984	0,04705	1,27842	0,27640	2,35030	0,61351	0,00000	0,09278	2,22657	0,08251	0,47785	0,83100	0,06486	0,00000	0,65593	1,08029
secagem	0,04744	2,19398	1,15314	0,02577	0,08770	1,84937	0,89256	0,10951	0,21755	0,04347	1,04107	0,35392	2,89407	0,00000	0,00000	0,00000	0,68185	1,39752
compatibilidade	0,00000	0,20497	0,69098	0,03288	0,59703	1,23015	1,02448	0,06963	0,17560	0,02246	3,28528	0,19647	1,66680	1,12033	0,07403	0,00000	0,64944	1,41693
fator	0,04817	1,65546	0,96367	0,15338	1,26186	0,66848	1,94419	0,19212	0,07221	0,04352	1,75124	0,23537	1,14495	0,39565	0,04911	0,01308	0,66203	1,04084
limpeza	0,00000	0,60801	0,92448	0,00000	0,40880	0,21271	1,05952	0,91686	0,06130	0,00000	3,30492	0,14115	2,67729	0,00000	0,00000	0,00000	0,64469	1,29961
imagem	0,00000	0,04018	0,95981	0,00000	1,73942	0,16423	1,22720	0,10795	0,00000	0,03850	3,23256	0,38398	0,25783	2,06292	0,01381	0,01046	0,63993	1,33521
metálico	0,00000	0,56351	1,27215	0,23839	0,45627	0,10626	2,53839	1,21909	0,08684	0,15774	2,30089	0,11314	1,06513	0,14739	0,04815	0,00000	0,64458	1,12332
reação	0,00000	1,41593	0,39796	0,08203	0,25385	0,77495	0,51094	0,42000	0,08743	0,04634	2,44462	0,11667	4,16359	0,03559	0,00000	0,00000	0,67187	1,45045
útil	0,02372	1,06290	1,56269	0,06798	1,01182	0,05741	1,67484	0,81778	0,07936	0,02496	2,63163	0,02166	1,13627	0,25598	0,01618	0,01465	0,65374	1,02953
transmissão	0,00000	0,07236	0,78968	0,00000	2,59864	0,01437	2,93860	0,20001	0,00000	0,00000	1,51331	0,07920	0,36441	1,30332	0,23963	0,00000	0,63210	1,24570
insumo	0,07954	1,38640	0,82321	0,00000	0,51952	1,84549	0,41629	0,17156	0,13525	0,13705	3,60945	0,08921	1,42750	0,08304	0,01218	0,00000	0,67098	1,30500
formato	0,05421	1,50165	1,25984	0,05119	0,34011	0,10273	1,20373	0,37366	0,11329	0,22066	2,67046	0,21558	0,71649	1,56526	0,07358	0,00000	0,65390	0,96887
usinagem	0,00000	0,13848	2,11813	0,03604	0,05611	0,00000	3,86565	2,79441	0,00000	0,23248	0,89324	0,00000	0,02819	0,00000	0,00000	0,00000	0,63517	1,53992
sensorial	0,01965	4,95653	0,07781	0,00000	0,14358	0,37778	0,03433	0,00000	0,07501	0,00000	4,70656	0,11963	1,91858	0,00000	0,00000	0,00000	0,77684	2,08160
local	0,02023	0,47590	1,02859	0,22472	1,31692	0,82262	1,33721	0,30665	0,07618	0,00000	2,41661	0,19813	0,87579	0,99113	0,01863	0,00000	0,63183	0,89568
dimensional	0,00000	0,20215	1,72801	0,06397	0,47779	0,03099	3,37162	2,52106	0,00000	0,01925	1,16673	0,05065	0,24811	0,13426	0,00000	0,00000	0,62591	1,38994

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
contato	0,00000	0,53145	1,49009	0,05802	1,26034	0,30697	1,88185	0,40000	0,14445	0,02592	1,92402	0,45333	1,16632	0,39573	0,04655	0,00000	0,63031	0,99593
matriz	0,11214	1,06283	1,39438	0,00000	0,67321	0,37001	1,32473	1,50639	0,02277	0,66583	1,38405	0,32007	1,09149	0,26511	0,00000	0,01786	0,63818	0,87844
aplicativo	0,00000	0,04350	0,76205	0,02172	1,23562	0,00000	0,85090	0,00000	0,00000	0,00000	3,30361	0,03379	0,06857	3,48802	0,12450	0,00000	0,62077	1,45067
especial	0,01911	0,62297	1,68223	0,06553	0,77233	0,26088	2,30350	0,85971	0,09562	0,05272	1,58028	0,24953	1,17015	0,26600	0,00000	0,02506	0,62660	0,94562
corrente	0,00000	0,15015	2,13652	0,11977	2,17422	0,05496	2,79218	0,39107	0,00000	0,02407	1,19356	0,00000	0,66036	0,22486	0,01537	0,00000	0,62107	1,25318
pó	0,00000	2,59517	0,33774	0,03748	0,16196	0,69423	0,86507	0,75811	0,06641	0,00000	2,49878	0,26528	2,25113	0,00000	0,00000	0,00000	0,65821	1,35809
interação	0,06930	1,11615	0,44996	0,06791	0,87391	1,11321	1,25995	0,04536	0,08323	0,03520	2,29375	0,11331	1,26849	1,21501	0,05600	0,00000	0,62880	0,91204
polímero	0,00000	0,40428	0,79628	0,01802	0,20799	0,99196	0,50451	0,58840	0,20032	0,13232	2,81251	0,19450	3,07532	0,00000	0,00000	0,00000	0,62040	1,20091
acesso	0,00000	0,15757	0,57568	0,13628	1,52946	0,31685	1,31132	0,16135	0,02212	0,02324	2,43218	0,08402	0,03927	2,91181	0,12110	0,00000	0,61389	1,24735
plástico	0,00000	0,43302	1,23323	0,01810	0,81168	0,15256	3,15880	0,83071	0,00000	0,25017	1,95126	0,13524	0,87528	0,07118	0,00000	0,00000	0,62008	1,15563
gás	0,00000	0,31182	0,74762	0,04190	1,01125	0,23285	2,72028	0,89710	0,28958	0,02407	1,96865	0,00000	1,44900	0,07007	0,00000	0,00000	0,61026	1,11135
circuito	0,00000	0,16386	2,50670	0,01646	2,74267	0,00000	2,70957	0,37803	0,11081	0,00000	0,45857	0,19346	0,07922	0,34534	0,05647	0,00000	0,61007	1,35443
adição	0,00000	2,31520	0,30862	0,09635	0,39504	0,84881	0,58166	1,47830	0,24466	0,05085	1,88016	0,19102	1,57877	0,29164	0,00000	0,00000	0,64132	1,05689
grau	0,10908	0,86072	1,30845	0,07779	0,91706	0,49698	1,53473	0,96677	0,17975	0,04574	1,41024	0,12377	1,13145	0,66061	0,00000	0,00000	0,61395	0,92195
modificação	0,03439	1,21164	0,92881	0,01911	0,36251	0,32778	2,28138	0,89380	0,02497	0,03018	1,45117	0,09841	1,66587	0,50907	0,00000	0,00000	0,61494	0,88057
dimensão	0,00000	0,39199	1,66732	0,21046	0,83251	0,00000	3,39668	0,83532	0,00000	0,26186	1,53968	0,00000	0,25312	0,28098	0,02478	0,00000	0,60592	1,17027
planejamento	0,00000	0,26173	0,52652	0,04004	0,98864	0,42048	1,38076	1,07210	0,13233	0,06161	1,82719	0,09837	0,75654	2,10922	0,03942	0,02207	0,60856	1,15266
seguinte	0,08261	0,96224	0,86895	0,07850	1,00416	0,30404	1,80461	0,34450	0,21495	0,03060	1,82769	0,11113	0,99985	1,49190	0,05557	0,00893	0,63689	1,07852
design	0,00000	0,07916	1,24029	0,00000	0,96118	0,12911	1,92285	0,47402	0,00000	0,07940	3,38253	0,03379	0,19526	0,96952	0,00000	0,00000	0,59169	1,15181
comprovação	0,00000	1,74364	1,04955	0,09191	0,48175	1,40888	1,40543	0,32350	0,12485	0,15260	1,73828	0,02485	0,84957	0,36068	0,03349	0,00000	0,61181	1,12008
comercial	0,10721	0,86074	0,70680	0,00000	1,27254	0,01077	2,22263	0,03597	0,09803	0,02324	1,96865	0,23417	1,26332	1,00066	0,00000	0,02003	0,61405	1,01183
alumínio	0,00000	0,09793	1,80396	0,00000	0,31812	0,22291	2,01776	1,19861	0,11030	0,15029	2,45376	0,10091	0,99830	0,00000	0,00000	0,00000	0,59205	1,16961
dia	0,07765	2,23824	0,60383	0,06453	0,95995	1,62852	0,64030	0,14324	0,00000	0,00000	1,59194	0,00000	1,92362	1,14718	0,00000	0,00000	0,68869	1,27145
levantamento	0,00000	0,86782	1,05274	0,09952	1,14915	0,13562	1,76106	0,34018	0,00654	0,00000	2,24882	0,11382	0,88400	1,45765	0,01537	0,01263	0,63406	1,58672
gerenciamento	0,00000	0,21369	1,42732	0,05829	1,09270	0,03011	1,33875	0,12194	0,06966	0,02106	2,15356	0,01628	0,08680	2,71675	0,12440	0,00000	0,59196	1,15838
potência	0,00000	0,04520	2,52601	0,03194	2,41128	0,05629	2,72126	0,13604	0,03225	0,00000	0,87651	0,03129	0,19504	0,24137	0,05006	0,00000	0,58466	1,26036
inicial	0,02219	1,19315	0,94684	0,10117	0,79289	1,01024	1,63057	0,77731	0,06278	0,00000	1,39596	0,17824	1,16241	0,91654	0,03063	0,00000	0,63881	0,99893
aspecto	0,03620	1,26585	1,12538	0,05250	0,99888	0,33611	1,10737	0,27233	0,10863	0,00000	1,88973	0,32216	1,54812	0,65997	0,01983	0,02247	0,61035	0,90459
caixa	0,00000	0,83394	1,67740	0,00000	0,77738	0,00000	2,57862	0,79528	0,02765	0,04998	1,17444	0,16686	0,54628	0,64125	0,04392	0,00000	0,58206	0,97182
hardware	0,00000	0,04350	0,63342	0,04713	3,00266	0,00000	2,07158	0,00000	0,00000	0,00000	1,34113	0,00000	0,18877	1,76728	0,12173	0,00000	0,57608	1,19052
suporte	0,00000	0,16464	0,85269	0,04222	0,97947	0,14808	2,34587	0,55443	0,09175	0,85604	1,27299	0,03017	0,71297	1,20771	0,06528	0,00000	0,58277	0,84983
terceiro	0,15169	0,62015	0,55163	0,00000	0,62401	0,37284	1,53268	0,96152	0,02346	0,02496	2,62251	0,19395	0,56688	1,29631	0,00000	0,00000	0,59641	1,27842
potencial	0,29807	0,99541	0,58186	0,03949	0,84004	0,53042	1,09683	0,36909	0,20024	0,02246	2,02417	0,48905	1,63457	0,38239	0,00000	0,00000	0,59401	0,87159
caracterização	0,11320	0,44209	0,36383	0,04444	1,22884	0,38227	1,37047	0,84313	0,34960	0,01821	2,35888	0,12442	1,55028	0,14920	0,01119	0,00000	0,58438	1,07546
consumidor	0,10560	2,53080	0,24680	0,00000	1,12505	0,33058	0,34777	0,24893	0,00000	0,05800	2,87001	0,04722	1,30193	0,60824	0,00000	0,00000	0,61381	1,08244
seguro	0,00000	0,67452	0,58634	0,12743	0,54425	0,71458	1,78509	0,12277	0,04730	0,03340	2,34646	0,13117	0,35348	1,62179	0,04059	0,00000	0,57057	1,00977

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
aditivo	0,00000	2,61355	0,27369	0,10241	0,06372	0,30817	0,38367	0,11352	0,77548	0,03476	1,55234	0,15650	3,39665	0,00000	0,00000	0,00990	0,61152	1,19796
coleta	0,10927	0,73878	0,83195	0,05012	1,14448	0,29869	1,55157	0,58517	0,07072	0,00000	2,13391	0,26249	0,40719	1,14336	0,02562	0,00000	0,58458	0,95227
automação	0,04901	1,17634	0,97990	0,02628	0,85166	0,12507	1,92275	0,43722	0,00000	0,00000	1,67448	0,07464	0,18185	1,66299	0,04029	0,00000	0,57515	0,88077
linguagem	0,00000	0,40971	0,42960	0,03563	0,92184	0,00000	0,70755	0,39784	0,02419	0,00000	2,18786	0,00000	0,09965	3,65985	0,08993	0,00000	0,56023	1,29129
solo	0,39257	0,85678	0,28633	0,45937	0,81546	0,00000	2,27632	0,38155	0,11367	0,00000	1,59303	0,64651	1,10685	0,12303	0,00000	0,00000	0,56572	0,89577
analítico	0,00000	0,58974	0,44237	0,00000	0,66569	3,10473	0,87390	0,04536	0,00000	0,04653	2,13350	0,04023	0,74644	0,54137	0,00000	0,00000	0,57687	1,40395
sabor	0,05516	7,45369	0,00000	0,00000	0,00000	0,70006	0,00000	0,00000	0,00000	0,00000	1,08748	0,07586	1,20183	0,00000	0,00000	0,00000	0,66088	2,18353
partícula	0,00000	0,72122	0,56319	0,01752	0,13260	1,72267	1,17807	0,35675	0,33605	0,00000	2,87571	0,10165	1,04910	0,00000	0,00000	0,00000	0,56591	1,25281
relatório	0,05411	0,20617	0,53794	0,00000	0,82690	0,18849	1,72983	0,28677	0,05795	0,02496	1,43400	0,15514	0,57918	2,88628	0,03941	0,00000	0,56294	1,09024
setor	0,04289	0,63688	0,81650	0,00000	1,21004	0,03293	2,47814	0,39277	0,06181	0,02246	1,58818	0,12726	0,59220	0,96762	0,02154	0,00000	0,56195	0,85981
computacional	0,00000	0,07615	0,79186	0,02263	1,92881	0,00000	1,94398	0,58768	0,05017	0,02246	1,98832	0,08192	0,16812	1,14797	0,04285	0,00990	0,55393	1,17184
móvel	0,00000	0,00000	0,51217	0,03663	0,99442	0,00000	1,40455	0,14943	0,17347	0,23933	2,18643	0,00000	0,45410	2,39649	0,25995	0,00000	0,55044	1,07910
fragrância	0,00000	0,06193	0,00000	0,00000	0,00000	0,10484	0,00000	0,00000	0,00000	0,00000	7,97454	0,00000	0,66226	0,00000	0,00000	0,00000	0,55022	2,35388
tecido	0,04046	0,15610	3,07432	0,00000	0,00000	0,31582	0,53234	0,05721	0,14356	0,00000	3,11336	0,12704	1,08775	0,12259	0,00000	0,09428	0,55405	1,28015
fato	0,02372	1,04192	0,86290	0,02089	0,53058	0,37731	1,28428	1,07508	0,04634	0,00000	2,25994	0,09660	0,56886	0,59363	0,05522	0,00000	0,55233	1,17015
sólido	0,00000	1,46209	0,70091	0,01263	0,43968	1,06912	0,57673	0,46943	0,24728	0,03854	1,65720	0,03379	2,32764	0,06886	0,03757	0,00000	0,57134	1,01071
viscosidade	0,01678	2,13134	0,52958	0,01872	0,15811	0,38344	0,27151	0,56683	0,07076	0,00000	2,15020	0,13039	3,12834	0,00000	0,00000	0,00000	0,59725	1,27311
pele	0,01638	0,10390	0,21865	0,00000	0,00000	0,47333	0,03345	0,00000	0,04261	0,00000	5,72575	0,07708	2,01788	0,00000	0,00000	0,00939	0,54490	1,98727
bibliográfico	0,00000	0,96511	1,20609	0,06440	0,56126	0,88015	1,00067	0,77193	0,11448	0,02407	2,09455	0,27060	1,15919	0,32971	0,00000	0,00000	0,59014	1,43181
laboratorial	0,08873	1,70759	0,49490	0,04767	0,58426	0,47576	1,08759	0,74880	0,24622	0,11032	2,52909	0,31670	1,24682	0,31168	0,05575	0,01046	0,62890	1,35906
benefício	0,02862	1,29224	0,31830	0,05240	0,26178	0,87895	0,88672	0,18404	0,16329	0,02496	2,98813	0,06336	1,17954	0,43776	0,00000	0,00000	0,54751	0,96620
resistente	0,27477	0,70156	0,88162	0,05201	0,30952	0,37775	1,94520	0,37514	0,11919	0,19632	2,09321	0,21620	1,07817	0,00000	0,00000	0,02219	0,54018	1,06274
acompanhamento	0,04757	0,61797	1,02301	0,01802	1,01711	0,69519	0,96987	0,97212	0,00000	0,05183	1,60836	0,08337	0,79665	1,11456	0,01808	0,00000	0,56461	1,01923
solda	0,00000	0,16168	1,33380	0,05661	0,88427	0,00000	3,20697	1,09106	0,00000	0,35377	1,10169	0,03379	0,34633	0,02982	0,01576	0,00000	0,53847	1,02876
emissão	0,06743	0,17736	0,47294	0,02336	0,80944	0,16203	3,26490	0,16946	0,15686	0,04478	1,94036	0,04446	0,54448	0,73233	0,01863	0,00000	0,53930	0,96693
próprio	0,00000	0,41950	0,88716	0,00000	0,78797	0,28091	1,53720	0,40989	0,05844	0,12459	1,51687	0,15708	0,59977	1,90713	0,07357	0,00000	0,54750	0,85300
documento	0,00000	0,20010	0,76053	0,10683	1,79407	0,03763	0,95615	0,59756	0,00539	0,00000	1,70223	0,00000	0,17001	2,37577	0,00000	0,00000	0,54414	1,05137
complexo	0,03692	0,69028	0,70148	0,00000	0,75783	0,28687	1,53623	0,23968	0,07067	0,08300	2,27399	0,07681	0,79198	1,11997	0,07201	0,00000	0,54611	1,06822
esforço	0,02150	0,32501	1,10005	0,15777	0,41512	0,22079	2,98778	0,48754	0,02580	0,02592	1,52527	0,00000	0,56517	0,71327	0,01593	0,00939	0,53727	0,99803
conexão	0,00000	0,06082	1,05897	0,01597	1,22548	0,00000	2,55669	0,37078	0,00000	0,02246	1,63427	0,00000	0,22743	1,25290	0,12170	0,00000	0,53422	1,04150
tanque	0,02991	0,52730	0,85991	0,03319	0,10783	0,16742	3,69506	0,50733	0,14013	0,02246	1,03912	0,13747	1,15794	0,21773	0,00000	0,00000	0,54018	1,05585
digital	0,02150	0,05746	0,71975	0,00000	1,34088	0,01437	2,26744	0,04347	0,03171	0,00000	1,36950	0,00000	0,15752	2,27500	0,24666	0,01046	0,53473	1,08971
cabo	0,00000	0,03191	1,32388	0,09553	1,95729	0,00000	2,11803	1,23514	0,12037	0,01604	0,78425	0,04722	0,46681	0,12113	0,14486	0,00000	0,52890	1,08282
cartão	0,00000	0,08580	0,00000	0,05380	2,86388	0,00000	0,24076	0,08694	0,00000	0,00000	2,63731	0,00000	0,00000	2,42578	0,03780	0,00000	0,52700	1,30042
precisão	0,02023	0,45608	1,14603	0,01810	1,07321	0,15557	2,32963	0,62227	0,04553	0,08773	1,61360	0,14999	0,34178	0,43763	0,01983	0,00964	0,53293	0,93935
anterior	0,00000	0,52040	0,69405	0,05899	0,69531	0,35264	1,93808	0,64753	0,11165	0,02930	1,50761	0,23708	0,46395	1,19571	0,05113	0,02089	0,53277	0,77840

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
filtro	0,00000	0,23116	0,86966	0,09834	0,58383	0,24375	2,15887	0,33895	0,00000	0,00000	2,48503	0,33695	0,53233	0,56059	0,00000	0,00000	0,52747	1,03678
doença	0,31881	0,61341	0,00000	0,00000	0,00000	2,15163	0,26757	0,03478	0,00000	0,00000	3,16435	0,47919	1,31416	0,04247	0,00000	0,00000	0,52415	1,45649
negócio	0,03439	0,40977	0,47855	0,00000	0,53728	0,00000	0,30361	0,07745	0,00591	0,00000	2,71020	0,12924	0,09336	3,53505	0,09653	0,00000	0,52571	1,20184
corrosão	0,00000	0,05825	1,03562	0,04861	0,40033	0,00000	2,15728	1,34953	0,19866	0,00000	1,47808	0,09021	1,50030	0,03941	0,00000	0,00000	0,52227	0,97919
tolerância	0,94377	0,26784	0,96653	0,01597	0,35051	0,06410	2,13431	1,73024	0,00000	0,00000	1,19633	0,04828	0,27694	0,30017	0,01576	0,00000	0,51942	1,05415
inclusão	0,00000	1,78242	0,47985	0,01752	0,59048	0,39178	1,22991	0,90848	0,02092	0,03456	2,80838	0,04324	0,09470	0,50990	0,08981	0,00000	0,56262	0,98263
rendimento	0,20182	1,40344	0,86396	0,00000	0,16887	0,67931	0,86599	0,79611	0,16213	0,00000	1,66167	0,17510	1,70085	0,06951	0,00000	0,02073	0,54809	0,85623
aderência	0,00000	0,51253	0,62994	0,01802	0,12908	0,35457	1,16456	0,33445	0,00000	0,00000	1,44112	0,11166	2,75749	1,00574	0,00000	0,02156	0,53005	1,01990
adesivo	0,00000	0,02712	1,20478	0,00000	0,17161	0,21057	0,42489	0,66200	0,00000	0,05664	1,07739	0,34036	4,08624	0,00000	0,00000	0,00000	0,51635	1,28858
paciente	0,00000	0,00000	0,07996	0,00000	0,23500	4,79065	0,00000	0,00000	0,00000	0,00000	2,27047	0,00000	0,30107	0,57952	0,00000	0,00000	0,51604	1,82408
resposta	0,02023	0,56815	0,70476	0,01429	0,40263	1,55447	0,94279	0,26744	0,08873	0,00000	1,49498	0,10181	0,93202	1,29722	0,04845	0,00000	0,52737	0,92809
confiabilidade	0,00000	0,22429	0,89532	0,03130	1,64842	0,28571	2,46125	0,23624	0,02212	0,00000	1,08689	0,10681	0,40572	0,79954	0,01921	0,00000	0,51393	0,90891
segmento	0,02866	0,39756	0,67820	0,15174	0,55942	0,45388	1,29566	0,59408	0,03343	0,49431	1,44887	0,14953	0,90159	1,04401	0,02459	0,00000	0,51597	0,79366
compatível	0,02372	0,28742	0,42712	0,03003	0,64997	0,46508	1,88784	0,09800	0,04749	0,00000	1,66764	0,10556	1,36018	1,12577	0,01183	0,00000	0,51173	0,94935
liberação	0,00000	0,57152	0,97010	0,00000	0,29344	2,25379	1,46634	0,22559	0,02580	0,00000	1,64342	0,09901	0,53599	0,28095	0,00000	0,00000	0,52287	1,15761
regra	0,00000	0,11055	1,40665	0,00000	0,49361	0,03099	0,06144	0,05039	0,00000	0,00000	2,64146	0,00000	0,16323	3,17529	0,04198	0,00000	0,51098	1,20549
convencional	0,04392	0,51559	0,89061	0,14464	0,69607	0,22275	1,33012	0,80230	0,18654	0,09763	1,57840	0,16855	1,19903	0,36969	0,00000	0,00000	0,51537	0,74601
opção	0,00000	0,42085	0,58325	0,06321	0,66107	0,48925	1,11210	0,15866	0,00000	0,07141	2,60039	0,13548	1,01957	1,02196	0,03302	0,00000	0,52314	0,89682
acabamento	0,00000	0,18700	1,54211	0,01971	0,16019	0,00000	1,97752	0,77849	0,02580	0,32619	0,81767	0,21332	2,03351	0,00000	0,00000	0,05148	0,50831	0,98268
versão	0,00000	0,44312	0,56812	0,00000	1,00098	0,00000	1,26777	0,06648	0,00000	0,00000	1,87595	0,00000	0,24327	2,62455	0,12141	0,00000	0,51323	0,98104
flexível	0,00000	0,17233	0,98810	0,01810	0,43126	0,48041	0,94812	0,11373	0,02497	0,09715	2,67966	0,03379	0,90118	1,19737	0,02365	0,00000	0,50686	1,07028
alimentação	0,02219	1,32857	1,64128	0,00000	1,02902	0,08891	1,88427	0,81444	0,14567	0,02324	0,83998	0,00000	0,14288	0,27641	0,14967	0,00000	0,52416	0,92150
tubo	0,00000	0,20225	0,64099	0,03826	0,07418	0,00000	2,48286	1,75015	0,02977	0,14624	1,88667	0,02816	0,81877	0,00000	0,00000	0,00000	0,50614	1,03812
escória	0,00000	0,00000	0,22013	0,00000	0,00000	0,00000	0,05732	1,17510	0,04926	0,00000	6,56061	0,00000	0,00000	0,00000	0,00000	0,00000	0,50390	2,59066
extração	0,01965	0,98212	0,46690	0,04111	0,20616	0,08151	0,84450	0,55879	0,10824	0,03503	3,03393	0,31214	0,51263	1,32347	0,03194	0,00000	0,53488	1,07722
similar	0,02457	0,96341	0,94448	0,06137	0,37451	0,55710	1,37081	0,56658	0,09752	0,05912	1,58041	0,18427	1,15480	0,25636	0,00000	0,00000	0,51221	0,74467
eixo	0,00000	0,07031	1,28738	0,03607	0,19654	0,08106	4,03506	0,72046	0,00000	0,05615	1,48991	0,00000	0,00000	0,07239	0,00000	0,00000	0,50283	1,13742
prática	0,00000	0,68716	0,75160	0,16819	0,63106	0,38232	0,39248	0,35403	0,00000	0,00000	2,65495	0,35360	0,26149	1,51674	0,04087	0,01402	0,51303	0,96994
troca	0,02150	0,54134	0,96269	0,00000	0,50572	0,10820	2,43241	0,34351	0,08416	0,00000	1,34556	0,13381	0,45441	1,17577	0,08575	0,00000	0,51218	0,83607
corpo	0,00000	0,50192	1,14100	0,10813	0,52024	0,02702	1,42127	0,61924	0,11659	0,10821	2,35159	0,00000	1,11040	0,07253	0,01883	0,00939	0,50790	0,98311
visual	0,00000	0,72073	0,96145	0,03459	0,54652	0,08189	2,38826	0,38629	0,06711	0,01872	1,33393	0,00000	0,40421	1,26873	0,00000	0,03035	0,51517	1,05968
certificação	0,00000	0,15598	0,74945	0,00000	1,01601	0,17947	3,63049	0,17282	0,00000	0,00000	1,03505	0,03379	0,08437	0,94286	0,03590	0,00000	0,50226	1,30703
ferramental	0,00000	0,05389	1,47893	0,00000	0,12253	0,09580	3,70404	1,40272	0,00000	0,16758	0,74219	0,04693	0,00000	0,19662	0,00000	0,00000	0,50070	1,20655
interferência	0,04525	0,45428	0,84051	0,11965	0,98732	0,29241	2,42113	0,17217	0,00000	0,00000	1,27423	0,02347	0,61136	0,65831	0,08880	0,00000	0,49930	0,86540
ii	0,00000	0,16275	0,12613	0,01752	1,53516	1,10119	1,26725	0,22031	0,04380	0,00000	2,26687	0,00000	0,26903	1,03357	0,01430	0,00000	0,50362	0,97864
contínuo	0,01965	0,84146	1,10663	0,00000	0,68789	0,21856	1,07475	1,43418	0,05110	0,07049	1,11550	0,09641	0,61550	0,76217	0,02196	0,00000	0,50727	0,77451

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
operador	0,00000	0,17485	1,34452	0,05720	0,62796	0,14159	3,31992	0,76007	0,00000	0,00000	0,67858	0,01816	0,25101	0,61143	0,00000	0,00000	0,49908	1,02544
liga	0,00000	0,00000	0,62206	0,00000	0,76793	0,00000	2,45244	3,10453	0,00000	0,00000	0,70720	0,00000	0,16774	0,06227	0,00000	0,00000	0,49276	1,29814
literatura	0,00000	0,78116	0,59893	0,03668	0,67749	0,84860	0,76518	0,63203	0,16942	0,00000	2,05107	0,14875	0,83740	0,59094	0,06509	0,00000	0,51267	0,90475
umidade	0,06735	2,03457	0,78934	0,09038	0,38767	0,76935	0,80277	0,14045	0,14910	0,09742	1,63694	0,34611	1,25163	0,00000	0,00000	0,00000	0,53519	0,92112
célula	0,00000	0,29095	0,59593	0,15875	0,60322	0,91692	2,58589	0,25454	0,02977	0,00000	1,44022	0,04970	0,81960	0,09244	0,12293	0,00000	0,49755	0,96748
distinto	0,01911	0,72070	0,48661	0,00000	0,68621	0,38650	0,91694	0,15897	0,08536	0,01872	1,60812	0,25494	0,98333	1,61028	0,10298	0,02711	0,50412	0,83250
plano	0,00000	0,20340	0,90226	0,01358	0,95235	0,06798	1,59866	0,86131	0,03991	0,02042	1,41559	0,19017	0,52423	1,09931	0,04994	0,00000	0,49620	0,91054
confecção	0,07455	0,33228	1,22766	0,02969	0,52989	0,08042	1,98146	0,87438	0,02669	0,22087	1,28900	0,02607	1,20231	0,11423	0,02619	0,00990	0,50285	0,97457
calor	0,04986	0,73970	1,26210	0,07691	0,74986	0,36381	2,07285	0,33716	0,00000	0,01821	1,41556	0,04149	0,69107	0,18744	0,03854	0,00000	0,50278	0,85312
rotina	0,06847	0,06619	0,84462	0,00000	0,61936	0,15729	2,05609	0,07167	0,00000	0,00000	1,46550	0,00000	0,13877	2,34553	0,01593	0,00000	0,49059	0,91166
responsável	0,02219	1,16259	0,60509	0,01752	0,71749	0,26036	1,25768	0,14714	0,01935	0,03850	1,46888	0,10824	0,73920	1,24026	0,07347	0,00000	0,49237	0,73931
homologação	0,00000	0,09958	0,27107	0,00000	0,64420	0,05692	1,70734	0,34063	0,00654	0,04813	1,82437	0,04023	0,59006	2,06636	0,14531	0,00000	0,49005	1,20847
densidade	0,02219	1,33108	0,80210	0,05810	0,65212	0,64997	1,11317	0,23449	0,31564	0,08358	1,09286	0,25372	1,55880	0,03152	0,00000	0,00000	0,51246	0,73248
fábrica	0,04299	1,29838	1,28477	0,01358	0,69995	0,03903	2,73622	0,17923	0,00000	0,01925	0,85098	0,29713	0,73850	0,21962	0,00000	0,00000	0,52623	0,89935
período	0,09226	1,62555	0,51545	0,02674	0,60834	1,78681	0,79947	0,11439	0,02497	0,00000	0,97238	0,19677	1,17945	0,64125	0,02196	0,01110	0,53856	0,94825
frequência	0,07974	0,31554	1,46816	0,05336	1,57287	0,11932	2,43736	0,20855	0,03366	0,00000	0,94791	0,02414	0,03436	0,32525	0,13448	0,00000	0,48467	0,93807
limite	0,02457	0,51794	0,95271	0,13111	0,80989	0,19722	1,95573	0,75949	0,00000	0,00000	1,29317	0,09751	0,48679	0,53238	0,00000	0,00000	0,48491	0,73003
espessura	0,00000	0,54716	1,36905	0,12536	0,16779	0,03399	2,41319	0,95515	0,08566	0,19088	0,86130	0,13968	0,99027	0,03448	0,00000	0,00893	0,49518	0,82857
papel	0,00000	0,65779	0,68984	0,00000	0,34399	0,10073	0,28461	0,00000	0,00000	0,00000	1,82519	2,23198	1,16232	0,53213	0,01757	0,00000	0,49038	0,92229
essencial	0,01859	1,21068	0,23623	0,02523	0,35115	0,26568	0,32930	0,00000	0,06230	0,00000	4,61094	0,08197	0,26773	0,41040	0,00000	0,00000	0,49189	1,81616
prova	0,00000	0,09604	0,66097	0,09554	0,42631	0,07667	1,62898	0,38873	0,02277	0,00000	1,45401	0,04294	1,09073	1,62175	0,09235	0,02093	0,48242	1,02944
ácido	0,00000	2,29453	0,54048	0,00000	0,09027	0,92035	0,38631	0,15482	0,22337	0,00000	1,63734	0,11689	1,91426	0,00000	0,00000	0,00000	0,51741	1,03994
sinal	0,00000	0,03874	0,79896	0,00000	1,98148	0,19062	1,40480	0,10639	0,02765	0,00000	1,39586	0,04806	0,32891	1,04324	0,28268	0,00000	0,47796	0,83440
armazenamento	0,00000	0,97054	0,56780	0,01468	0,70150	0,35276	1,07709	0,20896	0,00000	0,02246	1,73425	0,04722	0,87585	1,43874	0,03498	0,00000	0,50293	0,76677
frio	0,02646	0,59803	0,14537	0,00000	0,06134	0,03513	1,59536	3,89603	0,22503	0,08022	0,40405	0,15655	0,54203	0,00000	0,00000	0,00000	0,48535	1,40296
transferência	0,01859	0,40304	0,82796	0,03393	0,83938	0,30287	1,00616	0,32946	0,12858	0,00000	1,76009	0,10966	1,05055	0,86086	0,03416	0,00000	0,48158	0,71693
combustível	0,00000	0,09707	0,26723	0,00000	0,25084	0,03586	4,56524	0,43605	0,15608	0,00000	1,11764	0,14868	0,34631	0,16763	0,00000	0,00000	0,47429	1,23974
hora	0,00000	0,66120	1,10474	0,00000	0,24557	0,45245	1,58751	0,41800	0,02867	0,00000	1,36789	0,07943	1,04011	0,70710	0,02365	0,00000	0,48227	0,77301
força	0,00000	0,25846	1,25300	0,09282	0,27501	0,06234	3,05170	0,47594	0,01935	0,01821	1,27293	0,03997	0,54004	0,14233	0,03617	0,00000	0,47114	0,97404
aprovação	0,00000	0,57997	1,21026	0,02593	0,22350	0,93101	1,15238	0,52707	0,00000	0,07966	1,51268	0,08932	0,83660	0,78831	0,00000	0,00990	0,49791	1,00182
falta	0,01810	0,62486	0,99521	0,09253	1,02998	0,28553	1,67092	0,21408	0,03932	0,00000	1,29015	0,17197	0,34209	0,64374	0,00000	0,01144	0,46437	0,89125
código	0,00000	0,03806	0,43983	0,00000	0,97281	0,20001	0,78461	0,11799	0,00000	0,00000	1,93328	0,00000	0,03927	2,72029	0,17896	0,00000	0,46407	1,00119
estável	0,03824	1,01016	0,34438	0,00000	0,19158	1,38307	0,83153	0,23378	0,15903	0,00000	1,06532	0,07836	1,77352	0,43785	0,00000	0,00000	0,47168	0,99261
taxa	0,00000	0,54836	0,46271	0,00000	0,64560	1,11028	0,60960	0,62606	0,22187	0,00000	1,83310	0,12458	0,73351	0,53032	0,05132	0,00000	0,46858	0,80398
pigmento	0,00000	0,09637	0,40079	0,00000	0,00000	0,02702	0,17377	0,00000	0,05160	0,00000	5,41770	0,17592	1,03333	0,00000	0,00000	0,00000	0,46103	2,10980
entrada	0,02751	0,43381	1,24515	0,08805	1,20982	0,16984	1,34580	0,19957	0,03970	0,00000	1,32219	0,09315	0,30671	0,94810	0,05206	0,00990	0,46821	0,74097

(Continues...)

Table 64 – ... Continuation

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
estado	0,12544	0,25938	0,77443	0,00000	1,27486	0,38413	0,74824	0,19917	0,06888	0,00000	1,68782	0,08479	0,51258	1,11005	0,11013	0,00000	0,45874	0,71180
questão	0,01965	0,75288	0,50935	0,08432	0,61656	0,19170	1,32428	0,22369	0,05946	0,09894	1,25798	0,08995	1,37843	0,78480	0,01972	0,00000	0,46323	0,77165
enzima	0,00000	3,23105	0,03842	0,00000	0,00000	0,22923	0,02993	0,00000	0,00000	0,00000	1,39491	0,40256	2,62361	0,00000	0,00000	0,00000	0,49686	1,34018
revisão	0,00000	0,24255	0,65401	0,00000	0,93990	0,42022	1,29255	0,69422	0,08450	0,00000	1,79676	0,13619	0,65978	0,60221	0,03860	0,00000	0,47259	0,89163
porta	0,08081	0,07439	0,74390	0,15633	0,52699	0,04811	3,83372	0,11503	0,00000	0,85240	0,22694	0,00000	0,07776	0,51071	0,03626	0,00000	0,45521	1,06077
separação	0,07428	0,51805	0,48787	0,04917	0,10749	0,15304	2,25502	0,27843	0,21915	0,00000	1,74468	0,03129	1,10762	0,37808	0,03740	0,00000	0,46510	0,95897
total	0,02646	1,28286	0,83926	0,07901	0,64907	0,31484	1,32905	0,39088	0,02669	0,00000	1,47981	0,06824	0,57377	0,59838	0,01119	0,00000	0,47934	0,69081
3d	0,00000	0,00000	1,08103	0,04059	0,57698	0,12507	2,46938	0,30350	0,04547	0,00000	1,85178	0,00000	0,66675	0,10287	0,00000	0,01402	0,45484	1,36841
limitação	0,00000	0,88216	0,92297	0,02866	0,59477	0,28040	1,34772	0,63572	0,00000	0,04998	1,55380	0,09267	0,51006	0,93564	0,05846	0,00000	0,49331	0,86099
animal	0,00000	5,23574	0,09240	0,00000	0,03611	1,17323	0,68455	0,04968	0,04691	0,00000	1,10853	0,00000	0,30599	0,00000	0,00000	0,00000	0,54582	1,60293
laminação	0,00000	0,36975	0,32024	0,00000	0,10833	0,00000	0,07360	4,40295	0,05999	0,00000	1,31104	0,05638	0,58413	0,00000	0,00000	0,00000	0,45540	1,51745
cilindro	0,16287	0,04567	0,11629	0,08345	0,00000	0,00000	4,05619	1,36623	0,10321	0,00000	0,82345	0,12548	0,24676	0,00000	0,00000	0,01953	0,44682	1,10748
canal	0,00000	0,09589	0,68586	0,07049	0,91558	0,13231	0,95690	0,85997	0,05442	0,00000	1,92034	0,00000	0,00000	1,43200	0,09952	0,00000	0,45145	0,78934
forno	0,04586	0,82108	0,89253	0,00000	0,08552	0,00000	0,86988	2,13252	0,21465	0,02106	1,86088	0,05844	0,36253	0,00000	0,00000	0,00000	0,46031	0,90281
mapeamento	0,02372	0,64876	0,64305	0,03271	0,85786	0,05645	1,10695	0,22036	0,06095	0,04492	2,02911	0,07207	0,70460	1,06502	0,07588	0,00000	0,47765	0,82407
movimentação	0,00000	0,21861	1,05338	0,09072	0,44995	0,07415	3,14793	0,39280	0,00000	0,04765	0,96069	0,06746	0,16650	0,49979	0,00000	0,00000	0,44810	0,88991
lógico	0,00000	0,09850	0,69721	0,01468	0,95149	0,01744	1,28668	1,05857	0,05742	0,03789	1,44909	0,00000	0,18544	1,22415	0,11721	0,00000	0,44974	0,76650
posterior	0,04231	0,55509	0,80248	0,04470	0,34548	0,40551	1,66504	1,21199	0,07762	0,09266	1,19049	0,16968	0,56087	0,29468	0,03456	0,00852	0,46886	0,80262
layout	0,00000	0,33833	0,84742	0,03522	0,66164	0,02927	2,74764	0,22150	0,00000	0,33426	0,78677	0,04970	0,22201	0,94248	0,02049	0,00000	0,45230	0,92816
solvente	0,00000	0,11709	0,47422	0,00000	0,00000	1,36216	0,39043	0,35752	0,28966	0,00000	1,28661	0,02414	2,86298	0,00000	0,00000	0,00000	0,44780	1,02516
dinâmico	0,00000	0,13425	0,89655	0,10747	0,59521	0,00000	1,73132	0,08129	0,12073	0,02407	1,56450	0,29464	0,19542	1,42064	0,00000	0,00000	0,44788	0,81533
cenário	0,04927	0,28697	0,26872	0,11915	1,00217	0,18559	0,70688	0,29976	0,04427	0,02246	1,54861	0,15226	0,65118	1,84722	0,09634	0,00000	0,45505	0,83599
mineral	0,00000	2,83232	0,21440	0,00000	0,37847	0,67260	0,26831	0,80064	0,25686	0,00000	1,24601	0,18310	0,92802	0,03343	0,00000	0,00000	0,48839	0,94365
estatístico	0,07605	0,72776	0,40461	0,01207	0,71238	0,33914	0,98294	1,15787	0,02867	0,00000	1,78910	0,02485	0,54216	0,60613	0,02793	0,00000	0,46448	1,02315
bateria	0,00000	0,00000	2,57866	0,00000	1,34591	0,00000	1,08713	0,18867	0,13269	0,00000	1,11929	0,00000	0,03141	0,51584	0,04103	0,00000	0,44004	1,05863
cadeia	0,00000	1,09319	0,22337	0,00000	0,27953	0,15363	0,50756	0,06158	0,00000	0,00000	3,29462	0,00000	1,22096	0,41041	0,00000	0,00000	0,45280	1,34456
gráfico	0,00000	0,04716	0,49865	0,04022	1,16122	0,11554	0,63143	0,06375	0,00000	0,02407	2,14247	0,08448	0,14256	2,07050	0,03513	0,00000	0,44107	0,89280
vegetal	0,14985	2,07967	0,09721	0,00000	0,39775	0,17583	0,07623	0,15230	0,06557	0,00000	3,05458	0,13909	1,02460	0,00000	0,00000	0,00000	0,46329	1,33589
emulsão	0,00000	0,29259	0,05859	0,00000	0,00000	0,18346	0,08714	0,41773	0,14519	0,04669	3,54929	0,00000	2,24511	0,00000	0,00000	0,00000	0,43911	1,21013
compressão	0,00000	0,29926	0,54985	0,14467	0,40891	1,64060	0,67650	0,03068	0,23466	0,04228	0,82003	0,10237	1,62110	0,50443	0,05058	0,00000	0,44537	1,06169
ph	0,02293	1,79442	0,30320	0,00000	0,46234	0,84243	0,12192	0,45342	0,07473	0,00000	1,81771	0,27873	1,62830	0,09422	0,00000	0,00000	0,49340	0,91172
complexidade	0,00000	0,16959	0,49558	0,05722	0,73924	0,15624	0,88404	0,13668	0,12459	0,00000	2,01098	0,06587	0,44864	1,65711	0,02855	0,00000	0,43590	1,03151
filme	0,00000	0,59545	0,41453	0,00000	0,16870	0,24986	0,32011	0,25916	0,14713	0,06618	2,34740	0,33058	2,18476	0,00000	0,00000	0,00000	0,44274	0,94927
sintético	0,01720	0,21945	0,24919	0,01697	0,31618	0,35751	0,22522	0,14398	0,11981	0,00000	4,77980	0,04828	0,45690	0,00000	0,00000	0,01308	0,43522	2,06404
dosagem	0,04586	1,99719	0,33303	0,01429	0,06655	0,81810	1,10735	0,16993	0,13548	0,00000	1,14773	0,28979	1,36031	0,11876	0,00000	0,00000	0,47527	0,78016
padronização	0,00000	0,94595	1,36930	0,04004	0,53144	0,11292	0,85661	0,34004	0,03686	0,01821	1,37664	0,11929	0,28703	1,06181	0,04931	0,00000	0,44659	0,74325

(Continues...)

Table 64 – Conclusion

Entities	AGR	FOD	CSG	CON	ELE	PHA	MEC	MET	MIN	FUR	OTH	PAP	PET	TEL	TIC	TXT	AVG	SDV
parcela	0,14948	0,05023	0,03183	0,00000	0,03186	0,00000	0,08200	0,00000	0,00000	0,00000	6,34624	0,00000	0,02893	0,13002	0,00000	0,00000	0,42816	3,68620
gase	0,16923	0,36780	0,71090	0,02089	1,02025	0,02289	1,82255	0,51448	0,19116	0,00000	1,05697	0,00000	1,14257	0,00000	0,00000	0,00000	0,43998	0,73225
conteúdo	0,00000	0,38386	0,20282	0,00000	0,57873	0,40699	0,45860	0,11375	0,02092	0,02407	2,56505	0,00000	0,15147	1,96248	0,20202	0,00000	0,44192	0,96919
pintura	0,00000	0,00000	1,39778	0,01502	0,07342	0,00000	2,62682	0,55527	0,00000	0,19912	0,26040	0,00000	1,79215	0,00000	0,00000	0,00000	0,43250	0,94899
vibração	0,00000	0,00000	1,60417	0,18582	0,79965	0,00000	3,47171	0,06429	0,02419	0,00000	0,56568	0,00000	0,14077	0,00000	0,05281	0,00000	0,43182	1,13480
superficial	0,00000	0,17120	0,64724	0,00000	0,17787	0,17568	1,03183	2,36221	0,09833	0,00000	0,88258	0,44130	0,92065	0,00000	0,00000	0,00000	0,43181	0,88933
espaço	0,01965	0,52415	0,97339	0,03850	0,88244	0,00000	2,06158	0,06527	0,03366	0,16219	1,37746	0,20981	0,18162	0,46910	0,00000	0,00000	0,43743	0,75775
válvula	0,02751	0,17797	0,72367	0,00000	0,05035	0,05546	3,13446	0,73549	0,05557	0,00000	1,29647	0,07240	0,62455	0,00000	0,00000	0,00000	0,43462	1,08017
fácil	0,00000	0,59656	0,58614	0,00000	0,68358	0,34387	1,11967	0,23681	0,04300	0,00000	1,63008	0,17468	0,76015	0,67735	0,00000	0,00000	0,42824	0,77645

Table 64 – Source: Produced by the author in August, 2022

Apendix H – Results on both 2014 and 2015 data

Table 65 – Results of first experiment on 2014 and 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,3886485	58,00%	0,7147621	55,00%	n/a	n/a
1	0,6538010	53,50%	0,6868654	48,50%	n/a	n/a
2	0,7660685	56,50%	0,7295969	61,00%	n/a	n/a
3	0,4492395	63,50%	0,7307356	62,00%	n/a	n/a
4	0,5250385	70,50%	0,7535233	66,00%	n/a	n/a
5	0,8700364	64,50%	0,7034421	63,50%	n/a	n/a
6	1,0286618	61,50%	0,5358914	69,00%	n/a	n/a
7	0,7724543	63,50%	0,6649615	66,50%	n/a	n/a
8	0,6712085	63,50%	0,5738838	65,50%	n/a	n/a
9	0,7031614	61,00%	0,7322342	60,50%	n/a	n/a
10	0,8662608	64,00%	0,6998888	58,00%	n/a	n/a
11	0,9785979	58,50%	0,6916351	62,00%	n/a	n/a
12	0,4080084	61,50%	0,8091433	62,00%	n/a	n/a
13	0,6575530	63,50%	0,4091890	64,00%	n/a	n/a
14	0,6529043	68,50%	0,5200529	64,50%	n/a	n/a
15	0,3381422	64,50%	0,8529295	66,50%	n/a	n/a
16	0,7877655	58,50%	0,6224316	66,00%	n/a	n/a
17	0,9755859	62,50%	0,7877079	67,00%	n/a	n/a
18	0,9876770	61,50%	0,7161641	62,50%	n/a	n/a
19	0,9028913	58,00%	0,7718223	64,50%	n/a	n/a
Avg	0,7191852	61,85%	0,6853430	62,73%	0,4542553	65,93%

Source: Produced by the author in August, 2022

Table 66 – Results of second experiment on 2014 and 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5023926	69,00%	0,7853353	63,50%	n/a	n/a
1	0,8726247	60,50%	0,6165007	63,00%	n/a	n/a
2	0,5063334	59,00%	0,6207035	57,50%	n/a	n/a
3	0,2613795	65,50%	0,9280383	61,00%	n/a	n/a

(Continues...)

Table 66 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
4	0,5762354	67,00%	0,7180801	55,00%	n/a	n/a
5	0,7562460	57,50%	0,6136654	68,00%	n/a	n/a
6	0,5631293	59,50%	0,5823141	64,00%	n/a	n/a
7	0,5010618	68,00%	0,6252527	57,00%	n/a	n/a
8	0,4419042	68,50%	0,6045283	57,00%	n/a	n/a
9	0,6602542	61,50%	0,7792217	45,50%	n/a	n/a
10	0,4476378	60,50%	0,7557794	53,50%	n/a	n/a
11	0,3314600	69,50%	0,8671335	59,00%	n/a	n/a
12	0,5900602	58,00%	0,7570684	44,00%	n/a	n/a
13	0,3188139	62,00%	0,6401137	47,00%	n/a	n/a
14	0,5665224	69,50%	0,8280228	43,00%	n/a	n/a
15	0,8134527	60,00%	0,7083869	52,00%	n/a	n/a
16	0,3843022	63,50%	0,7911255	48,00%	n/a	n/a
17	0,5722710	68,00%	0,7274208	37,00%	n/a	n/a
18	0,5379304	59,50%	0,7059563	59,50%	n/a	n/a
19	0,5022351	56,00%	0,6669690	43,00%	n/a	n/a
Avg	0,5353123	63,13%	0,7160808	53,88%	0,7725949	42,59%

Source: Produced by the author in August, 2022

Table 67 – Results of third experiment on 2014 and 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5890881	66,00%	0,5680518	64,50%	n/a	n/a
1	0,9942089	63,50%	0,5412942	63,50%	n/a	n/a
2	0,5165772	57,50%	0,6783755	53,00%	n/a	n/a
3	0,4905117	62,50%	0,7037501	47,50%	n/a	n/a
4	0,5057909	61,00%	0,6441801	46,00%	n/a	n/a
5	0,7785524	63,00%	0,7364213	48,00%	n/a	n/a
6	0,9561334	67,50%	0,5900215	63,00%	n/a	n/a
7	0,4935753	61,50%	0,7275422	46,00%	n/a	n/a
8	0,4540006	63,00%	0,7246974	62,50%	n/a	n/a
9	1,0124822	69,50%	0,5692288	61,50%	n/a	n/a
10	0,5151863	63,00%	0,7138008	53,50%	n/a	n/a
11	0,7902510	63,50%	0,5912863	51,50%	n/a	n/a
12	0,6343316	65,00%	0,6469668	66,00%	n/a	n/a
13	0,4455733	63,50%	0,8144121	49,50%	n/a	n/a
14	0,5892111	64,00%	0,7105876	48,00%	n/a	n/a

(Continues...)

Table 67 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
15	0,8184833	50,50%	0,7031222	56,00%	n/a	n/a
16	0,3949355	65,00%	0,7050660	48,50%	n/a	n/a
17	0,4910039	67,50%	0,7927505	34,00%	n/a	n/a
18	0,3193203	63,50%	0,6160908	63,50%	n/a	n/a
19	0,4919238	67,00%	0,5709489	65,00%	n/a	n/a
Avg	0,6140570	63,38%	0,6674297	54,55%	0,5456628	63,91%

Source: Produced by the author in August, 2022

Table 68 – Results of fourth experiment on 2014 and 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6807530	64,50%	0,4698048	59,50%	n/a	n/a
1	0,3528185	67,00%	0,7316462	60,50%	n/a	n/a
2	0,5337496	66,50%	0,6878811	60,00%	n/a	n/a
3	0,6147767	62,50%	0,6904351	48,50%	n/a	n/a
4	0,4272808	62,00%	0,8303249	59,50%	n/a	n/a
5	0,7287048	61,50%	0,6816573	58,50%	n/a	n/a
6	0,9085168	66,00%	0,7872552	65,50%	n/a	n/a
7	0,6561899	73,00%	0,5785746	64,50%	n/a	n/a
8	0,6814299	59,50%	0,6399240	57,00%	n/a	n/a
9	0,8957055	61,50%	0,7147590	59,00%	n/a	n/a
10	1,1061902	55,50%	0,6037158	62,50%	n/a	n/a
11	0,7611868	57,50%	0,7142638	60,00%	n/a	n/a
12	0,7814903	54,00%	0,7173154	56,00%	n/a	n/a
13	0,8247769	59,50%	0,7179898	47,00%	n/a	n/a
14	0,9718803	67,00%	0,5788212	61,50%	n/a	n/a
15	0,6777171	54,00%	0,6663795	53,50%	n/a	n/a
16	0,5701951	63,50%	0,7122833	63,00%	n/a	n/a
17	0,7096373	56,00%	0,6501163	59,50%	n/a	n/a
18	0,8123898	65,00%	0,7019785	54,50%	n/a	n/a
19	0,5052955	64,50%	0,7461426	58,00%	n/a	n/a
Avg	0,7100342	62,03%	0,6810634	58,40%	0,6563075	57,66%

Source: Produced by the author in August, 2022

Table 69 – Results of fifth experiment on 2014 and 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5258713	70,00%	0,6797404	37,00%	n/a	n/a
1	0,7697705	61,00%	0,8153795	48,50%	n/a	n/a
2	1,1279777	65,50%	0,7385701	59,50%	n/a	n/a
3	0,5253782	66,00%	0,4668018	62,50%	n/a	n/a
4	0,5027798	62,00%	0,9429209	60,00%	n/a	n/a
5	0,6006927	67,00%	0,7564625	62,00%	n/a	n/a
6	0,6648994	69,50%	0,7773873	56,50%	n/a	n/a
7	0,5427002	59,00%	0,6442705	62,50%	n/a	n/a
8	0,6439949	61,50%	0,6228808	65,00%	n/a	n/a
9	0,3903089	61,00%	0,6760572	66,50%	n/a	n/a
10	0,4219078	73,50%	0,5645092	64,50%	n/a	n/a
11	0,4395999	66,00%	0,6387715	60,00%	n/a	n/a
12	0,7981240	70,00%	0,6963936	62,50%	n/a	n/a
13	0,5990337	46,00%	0,7600437	67,50%	n/a	n/a
14	0,4334487	71,50%	0,5443265	61,00%	n/a	n/a
15	0,4271961	66,50%	0,7043959	54,00%	n/a	n/a
16	0,6358114	64,50%	0,6467639	56,50%	n/a	n/a
17	0,5984063	67,00%	0,6841676	69,50%	n/a	n/a
18	0,6788583	54,00%	0,6533815	62,50%	n/a	n/a
19	0,6101220	56,50%	0,7118124	57,50%	n/a	n/a
Avg	0,5968441	63,90%	0,6862518	59,78%	0,6960490	59,04%

Source: Produced by the author in August, 2022

Table 70 – Results of sixth experiment on 2014 and 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5867051	56,00%	0,6224812	49,50%	n/a	n/a
1	0,5136875	58,50%	0,6159082	59,00%	n/a	n/a
2	0,4756929	65,00%	0,7458933	45,00%	n/a	n/a
3	0,6280274	61,00%	0,6647198	56,00%	n/a	n/a
4	0,7166486	59,50%	0,6310710	57,50%	n/a	n/a
5	0,8486899	63,00%	0,6740686	65,00%	n/a	n/a
6	0,7295178	63,50%	0,6055002	53,50%	n/a	n/a
7	0,4172276	60,50%	0,6408778	66,50%	n/a	n/a
8	0,6410120	59,00%	0,5232891	58,00%	n/a	n/a
9	0,4725471	64,50%	0,8323055	64,50%	n/a	n/a

(Continues...)

Table 70 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
10	0,9335210	58,50%	0,5852129	68,50%	n/a	n/a
11	0,7404841	61,50%	0,4636629	69,50%	n/a	n/a
12	0,9527619	53,00%	0,8461193	64,00%	n/a	n/a
13	0,5300083	62,50%	0,3675491	66,00%	n/a	n/a
14	0,8424282	62,00%	0,5792518	69,00%	n/a	n/a
15	0,5389841	68,00%	0,4638059	69,00%	n/a	n/a
16	0,4748057	66,50%	0,4342717	63,50%	n/a	n/a
17	0,8179801	69,00%	0,8544983	58,50%	n/a	n/a
18	0,6575661	71,50%	0,7139076	61,00%	n/a	n/a
19	0,4831816	63,00%	0,7938212	59,00%	n/a	n/a
Avg	0,6500738	62,30%	0,6329108	61,13%	0,6772864	57,74%

Source: Produced by the author in August, 2022

Table 71 – Results of seventh experiment on 2014 and 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7413872	60,50%	0,6889288	61,00%	n/a	n/a
1	0,8935277	59,00%	0,6604004	71,00%	n/a	n/a
2	0,8243706	71,50%	0,4816147	68,50%	n/a	n/a
3	0,8387600	59,50%	0,6072698	58,50%	n/a	n/a
4	0,4567389	67,50%	0,6878220	63,50%	n/a	n/a
5	0,3608966	59,50%	0,5587195	70,50%	n/a	n/a
6	0,7202919	50,50%	0,7445267	70,00%	n/a	n/a
7	0,5422295	58,00%	0,6689665	65,00%	n/a	n/a
8	0,9403460	65,00%	0,7715655	68,50%	n/a	n/a
9	0,5540283	74,00%	0,5979690	66,50%	n/a	n/a
10	0,9078514	75,50%	0,9029780	61,50%	n/a	n/a
11	0,4103739	62,50%	0,6921678	68,50%	n/a	n/a
12	0,3434760	69,00%	0,6461065	63,00%	n/a	n/a
13	0,7098682	59,50%	0,6373383	48,00%	n/a	n/a
14	0,6046124	62,00%	0,5828266	64,00%	n/a	n/a
15	0,3304689	65,00%	0,6025137	64,00%	n/a	n/a
16	0,3934766	58,50%	0,5574169	62,00%	n/a	n/a
17	0,5744703	72,50%	0,5413923	66,00%	n/a	n/a
18	1,0547897	59,50%	0,5932189	71,50%	n/a	n/a
19	0,5512179	63,50%	0,7198685	66,50%	n/a	n/a
Avg	0,6376591	63,63%	0,6471805	64,90%	0,8221304	66,39%

(Continues...)

Table 71 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc

Source: Produced by the author in August, 2022

Table 72 – Results of eighth experiment on 2014 and 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,9180663	67,00%	0,6769658	58,50%	n/a	n/a
1	0,6071287	66,50%	0,6500354	66,00%	n/a	n/a
2	0,8530509	67,50%	0,6292945	63,50%	n/a	n/a
3	0,6924880	59,50%	0,5707852	68,50%	n/a	n/a
4	0,7816710	70,50%	0,5977179	61,00%	n/a	n/a
5	0,5114037	68,00%	0,6404921	60,00%	n/a	n/a
6	1,2783288	58,50%	0,6855656	65,00%	n/a	n/a
7	0,6760320	66,00%	0,6571416	52,00%	n/a	n/a
8	0,6600112	66,00%	0,7284747	54,00%	n/a	n/a
9	0,3276981	60,50%	0,7311759	63,00%	n/a	n/a
10	0,5348225	69,50%	0,7321590	64,50%	n/a	n/a
11	0,7263323	63,50%	0,7141557	55,50%	n/a	n/a
12	0,3485471	73,00%	0,7105244	69,00%	n/a	n/a
13	0,4752225	67,00%	0,7913493	50,50%	n/a	n/a
14	0,4042814	64,50%	0,7600546	54,50%	n/a	n/a
15	0,8602138	67,50%	0,6515435	59,00%	n/a	n/a
16	0,4285547	57,50%	0,7576728	41,00%	n/a	n/a
17	0,9036069	64,50%	0,7974396	41,50%	n/a	n/a
18	0,4631870	66,50%	0,6524177	48,00%	n/a	n/a
19	0,8986227	62,50%	0,8368509	34,00%	n/a	n/a
Avg	0,6674635	65,30%	0,6985908	56,45%	0,6876231	36,42%

Source: Produced by the author in August, 2022

Table 73 – Results of ninth experiment on 2014 and 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	1,1386532	69,00%	0,5943304	68,50%	n/a	n/a
1	0,6362228	61,00%	0,5644813	68,00%	n/a	n/a
2	0,7335097	60,00%	0,7092119	62,00%	n/a	n/a
3	0,5681852	60,00%	0,7462019	68,50%	n/a	n/a
4	1,0290625	66,00%	0,9423324	63,50%	n/a	n/a
5	0,3762296	62,50%	0,7681576	67,00%	n/a	n/a

(Continues...)

Table 73 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	0,7083027	64,50%	0,6236790	59,50%	n/a	n/a
7	0,3697030	59,00%	0,6098647	67,50%	n/a	n/a
8	0,6347813	61,00%	0,7909496	61,50%	n/a	n/a
9	1,0327806	64,50%	0,6358469	66,00%	n/a	n/a
10	0,7846305	62,50%	0,6051008	65,00%	n/a	n/a
11	0,6521821	61,00%	0,6340288	62,00%	n/a	n/a
12	0,6003810	64,50%	0,7153733	52,00%	n/a	n/a
13	0,7727878	55,50%	0,5144073	64,50%	n/a	n/a
14	0,8613833	60,00%	0,8027006	64,00%	n/a	n/a
15	0,6484346	69,00%	0,5784854	67,50%	n/a	n/a
16	0,6179071	61,00%	0,5819254	67,50%	n/a	n/a
17	0,4818470	67,50%	0,7245796	66,00%	n/a	n/a
18	0,9272049	63,50%	0,7486579	64,00%	n/a	n/a
19	0,4309922	66,00%	0,6107261	66,50%	n/a	n/a
Avg	0,6674635	65,30%	0,6985908	56,45%	0,4020247	66,51%

Source: Produced by the author in August, 2022

Table 74 – Results of tenth experiment on 2014 and 2015 data

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6760248	60,50%	0,5231963	59,50%	n/a	n/a
1	0,6218987	64,50%	0,7732202	64,50%	n/a	n/a
2	0,6241614	63,50%	0,6960313	70,50%	n/a	n/a
3	0,7356898	67,00%	0,5374780	69,00%	n/a	n/a
4	0,6508605	63,00%	0,3728794	67,50%	n/a	n/a
5	0,6776663	64,50%	0,6417560	68,00%	n/a	n/a
6	0,6637805	61,50%	0,7543549	62,00%	n/a	n/a
7	0,5165954	63,00%	0,8446635	58,00%	n/a	n/a
8	0,5566117	62,50%	0,5067893	58,50%	n/a	n/a
9	0,8142537	60,50%	0,8324468	65,50%	n/a	n/a
10	0,8240569	64,00%	0,6593827	63,00%	n/a	n/a
11	0,4837590	57,00%	0,6860142	49,00%	n/a	n/a
12	0,6609454	68,00%	0,5118853	64,00%	n/a	n/a
13	0,6680374	56,00%	0,5620289	62,50%	n/a	n/a
14	0,8594114	59,50%	0,7430278	50,00%	n/a	n/a
15	1,1857014	69,00%	0,6854539	62,00%	n/a	n/a
16	0,4116877	65,50%	0,4606022	69,00%	n/a	n/a

(Continues...)

Table 74 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
17	0,5838583	61,00%	0,6395733	58,50%	n/a	n/a
18	0,6100718	62,50%	0,8108097	63,50%	n/a	n/a
19	0,4785272	68,50%	0,7897341	67,00%	n/a	n/a
Avg	0,6651800	63,08%	0,6515664	62,58%	0,4284774	65,97%

Source: Produced by the author in August, 2022

Apendix I – Results on potential domain 1 - 2015

Table 75 – Results of first experiment on potential domain 1 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8199777	73,00%	0,3168526	75,50%	n/a	n/a
1	0,5416527	76,00%	0,9641059	75,50%	n/a	n/a
2	0,5913037	79,50%	0,2800530	77,00%	n/a	n/a
3	0,6411119	76,00%	0,4471789	75,00%	n/a	n/a
4	0,3428064	77,00%	0,4127478	74,50%	n/a	n/a
5	0,2607795	79,00%	0,6215457	73,50%	n/a	n/a
6	0,5504788	80,50%	0,3591959	77,00%	n/a	n/a
7	0,2661320	75,00%	0,5211365	78,00%	n/a	n/a
8	0,3728493	78,00%	0,3353113	78,00%	n/a	n/a
9	0,3005441	74,00%	0,7287017	77,00%	n/a	n/a
10	0,9107645	75,50%	0,4514198	82,00%	n/a	n/a
11	0,3823148	77,50%	0,8840349	81,50%	n/a	n/a
12	0,3070083	79,00%	0,5595661	83,00%	n/a	n/a
13	0,4844763	79,00%	0,5808970	81,00%	n/a	n/a
14	0,9801958	78,50%	0,3835474	80,50%	n/a	n/a
15	0,9301703	81,50%	0,3287211	74,00%	n/a	n/a
16	0,0466352	83,00%	0,2702201	79,00%	n/a	n/a
17	0,1671710	83,00%	0,8080448	83,50%	n/a	n/a
18	0,7384597	81,50%	0,5623266	87,00%	n/a	n/a
19	0,6242334	79,50%	0,5711486	82,00%	n/a	n/a
Avg	0,5129533	78,30%	0,5193378	78,73%	0,3949083	84,91%

Source: Produced by the author in August, 2022

Table 76 – Results of second experiment on potential domain 1 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5853352	77,00%	0,3933160	78,50%	n/a	n/a
1	0,2111130	80,00%	0,3576816	72,00%	n/a	n/a
2	0,4175219	76,00%	0,4308411	78,50%	n/a	n/a
3	0,6857882	77,00%	0,3559482	77,50%	n/a	n/a

(Continues...)

Table 76 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
4	0,3103989	78,00%	0,6108391	79,50%	n/a	n/a
5	1,0426135	75,00%	0,6559814	76,00%	n/a	n/a
6	0,6537135	76,50%	0,4828998	80,50%	n/a	n/a
7	0,7770845	75,00%	0,5506740	76,00%	n/a	n/a
8	0,2953367	79,50%	0,6406440	75,00%	n/a	n/a
9	0,2245806	76,50%	0,5876883	66,00%	n/a	n/a
10	0,2714569	76,50%	0,6972476	71,50%	n/a	n/a
11	0,3974725	74,50%	0,5704069	80,00%	n/a	n/a
12	0,1614884	76,50%	0,7922956	84,00%	n/a	n/a
13	1,4555329	75,50%	0,4530510	80,00%	n/a	n/a
14	1,3408703	74,50%	0,7285243	81,50%	n/a	n/a
15	0,1098467	80,00%	0,6330905	81,50%	n/a	n/a
16	0,7210557	80,00%	0,3788731	81,50%	n/a	n/a
17	0,4826977	82,50%	0,6050946	79,50%	n/a	n/a
18	0,4885938	81,50%	0,6862635	79,50%	n/a	n/a
19	0,8075916	77,00%	0,6816624	82,50%	n/a	n/a
Avg	0,5720046	77,45%	0,5646511	78,05%	0,2623984	86,19%

Source: Produced by the author in August, 2022

Table 77 – Results of third experiment on potential domain 1 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,1763802	80,50%	0,3263398	74,00%	n/a	n/a
1	0,5425753	83,00%	0,8745543	75,50%	n/a	n/a
2	0,5580440	76,50%	0,7387735	76,00%	n/a	n/a
3	0,4635026	78,50%	0,5375480	77,00%	n/a	n/a
4	0,5887928	75,00%	0,2994072	78,50%	n/a	n/a
5	0,4484881	75,50%	0,3790283	75,50%	n/a	n/a
6	0,1395949	80,50%	0,6022638	76,00%	n/a	n/a
7	0,5624880	75,00%	0,4745786	81,00%	n/a	n/a
8	0,7201631	80,50%	0,3693533	75,50%	n/a	n/a
9	0,6064133	77,50%	0,6493967	75,50%	n/a	n/a
10	0,9040435	82,50%	0,5174701	75,00%	n/a	n/a
11	0,3887300	77,50%	0,8282153	82,00%	n/a	n/a
12	0,6482493	79,50%	0,6040556	77,00%	n/a	n/a
13	0,2731319	79,00%	0,5323682	80,00%	n/a	n/a
14	0,4752365	73,50%	0,8519825	82,00%	n/a	n/a

(Continues...)

Table 77 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
15	1,4474427	79,00%	0,4006872	81,50%	n/a	n/a
16	0,1541768	79,50%	0,4556671	82,50%	n/a	n/a
17	0,1082153	83,00%	0,6604697	84,00%	n/a	n/a
18	0,6171630	82,50%	0,6700835	82,50%	n/a	n/a
19	0,7436093	79,50%	0,2329261	86,50%	n/a	n/a
Avg	0,5283220	78,90%	0,5502584	78,88%	0,7948045	87,21%

Source: Produced by the author in August, 2022

Table 78 – Results of fourth experiment on potential domain 1 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6104446	74,50%	0,6530904	66,00%	n/a	n/a
1	0,3188853	77,50%	0,8032354	66,00%	n/a	n/a
2	0,6627033	83,00%	0,5781333	68,50%	n/a	n/a
3	0,6567844	78,50%	0,5692551	75,00%	n/a	n/a
4	0,2797355	77,50%	0,3366516	71,00%	n/a	n/a
5	1,1164914	76,00%	0,5543593	73,00%	n/a	n/a
6	0,5291679	76,50%	0,5963016	85,00%	n/a	n/a
7	0,2503699	78,00%	0,5972137	81,50%	n/a	n/a
8	0,2026295	79,00%	0,4149151	83,00%	n/a	n/a
9	0,4932453	78,00%	0,4631083	80,50%	n/a	n/a
10	0,3298969	78,00%	0,5480866	78,50%	n/a	n/a
11	0,7050511	80,50%	0,3827543	84,50%	n/a	n/a
12	0,4525181	78,50%	0,7686241	82,50%	n/a	n/a
13	0,6414027	74,50%	0,5317204	76,00%	n/a	n/a
14	0,3070091	78,50%	0,8685150	81,50%	n/a	n/a
15	0,3703456	81,50%	0,5038768	79,50%	n/a	n/a
16	0,6146868	78,00%	0,5828634	84,50%	n/a	n/a
17	0,4495285	83,00%	0,5540592	78,50%	n/a	n/a
18	0,6594931	82,00%	0,4955517	86,00%	n/a	n/a
19	0,2969307	78,00%	1,0221930	85,00%	n/a	n/a
Avg	0,4973660	78,55%	0,5912254	78,30%	0,4924087	82,10%

Source: Produced by the author in August, 2022

Table 79 – Results of fifth experiment on potential domain 1 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6683879	79,00%	0,5508174	73,00%	n/a	n/a
1	0,4770113	75,50%	0,5653525	68,50%	n/a	n/a
2	0,4767040	82,00%	0,5068421	72,00%	n/a	n/a
3	1,4508156	73,00%	0,4892642	76,50%	n/a	n/a
4	0,4403095	81,00%	0,2876003	81,00%	n/a	n/a
5	1,0483434	76,50%	0,5599005	87,00%	n/a	n/a
6	0,5951599	75,50%	0,4573874	86,50%	n/a	n/a
7	0,6496015	75,50%	0,4653589	82,00%	n/a	n/a
8	0,4621367	75,50%	0,3336649	85,00%	n/a	n/a
9	0,4210221	77,50%	0,3010532	86,50%	n/a	n/a
10	0,4435630	75,50%	0,5670246	84,00%	n/a	n/a
11	1,2652205	82,50%	0,3368395	88,00%	n/a	n/a
12	0,4009787	78,00%	0,3795156	86,50%	n/a	n/a
13	0,4955718	81,00%	0,5014768	84,50%	n/a	n/a
14	0,5307540	76,00%	0,7511122	88,50%	n/a	n/a
15	0,3319145	78,00%	0,4637946	82,50%	n/a	n/a
16	0,6519338	80,50%	0,4644141	88,00%	n/a	n/a
17	0,1082769	82,00%	0,4344290	86,50%	n/a	n/a
18	0,2394074	81,00%	0,4488633	86,50%	n/a	n/a
19	0,5407987	77,50%	0,3954280	83,00%	n/a	n/a
Avg	0,5848956	78,15%	0,4630070	82,80%	0,7062281	83,63%

Source: Produced by the author in August, 2022

Table 80 – Results of sixth experiment on potential domain 1 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7705991	79,00%	0,5515971	86,50%	n/a	n/a
1	0,4384261	69,50%	0,5318093	83,00%	n/a	n/a
2	0,2387757	81,00%	0,3043123	84,50%	n/a	n/a
3	0,1575683	79,50%	0,5818593	84,00%	n/a	n/a
4	0,2084031	75,50%	0,3504326	82,50%	n/a	n/a
5	0,7421640	70,00%	0,4300731	80,50%	n/a	n/a
6	0,8088793	83,00%	0,3975415	82,50%	n/a	n/a
7	0,4474835	81,00%	0,2913163	84,00%	n/a	n/a
8	0,7512953	82,00%	0,2504302	79,50%	n/a	n/a
9	0,6535088	80,50%	0,3370661	90,50%	n/a	n/a

(Continues...)

Table 80 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
10	0,1386832	83,00%	0,4372816	81,00%	n/a	n/a
11	0,7371647	77,00%	0,5775032	89,50%	n/a	n/a
12	0,4431576	78,50%	0,7504016	81,00%	n/a	n/a
13	0,9187346	78,00%	0,6944723	83,50%	n/a	n/a
14	0,3892705	72,00%	0,4492281	90,50%	n/a	n/a
15	0,2846331	78,00%	0,6568633	86,50%	n/a	n/a
16	0,4576106	80,50%	0,4446944	89,00%	n/a	n/a
17	0,2863814	82,50%	0,4803891	87,50%	n/a	n/a
18	0,7389690	76,50%	0,3936227	79,50%	n/a	n/a
19	0,1680444	80,00%	0,5748310	88,50%	n/a	n/a
Avg	0,4889876	78,35%	0,4742863	84,70%	0,3578029	86,70%

Source: Produced by the author in August, 2022

Table 81 – Results of seventh experiment on potential domain 1 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,2535911	81,00%	0,6238632	74,00%	n/a	n/a
1	0,5648895	78,50%	0,5759618	76,00%	n/a	n/a
2	0,9512125	76,00%	0,3711626	77,50%	n/a	n/a
3	0,3644921	80,00%	0,3483541	77,00%	n/a	n/a
4	0,6596744	77,50%	0,3830013	80,00%	n/a	n/a
5	1,0588542	79,00%	0,4862386	74,50%	n/a	n/a
6	0,4456039	76,00%	0,6507686	80,50%	n/a	n/a
7	0,2804878	75,50%	0,4988849	83,50%	n/a	n/a
8	0,1508148	84,50%	0,6068197	83,00%	n/a	n/a
9	0,3560824	80,50%	0,6554457	86,00%	n/a	n/a
10	0,1658981	79,00%	0,2940848	86,50%	n/a	n/a
11	0,3547119	81,50%	0,6298612	86,00%	n/a	n/a
12	0,3143421	84,50%	0,5572523	74,00%	n/a	n/a
13	0,6952415	77,50%	0,6269660	83,50%	n/a	n/a
14	0,3365456	76,50%	0,6966959	83,00%	n/a	n/a
15	0,9878556	79,50%	0,6884910	87,00%	n/a	n/a
16	0,2220482	81,50%	0,4934215	86,00%	n/a	n/a
17	0,8264437	78,00%	0,3710755	84,00%	n/a	n/a
18	0,6026205	74,00%	0,6554133	79,50%	n/a	n/a
19	0,6756522	83,50%	0,6273665	86,50%	n/a	n/a
Avg	0,5133531	79,20%	0,5420564	81,40%	0,7402041	85,42%

(Continues...)

Table 81 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc

Source: Produced by the author in August, 2022

Table 82 – Results of eighth experiment on potential domain 1 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7080650	77,50%	0,6184716	73,00%	n/a	n/a
1	0,3034371	76,50%	0,5499589	74,50%	n/a	n/a
2	0,9050089	76,00%	0,8871542	77,00%	n/a	n/a
3	0,7829065	80,00%	0,5838605	74,50%	n/a	n/a
4	0,2816626	80,00%	0,6243340	73,50%	n/a	n/a
5	0,2142958	77,00%	0,6847352	79,50%	n/a	n/a
6	0,3287497	83,00%	0,4600350	78,50%	n/a	n/a
7	0,1361899	82,00%	0,2915443	76,50%	n/a	n/a
8	0,2134208	83,00%	0,4565560	78,00%	n/a	n/a
9	0,3904493	77,50%	0,4384800	86,00%	n/a	n/a
10	0,8110632	75,50%	0,3792374	83,50%	n/a	n/a
11	0,7351106	75,50%	0,4226587	83,50%	n/a	n/a
12	0,8488501	78,00%	0,3619498	83,50%	n/a	n/a
13	0,1050927	80,00%	0,2229401	84,00%	n/a	n/a
14	0,5434564	81,50%	0,3722688	75,50%	n/a	n/a
15	0,7323875	75,50%	0,4036800	82,00%	n/a	n/a
16	1,0274260	78,00%	0,4164886	83,50%	n/a	n/a
17	0,6213709	78,00%	0,3809306	86,00%	n/a	n/a
18	0,2393298	81,50%	0,3490925	87,50%	n/a	n/a
19	1,0561665	80,00%	0,8508452	86,00%	n/a	n/a
Avg	0,5492220	78,80%	0,4877611	80,30%	0,5696760	83,89%

Source: Produced by the author in August, 2022

Table 83 – Results of ninth experiment on potential domain 1 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5683272	73,50%	0,6163160	66,50%	n/a	n/a
1	0,5688030	76,00%	0,7450548	79,50%	n/a	n/a
2	0,1266320	78,00%	0,5745785	73,00%	n/a	n/a
3	0,2850207	74,50%	0,7463244	78,00%	n/a	n/a
4	0,6115211	77,00%	0,5345367	81,00%	n/a	n/a
5	0,1711582	79,00%	0,4446458	83,00%	n/a	n/a

(Continues...)

Table 83 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	0,2914962	78,50%	0,4489172	84,00%	n/a	n/a
7	0,2639974	77,50%	0,6198720	72,00%	n/a	n/a
8	0,5768045	80,00%	0,7331513	83,50%	n/a	n/a
9	0,1830900	80,00%	0,5898223	80,00%	n/a	n/a
10	1,3785729	85,00%	0,3536700	84,50%	n/a	n/a
11	0,2679202	78,00%	0,4474438	78,00%	n/a	n/a
12	0,7859906	77,00%	0,4309726	77,50%	n/a	n/a
13	0,3438930	81,00%	0,5634649	81,00%	n/a	n/a
14	0,5579405	75,50%	0,6674870	75,50%	n/a	n/a
15	1,0942779	82,00%	0,3065542	82,50%	n/a	n/a
16	0,1331689	83,50%	0,5672101	86,50%	n/a	n/a
17	0,3768359	75,00%	0,6825789	70,00%	n/a	n/a
18	1,1176577	79,00%	0,4071103	79,00%	n/a	n/a
19	0,6197544	82,50%	0,5080231	82,50%	n/a	n/a
Avg	0,5161431	78,63%	0,5493867	78,88%	0,4614289	82,86%

Source: Produced by the author in August, 2022

Table 84 – Results of tenth experiment on potential domain 1 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7271868	72,00%	0,6080760	79,00%	n/a	n/a
1	0,9353194	81,00%	0,5790914	75,50%	n/a	n/a
2	0,4267026	74,50%	0,5759230	76,00%	n/a	n/a
3	0,2405485	79,50%	0,5589648	65,00%	n/a	n/a
4	0,3487877	78,00%	0,6714593	79,00%	n/a	n/a
5	0,1804238	79,00%	0,5745173	76,00%	n/a	n/a
6	0,4918509	78,50%	0,4778139	76,00%	n/a	n/a
7	0,4794498	77,00%	0,7712458	84,00%	n/a	n/a
8	0,1481102	80,50%	0,4824307	72,00%	n/a	n/a
9	0,9652416	75,00%	0,5032945	85,50%	n/a	n/a
10	0,6968186	80,50%	0,4159815	85,00%	n/a	n/a
11	0,5807704	74,00%	0,5779290	85,00%	n/a	n/a
12	1,0379007	81,00%	0,5222728	81,50%	n/a	n/a
13	0,2280377	74,50%	0,5605372	85,50%	n/a	n/a
14	0,8452133	72,00%	0,5547661	74,00%	n/a	n/a
15	0,5043268	81,50%	0,4613282	85,50%	n/a	n/a
16	0,2376101	79,50%	0,6340859	76,50%	n/a	n/a

(Continues...)

Table 84 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
17	0,1882068	78,00%	0,6066647	82,00%	n/a	n/a
18	0,7925748	83,50%	0,5270073	86,00%	n/a	n/a
19	0,6151815	79,50%	0,2262157	88,50%	n/a	n/a
Avg	0,5335131	77,95%	0,5444802	79,88%	0,1665477	85,93%

Source: Produced by the author in August, 2022

Apendix J – Results on potential domain 2 - 2015

Table 85 – Results of first experiment on potential domain 2 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,2490470	76,00%	0,6497121	71,50%	n/a	n/a
1	0,9058078	75,00%	0,5865284	75,00%	n/a	n/a
2	0,8829389	77,00%	0,7207784	70,50%	n/a	n/a
3	0,2411964	71,00%	0,6006169	74,50%	n/a	n/a
4	0,6071767	82,00%	0,6369670	71,50%	n/a	n/a
5	0,6526534	76,50%	0,7676795	69,50%	n/a	n/a
6	0,9170226	73,50%	0,5541548	64,00%	n/a	n/a
7	0,4974580	75,50%	0,6438200	69,50%	n/a	n/a
8	0,1822949	80,00%	0,3347548	69,00%	n/a	n/a
9	0,5775334	77,50%	0,4331423	68,00%	n/a	n/a
10	0,8770348	78,50%	0,7279912	71,00%	n/a	n/a
11	0,7132462	74,00%	0,7403221	69,50%	n/a	n/a
12	0,2981110	81,00%	0,5725345	80,00%	n/a	n/a
13	0,2874769	79,50%	0,4888239	74,50%	n/a	n/a
14	0,7717968	77,50%	0,5425438	72,50%	n/a	n/a
15	0,5324281	74,00%	0,6215786	76,00%	n/a	n/a
16	0,6603186	75,00%	0,6021544	74,00%	n/a	n/a
17	0,7273824	74,00%	0,3970176	69,00%	n/a	n/a
18	0,4731808	73,50%	0,5667424	76,50%	n/a	n/a
19	0,3857290	73,00%	0,5826178	74,00%	n/a	n/a
Avg	0,5719917	76,20%	0,5885240	72,00%	0,4979948	75,08%

Source: Produced by the author in August, 2022

Table 86 – Results of second experiment on potential domain 2 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7823533	71,50%	0,6035264	84,00%	n/a	n/a
1	0,5401382	71,00%	0,5294876	80,00%	n/a	n/a
2	0,2018586	76,50%	0,6524087	79,00%	n/a	n/a
3	0,5478313	73,00%	0,4062400	79,50%	n/a	n/a

(Continues...)

Table 86 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
4	0,6433356	65,50%	0,7007202	75,00%	n/a	n/a
5	0,6640248	73,50%	0,3810688	75,00%	n/a	n/a
6	0,2125962	81,00%	0,3624294	73,50%	n/a	n/a
7	0,4669626	73,50%	0,5876093	72,50%	n/a	n/a
8	0,3332689	71,00%	0,6391124	73,50%	n/a	n/a
9	0,5099131	78,00%	0,4016327	74,00%	n/a	n/a
10	0,3120450	79,00%	0,6937255	76,50%	n/a	n/a
11	0,6073948	79,00%	0,4711679	70,00%	n/a	n/a
12	0,4157759	70,00%	0,5678797	73,50%	n/a	n/a
13	0,5834280	78,00%	0,6840096	72,50%	n/a	n/a
14	0,5116104	71,00%	0,5029746	76,00%	n/a	n/a
15	0,1730121	77,00%	0,6517407	75,00%	n/a	n/a
16	1,7419109	74,50%	0,4490601	73,00%	n/a	n/a
17	0,5867844	76,00%	0,5415808	72,00%	n/a	n/a
18	0,7678156	68,50%	0,3154313	78,50%	n/a	n/a
19	1,3295567	75,00%	0,3653966	76,00%	n/a	n/a
Avg	0,5965808	74,13%	0,5253601	75,45%	0,3389537	75,89%

Source: Produced by the author in August, 2022

Table 87 – Results of third experiment on potential domain 2 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7057607	76,00%	0,2718427	73,50%	n/a	n/a
1	0,6223745	74,00%	0,5826848	79,50%	n/a	n/a
2	0,7837496	73,00%	0,6193760	73,50%	n/a	n/a
3	0,8562783	73,50%	0,5193216	74,00%	n/a	n/a
4	0,3899680	78,00%	0,5486818	76,00%	n/a	n/a
5	0,9225329	75,00%	0,6402481	69,50%	n/a	n/a
6	0,1898866	78,50%	0,4489905	71,00%	n/a	n/a
7	0,2340257	80,50%	0,3083217	73,00%	n/a	n/a
8	0,5782116	80,50%	0,4873400	79,50%	n/a	n/a
9	0,8825649	76,00%	0,5505648	76,00%	n/a	n/a
10	0,6141536	75,00%	0,6191831	79,00%	n/a	n/a
11	0,2946150	74,00%	0,6994041	72,50%	n/a	n/a
12	0,6629719	80,00%	0,6432911	67,50%	n/a	n/a
13	0,3270690	75,00%	0,6709231	45,50%	n/a	n/a
14	0,5587631	77,00%	0,6062455	72,00%	n/a	n/a

(Continues...)

Table 87 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
15	0,7871007	71,50%	0,5300069	71,50%	n/a	n/a
16	0,3905256	76,00%	0,6459518	58,00%	n/a	n/a
17	0,0973666	81,00%	0,4722306	77,00%	n/a	n/a
18	0,5375969	74,00%	0,6305036	54,50%	n/a	n/a
19	1,0996206	71,50%	0,6408999	62,50%	n/a	n/a
Avg	0,5767568	76,00%	0,5568006	70,28%	0,5497156	62,78%

Source: Produced by the author in August, 2022

Table 88 – Results of fourth experiment on potential domain 2 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,3199010	69,50%	0,3492614	74,50%	n/a	n/a
1	0,2607360	78,00%	0,5159157	68,00%	n/a	n/a
2	0,2956607	77,00%	0,9172875	68,00%	n/a	n/a
3	0,5309442	79,00%	0,7959324	68,00%	n/a	n/a
4	0,4765842	71,00%	0,5992159	69,00%	n/a	n/a
5	0,2319220	70,50%	0,8058347	71,50%	n/a	n/a
6	0,2783653	72,00%	0,5179716	70,50%	n/a	n/a
7	0,3517507	74,00%	0,7422640	62,00%	n/a	n/a
8	0,8765189	69,50%	0,6358881	70,50%	n/a	n/a
9	0,1291645	82,50%	0,6055065	77,00%	n/a	n/a
10	0,6835480	73,00%	0,5936790	81,00%	n/a	n/a
11	0,5084291	77,00%	0,6981407	74,00%	n/a	n/a
12	0,7134852	77,50%	0,4447134	75,50%	n/a	n/a
13	0,9187735	71,50%	0,5919939	79,00%	n/a	n/a
14	0,8068092	77,50%	0,5602769	72,50%	n/a	n/a
15	0,9032838	77,50%	0,7515982	76,50%	n/a	n/a
16	0,1375461	78,00%	0,3160040	77,50%	n/a	n/a
17	0,3393294	77,50%	0,7258954	72,00%	n/a	n/a
18	0,4460989	80,50%	0,5384840	65,00%	n/a	n/a
19	0,0785422	74,50%	0,4490106	78,50%	n/a	n/a
Avg	0,4643696	75,38%	0,6077437	72,53%	0,5307578	74,76%

Source: Produced by the author in August, 2022

Table 89 – Results of fifth experiment on potential domain 2 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,4157678	74,50%	0,6730016	72,50%	n/a	n/a
1	1,4517943	77,50%	0,7295310	67,50%	n/a	n/a
2	0,2499657	72,50%	0,5314268	64,50%	n/a	n/a
3	0,9596267	83,50%	0,3968375	66,50%	n/a	n/a
4	0,6754267	75,50%	0,8481641	68,50%	n/a	n/a
5	0,2788500	78,00%	0,5196747	71,00%	n/a	n/a
6	0,8306040	77,00%	0,5214580	67,50%	n/a	n/a
7	1,0197155	70,50%	0,5164819	69,50%	n/a	n/a
8	1,1043642	75,50%	0,5986707	66,00%	n/a	n/a
9	0,7486724	75,00%	0,5952889	81,00%	n/a	n/a
10	0,6733095	73,00%	0,6225911	67,00%	n/a	n/a
11	0,8296219	75,00%	0,5840060	80,50%	n/a	n/a
12	1,2503003	72,00%	0,6349765	70,00%	n/a	n/a
13	0,5659755	80,00%	0,5014881	80,50%	n/a	n/a
14	1,0039802	81,00%	0,7268619	75,50%	n/a	n/a
15	0,4224515	76,00%	0,5742214	73,50%	n/a	n/a
16	0,4261177	72,50%	0,4371202	78,00%	n/a	n/a
17	0,1726625	79,00%	0,4763921	77,50%	n/a	n/a
18	0,5770215	81,00%	0,5519537	75,50%	n/a	n/a
19	0,5303078	67,50%	0,5125235	80,50%	n/a	n/a
Avg	0,7093268	75,83%	0,5776335	72,65%	0,4770080	77,35%

Source: Produced by the author in August, 2022

Table 90 – Results of sixth experiment on potential domain 2 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8459007	74,00%	0,7585133	75,50%	n/a	n/a
1	0,4511338	80,00%	0,5835528	75,00%	n/a	n/a
2	0,8729857	81,50%	0,6007572	74,00%	n/a	n/a
3	0,3302414	76,50%	0,4101335	70,00%	n/a	n/a
4	0,6818296	78,50%	0,6537821	76,50%	n/a	n/a
5	0,2049921	79,50%	0,5532899	77,50%	n/a	n/a
6	0,4708014	74,50%	0,4433337	72,50%	n/a	n/a
7	0,4247434	70,50%	0,4109128	72,50%	n/a	n/a
8	0,4962483	73,50%	0,6018147	74,00%	n/a	n/a
9	0,6258656	80,50%	0,6201180	73,00%	n/a	n/a

(Continues...)

Table 90 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
10	0,6425980	65,00%	0,6685879	77,00%	n/a	n/a
11	0,6618103	78,50%	0,8137583	75,00%	n/a	n/a
12	0,3003533	78,50%	0,3050636	74,50%	n/a	n/a
13	0,4135949	77,50%	0,3432955	75,00%	n/a	n/a
14	0,8086311	80,50%	0,3830667	80,00%	n/a	n/a
15	0,4445796	74,50%	0,3981808	73,50%	n/a	n/a
16	0,2960609	78,00%	0,3086710	74,50%	n/a	n/a
17	0,5363495	79,00%	0,8229868	76,50%	n/a	n/a
18	0,4022141	79,00%	0,3564817	76,50%	n/a	n/a
19	0,7100136	81,00%	0,9018726	74,50%	n/a	n/a
Avg	0,5310474	77,03%	0,5469087	74,88%	0,8312402	75,73%

Source: Produced by the author in August, 2022

Table 91 – Results of seventh experiment on potential domain 2 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,9080858	76,00%	1,0000669	82,00%	n/a	n/a
1	0,2278024	73,50%	0,5730708	74,50%	n/a	n/a
2	0,4678563	75,50%	0,8373871	80,00%	n/a	n/a
3	1,0597805	80,00%	0,7191378	73,50%	n/a	n/a
4	0,3555703	77,00%	0,5893214	78,50%	n/a	n/a
5	0,2964956	75,50%	0,6768479	74,50%	n/a	n/a
6	0,2013373	81,00%	0,5831281	72,50%	n/a	n/a
7	0,5699792	73,50%	0,3863421	78,50%	n/a	n/a
8	0,1924929	75,50%	0,7722069	81,00%	n/a	n/a
9	0,4832015	78,50%	0,3498482	78,50%	n/a	n/a
10	1,5282562	70,00%	0,5761378	76,00%	n/a	n/a
11	0,1874279	77,50%	0,8330208	79,00%	n/a	n/a
12	0,6490517	77,00%	0,5737085	76,50%	n/a	n/a
13	0,2130282	77,50%	0,4612977	76,50%	n/a	n/a
14	0,3662222	78,00%	0,5867969	52,50%	n/a	n/a
15	0,4664530	77,50%	0,4767424	76,00%	n/a	n/a
16	0,4234533	74,50%	0,7140505	66,00%	n/a	n/a
17	0,5494888	74,50%	0,6201831	70,00%	n/a	n/a
18	0,9864983	73,00%	0,4901012	74,00%	n/a	n/a
19	0,6046780	75,00%	0,7095288	77,00%	n/a	n/a
Avg	0,5368580	76,03%	0,6264463	74,85%	0,3573535	76,54%

(Continues...)

Table 91 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc

Source: Produced by the author in August, 2022

Table 92 – Results of eighth experiment on potential domain 2 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6968209	78,50%	0,3827821	71,50%	n/a	n/a
1	0,2101471	76,00%	0,7848206	64,00%	n/a	n/a
2	0,3726834	77,00%	0,5811996	67,00%	n/a	n/a
3	0,6155925	78,00%	0,6153270	70,50%	n/a	n/a
4	0,7480676	77,50%	0,5293612	65,50%	n/a	n/a
5	0,6133580	70,00%	0,6629764	47,00%	n/a	n/a
6	0,4329292	73,50%	0,5769765	66,50%	n/a	n/a
7	0,8028238	75,50%	0,5738953	69,00%	n/a	n/a
8	0,2125562	79,50%	0,4121827	72,00%	n/a	n/a
9	0,1776174	81,00%	0,8269331	70,00%	n/a	n/a
10	0,8506328	71,00%	0,6065518	71,00%	n/a	n/a
11	0,7606480	74,00%	0,7449467	74,50%	n/a	n/a
12	0,5939295	78,00%	0,4848111	67,50%	n/a	n/a
13	0,1587031	75,00%	0,3657215	76,00%	n/a	n/a
14	0,2018861	79,50%	0,3454781	74,00%	n/a	n/a
15	0,8084063	77,50%	0,2668542	74,00%	n/a	n/a
16	0,4281929	76,00%	0,4405389	76,00%	n/a	n/a
17	0,4583176	77,50%	0,2521428	78,50%	n/a	n/a
18	0,5464256	75,50%	0,5817289	74,00%	n/a	n/a
19	0,4836733	70,00%	0,3992195	76,50%	n/a	n/a
Avg	0,5086706	76,03%	0,5217224	70,25%	0,6991765	78,96%

Source: Produced by the author in August, 2022

Table 93 – Results of ninth experiment on potential domain 2 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,2482004	77,00%	0,8330120	73,50%	n/a	n/a
1	0,6927750	69,50%	0,6939220	56,00%	n/a	n/a
2	0,2405665	76,50%	0,5182417	74,00%	n/a	n/a
3	0,6267299	74,50%	0,8138202	62,00%	n/a	n/a
4	0,2461465	74,00%	0,8703569	75,50%	n/a	n/a
5	0,2507803	78,00%	0,4045626	76,00%	n/a	n/a

(Continues...)

Table 93 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	0,2273157	73,50%	0,8104928	73,00%	n/a	n/a
7	0,3151953	73,50%	0,4895636	75,00%	n/a	n/a
8	0,5901096	77,50%	0,3800493	71,50%	n/a	n/a
9	0,5513719	79,00%	0,3778705	73,50%	n/a	n/a
10	0,4479524	75,50%	0,3814110	80,00%	n/a	n/a
11	0,2648231	77,50%	0,5863307	67,00%	n/a	n/a
12	0,1873676	76,00%	0,6207309	79,50%	n/a	n/a
13	0,5667937	74,50%	0,4940536	76,00%	n/a	n/a
14	0,4486380	70,00%	0,5440008	74,50%	n/a	n/a
15	0,1877635	75,00%	0,4555684	76,00%	n/a	n/a
16	0,3303723	68,00%	0,6905165	59,00%	n/a	n/a
17	0,5040488	80,00%	0,4734305	78,00%	n/a	n/a
18	0,6423542	72,00%	0,6518078	36,50%	n/a	n/a
19	0,3043811	78,50%	0,6888401	55,50%	n/a	n/a
Avg	0,3936843	75,00%	0,5889291	69,60%	0,6196129	53,40%

Source: Produced by the author in August, 2022

Table 94 – Results of tenth experiment on potential domain 2 - 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6934705	76,00%	0,3364614	76,50%	n/a	n/a
1	0,3337495	77,00%	0,5047840	72,00%	n/a	n/a
2	0,7809383	71,00%	0,3894104	80,00%	n/a	n/a
3	0,5053203	76,00%	0,3725819	77,00%	n/a	n/a
4	0,5523115	72,50%	0,8693510	72,00%	n/a	n/a
5	0,5685243	76,00%	0,7997905	77,00%	n/a	n/a
6	0,3246205	75,00%	0,5411588	78,50%	n/a	n/a
7	0,8751096	80,00%	0,7039015	78,50%	n/a	n/a
8	0,4924453	80,50%	0,5694008	73,00%	n/a	n/a
9	0,3238714	80,00%	0,5900863	77,50%	n/a	n/a
10	0,3809647	76,00%	0,6754603	66,50%	n/a	n/a
11	0,8440722	79,00%	0,5369726	75,50%	n/a	n/a
12	0,8051921	71,50%	0,6045310	66,50%	n/a	n/a
13	0,5423342	76,50%	0,5510389	73,00%	n/a	n/a
14	0,5588395	78,00%	0,4665126	74,50%	n/a	n/a
15	1,8071177	72,50%	0,6135926	74,00%	n/a	n/a
16	0,3021652	75,50%	0,4701617	82,00%	n/a	n/a

(Continues...)

Table 94 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
17	0,4444045	74,50%	0,6462632	82,00%	n/a	n/a
18	0,1930847	77,50%	0,8276774	74,00%	n/a	n/a
19	0,9884880	77,00%	0,4753910	76,50%	n/a	n/a
Avg	0,6158512	76,10%	0,5772264	75,33%	0,8656888	76,05%

Source: Produced by the author in August, 2022

Apendix K – Results on potential domain 3 - 2014

Table 95 – Results of first experiment on potential domain 3 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6599541	51,00%	0,6640775	58,00%	n/a	n/a
1	0,7919251	57,00%	0,6669334	51,50%	n/a	n/a
2	1,7077128	56,50%	0,5854132	59,50%	n/a	n/a
3	0,8517946	56,50%	0,8232580	53,00%	n/a	n/a
4	0,8105274	58,50%	0,7511179	53,50%	n/a	n/a
5	0,8801873	60,50%	0,6725137	55,50%	n/a	n/a
6	0,5545549	55,50%	0,6840222	49,50%	n/a	n/a
7	0,2880690	62,50%	0,7155960	57,00%	n/a	n/a
8	0,4077110	55,50%	0,9106533	54,50%	n/a	n/a
9	0,6898438	67,50%	0,6447201	57,00%	n/a	n/a
10	0,5974178	63,00%	0,5593089	60,00%	n/a	n/a
11	0,6299590	55,00%	0,5286500	61,50%	n/a	n/a
12	0,5902433	55,00%	0,7591962	61,50%	n/a	n/a
13	0,8579262	59,50%	0,7130194	56,50%	n/a	n/a
14	0,6007873	55,00%	1,0668586	61,00%	n/a	n/a
15	0,3489843	58,00%	0,7328958	61,50%	n/a	n/a
16	0,9450495	61,50%	0,6123711	58,00%	n/a	n/a
17	0,6777104	59,50%	0,7304890	57,50%	n/a	n/a
18	1,1468337	56,50%	0,5653983	62,00%	n/a	n/a
19	0,5190202	53,00%	0,6082015	59,50%	n/a	n/a
Avg	0,7278106	57,85%	0,6997347	57,40%	0,6521066	61,57%

Source: Produced by the author in August, 2022

Table 96 – Results of second experiment on potential domain 3 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7511173	51,00%	0,6480350	61,50%	n/a	n/a
1	1,0002863	56,50%	0,6423215	57,50%	n/a	n/a
2	0,8677337	52,50%	0,7104087	46,50%	n/a	n/a
3	0,7334155	48,00%	0,6468546	65,50%	n/a	n/a

(Continues...)

Table 96 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
4	0,7130638	53,00%	0,7086564	52,00%	n/a	n/a
5	0,7217673	54,50%	0,6368385	52,50%	n/a	n/a
6	0,9604399	57,00%	0,6215202	62,50%	n/a	n/a
7	0,6130942	52,00%	0,6804811	61,00%	n/a	n/a
8	0,7377093	54,50%	0,6203495	58,50%	n/a	n/a
9	0,7921255	52,00%	0,5718721	63,50%	n/a	n/a
10	0,6618134	57,00%	0,6239579	52,50%	n/a	n/a
11	0,7556732	57,50%	0,6500580	59,50%	n/a	n/a
12	0,5630742	55,50%	0,8792986	57,00%	n/a	n/a
13	0,6143670	58,50%	0,7564061	62,50%	n/a	n/a
14	0,3709512	54,50%	0,7644534	61,50%	n/a	n/a
15	0,8549833	55,50%	0,5534407	66,00%	n/a	n/a
16	0,6332690	53,50%	0,8598397	64,00%	n/a	n/a
17	0,7643284	56,50%	0,7161865	62,50%	n/a	n/a
18	0,4693688	62,00%	0,5919716	56,50%	n/a	n/a
19	0,7061397	53,50%	0,7303283	61,50%	n/a	n/a
Avg	0,7142360	54,75%	0,6806639	59,23%	0,5923843	56,61%

Source: Produced by the author in August, 2022

Table 97 – Results of third experiment on potential domain 3 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6870685	62,00%	0,6795810	55,50%	n/a	n/a
1	0,5913199	54,50%	0,7002931	53,00%	n/a	n/a
2	0,7683762	49,00%	0,6362763	64,50%	n/a	n/a
3	0,8934429	52,00%	0,5866181	61,00%	n/a	n/a
4	0,7538901	54,50%	0,6343632	65,00%	n/a	n/a
5	0,7020081	55,00%	0,6505834	56,50%	n/a	n/a
6	0,6246894	62,50%	0,6215733	62,00%	n/a	n/a
7	0,6639371	56,00%	0,6714834	56,50%	n/a	n/a
8	0,5033884	51,50%	0,8156798	64,00%	n/a	n/a
9	0,7633137	48,50%	0,6545827	64,00%	n/a	n/a
10	0,4252097	60,50%	0,6406782	65,50%	n/a	n/a
11	0,4537961	58,50%	0,6069744	61,50%	n/a	n/a
12	0,9193051	61,50%	0,6974176	55,00%	n/a	n/a
13	0,8321871	50,00%	0,6084006	61,00%	n/a	n/a
14	0,7432975	58,50%	0,5966741	60,00%	n/a	n/a

(Continues...)

Table 97 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
15	0,9300441	50,50%	0,7300285	64,00%	n/a	n/a
16	0,6480483	63,00%	0,4110386	63,00%	n/a	n/a
17	0,7499975	56,50%	0,8502925	64,00%	n/a	n/a
18	0,8093362	55,00%	0,6371533	58,00%	n/a	n/a
19	0,5720633	53,00%	0,5229520	58,50%	n/a	n/a
Avg	0,7017360	55,63%	0,6476322	60,63%	0,6732987	60,33%

Source: Produced by the author in August, 2022

Table 98 – Results of forth experiment on potential domain 3 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6641186	66,00%	0,6307223	59,50%	n/a	n/a
1	0,3326950	55,50%	0,7609306	54,00%	n/a	n/a
2	0,6108747	56,00%	0,6918846	51,50%	n/a	n/a
3	0,8511903	61,50%	0,7178138	55,00%	n/a	n/a
4	0,4605857	57,00%	0,6548058	55,00%	n/a	n/a
5	0,6184274	56,00%	0,6943377	53,50%	n/a	n/a
6	0,7203732	60,00%	0,7479690	43,00%	n/a	n/a
7	0,5607662	60,00%	0,6976023	60,00%	n/a	n/a
8	0,6010306	54,50%	0,5013373	61,50%	n/a	n/a
9	0,8053922	48,50%	0,6645583	62,00%	n/a	n/a
10	0,5557729	55,00%	0,7910934	47,00%	n/a	n/a
11	0,6440837	55,50%	0,7862235	58,00%	n/a	n/a
12	0,5838553	55,50%	0,7872225	64,00%	n/a	n/a
13	0,6979998	55,00%	0,5862199	62,50%	n/a	n/a
14	0,7222812	56,00%	0,6087210	56,00%	n/a	n/a
15	0,6695747	58,00%	0,7233372	58,00%	n/a	n/a
16	0,6230738	55,00%	0,6883730	61,00%	n/a	n/a
17	0,5815804	56,50%	0,5449603	62,50%	n/a	n/a
18	0,7299623	55,00%	0,8263274	47,00%	n/a	n/a
19	0,8789228	47,50%	0,5811953	57,50%	n/a	n/a
Avg	0,6456280	56,20%	0,6842818	56,43%	0,6183876	57,85%

Source: Produced by the author in August, 2022

Table 99 – Results of fifth experiment on potential domain 3 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,9137923	60,00%	0,6213918	52,00%	n/a	n/a
1	0,7544085	61,00%	0,6072705	52,00%	n/a	n/a
2	0,6310616	61,50%	0,3781545	57,50%	n/a	n/a
3	0,8066061	52,50%	0,7014478	57,50%	n/a	n/a
4	0,8416566	55,00%	0,7023979	57,50%	n/a	n/a
5	0,7479749	56,50%	0,8438365	53,00%	n/a	n/a
6	0,6559771	62,50%	0,7906128	52,50%	n/a	n/a
7	0,7623521	53,50%	0,6389679	58,50%	n/a	n/a
8	0,8334299	56,00%	0,7653596	49,00%	n/a	n/a
9	0,8247159	60,50%	0,8008375	56,00%	n/a	n/a
10	0,4466926	60,50%	0,4705229	58,50%	n/a	n/a
11	0,5422219	54,00%	0,6024351	54,50%	n/a	n/a
12	0,7532585	58,50%	0,5778728	54,50%	n/a	n/a
13	0,6958827	45,50%	0,8511255	51,00%	n/a	n/a
14	0,8085523	55,00%	0,7811253	54,50%	n/a	n/a
15	0,5673338	58,50%	0,8800679	52,50%	n/a	n/a
16	0,5132072	58,00%	0,6122471	53,00%	n/a	n/a
17	0,8079929	55,00%	0,4261322	57,00%	n/a	n/a
18	0,6684085	53,00%	0,5785803	55,00%	n/a	n/a
19	0,6946373	62,00%	0,6683707	54,50%	n/a	n/a
Avg	0,7135081	56,95%	0,6649378	54,53%	0,6730917	54,34%

Source: Produced by the author in August, 2022

Table 100 – Results of sixth experiment on potential domain 3 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,4881735	58,50%	0,7603972	54,50%	n/a	n/a
1	0,4819289	50,50%	0,6339924	52,00%	n/a	n/a
2	0,5112709	55,00%	0,7531071	54,50%	n/a	n/a
3	0,9001441	47,00%	0,6302107	55,50%	n/a	n/a
4	0,5341289	55,50%	0,6899717	60,50%	n/a	n/a
5	0,7749376	56,50%	0,6833702	46,50%	n/a	n/a
6	0,7396396	55,50%	0,6406482	60,50%	n/a	n/a
7	0,6097693	61,00%	0,7377437	61,00%	n/a	n/a
8	0,7554463	52,50%	0,6312951	54,00%	n/a	n/a
9	0,7441173	52,50%	0,6352019	48,50%	n/a	n/a

(Continues...)

Table 100 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
10	0,6204565	66,00%	0,7033109	55,00%	n/a	n/a
11	0,9485862	57,00%	0,7012407	55,00%	n/a	n/a
12	0,7932945	57,00%	0,7551714	52,50%	n/a	n/a
13	0,6999271	52,50%	0,8325404	45,50%	n/a	n/a
14	0,6822608	55,00%	0,5997309	58,50%	n/a	n/a
15	0,7971153	49,50%	0,7797943	56,00%	n/a	n/a
16	0,7286519	55,50%	0,6511872	53,50%	n/a	n/a
17	0,5841694	55,00%	0,4705569	56,50%	n/a	n/a
18	0,7007115	56,00%	0,7027329	61,00%	n/a	n/a
19	0,5472460	58,50%	0,7970847	62,50%	n/a	n/a
Avg	0,6820988	55,33%	0,6894644	55,18%	0,5774552	61,16%

Source: Produced by the author in August, 2022

Table 101 – Results of seventh experiment on potential domain 3 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7038934	52,00%	0,7233943	39,00%	n/a	n/a
1	0,7713823	57,50%	0,7482264	59,50%	n/a	n/a
2	0,7914966	64,50%	0,7329632	58,50%	n/a	n/a
3	0,7603214	56,50%	0,6881872	61,00%	n/a	n/a
4	0,8041099	55,50%	0,7356446	62,50%	n/a	n/a
5	0,6731710	63,50%	0,6250685	58,00%	n/a	n/a
6	0,6012223	48,00%	0,6995678	40,00%	n/a	n/a
7	0,5407828	49,00%	0,5665812	62,50%	n/a	n/a
8	0,7397435	57,50%	0,5525682	64,50%	n/a	n/a
9	0,7014984	58,00%	0,5907164	56,50%	n/a	n/a
10	0,6925265	53,00%	0,6609567	61,00%	n/a	n/a
11	0,7610843	50,50%	0,5952563	59,00%	n/a	n/a
12	0,4343076	58,50%	0,9882948	63,00%	n/a	n/a
13	0,6309579	56,50%	0,7724096	60,00%	n/a	n/a
14	0,5087595	62,00%	0,5390228	63,00%	n/a	n/a
15	0,6306598	55,50%	0,5808455	65,00%	n/a	n/a
16	0,8403773	61,00%	0,6384740	65,50%	n/a	n/a
17	0,9518157	57,00%	0,5373110	69,00%	n/a	n/a
18	0,6228547	59,00%	0,4763613	62,00%	n/a	n/a
19	0,3800403	59,50%	0,5434743	63,50%	n/a	n/a
Avg	0,6770503	56,73%	0,6497662	59,65%	0,8977550	59,30%

(Continues...)

Table 101 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc

Source: Produced by the author in August, 2022

Table 102 – Results of eighth experiment on potential domain 3 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7341074	53,50%	0,6336424	59,00%	n/a	n/a
1	0,6689560	46,50%	0,6537696	55,00%	n/a	n/a
2	0,6416297	52,50%	0,6592487	54,00%	n/a	n/a
3	0,6260964	54,00%	0,6507464	53,50%	n/a	n/a
4	0,6768888	46,00%	1,0034909	54,50%	n/a	n/a
5	0,7886999	57,50%	0,5342673	56,50%	n/a	n/a
6	0,5678449	56,50%	0,6542870	58,50%	n/a	n/a
7	1,1240524	59,00%	0,7821704	57,50%	n/a	n/a
8	0,8230589	52,50%	0,7477054	60,00%	n/a	n/a
9	0,6418004	57,50%	0,5639587	56,50%	n/a	n/a
10	0,4742385	49,00%	0,5937425	54,50%	n/a	n/a
11	0,6572030	62,50%	0,7063302	63,00%	n/a	n/a
12	0,9643954	56,50%	0,7068622	58,50%	n/a	n/a
13	0,8906609	50,00%	0,7977347	58,50%	n/a	n/a
14	0,7354618	60,00%	0,5583495	59,50%	n/a	n/a
15	0,4050297	58,00%	0,5767596	62,00%	n/a	n/a
16	0,8903971	52,00%	0,6211618	58,50%	n/a	n/a
17	0,6600763	55,50%	0,8592063	62,50%	n/a	n/a
18	0,7563882	57,00%	0,6244620	56,00%	n/a	n/a
19	0,5877805	55,50%	0,6319265	58,50%	n/a	n/a
Avg	0,7157383	54,58%	0,6779911	57,83%	0,6763110	61,57%

Source: Produced by the author in August, 2022

Table 103 – Results of ninth experiment on potential domain 3 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6496329	56,50%	0,7126866	62,50%	n/a	n/a
1	0,7750357	53,50%	0,7168587	61,00%	n/a	n/a
2	0,7105389	46,50%	0,7143714	50,50%	n/a	n/a
3	0,7136633	55,50%	0,7567494	59,50%	n/a	n/a
4	0,5341228	54,50%	0,5224010	60,50%	n/a	n/a
5	0,5494328	52,50%	0,6274812	58,50%	n/a	n/a

(Continues...)

Table 103 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	0,7566181	51,50%	0,6125885	60,50%	n/a	n/a
7	0,8575017	54,00%	0,6566920	66,00%	n/a	n/a
8	0,5396594	55,00%	0,8548149	63,50%	n/a	n/a
9	0,4773200	49,00%	0,4078164	61,00%	n/a	n/a
10	1,0963209	55,00%	0,4628265	63,00%	n/a	n/a
11	0,9077827	57,00%	0,6176830	62,00%	n/a	n/a
12	0,9614990	54,50%	0,6664241	62,50%	n/a	n/a
13	0,9212701	50,00%	0,6395518	58,50%	n/a	n/a
14	0,6129951	63,00%	0,7022839	60,00%	n/a	n/a
15	0,8416736	54,00%	0,8055416	63,50%	n/a	n/a
16	0,5230885	54,00%	0,5722786	64,50%	n/a	n/a
17	0,6218796	56,50%	0,7247299	61,00%	n/a	n/a
18	0,6558104	50,50%	0,6758971	56,00%	n/a	n/a
19	0,9406255	59,00%	0,7553523	61,00%	n/a	n/a
Avg	0,7323236	54,10%	0,6602514	60,78%	0,6271839	57,02%

Source: Produced by the author in August, 2022

Table 104 – Results of tenth experiment on potential domain 3 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,4813285	61,50%	0,5697549	55,50%	n/a	n/a
1	0,6677714	57,00%	0,6740631	57,00%	n/a	n/a
2	0,5685344	53,50%	0,6382335	55,00%	n/a	n/a
3	0,6691049	60,50%	0,7924929	59,50%	n/a	n/a
4	0,6718622	61,00%	0,5751272	55,50%	n/a	n/a
5	0,7799668	55,50%	0,4680119	55,50%	n/a	n/a
6	0,6591040	47,50%	0,6617377	61,00%	n/a	n/a
7	0,6064798	55,00%	0,8226146	59,00%	n/a	n/a
8	0,7847352	52,50%	0,5192180	61,50%	n/a	n/a
9	0,7099226	57,50%	0,6886063	58,50%	n/a	n/a
10	0,5418903	55,00%	0,5964458	63,50%	n/a	n/a
11	0,7804165	57,00%	0,8270733	61,50%	n/a	n/a
12	0,6339352	58,50%	0,7322036	56,00%	n/a	n/a
13	0,7091462	52,50%	0,8178325	59,00%	n/a	n/a
14	0,9048331	65,00%	0,3579434	57,50%	n/a	n/a
15	0,8524508	53,00%	0,5699328	61,00%	n/a	n/a
16	0,8954930	53,50%	0,6135790	55,50%	n/a	n/a

(Continues...)

Table 104 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
17	0,8727964	55,50%	0,5125406	56,50%	n/a	n/a
18	0,3138582	65,00%	0,5927768	59,50%	n/a	n/a
19	0,8240155	56,50%	0,9125167	60,00%	n/a	n/a
Avg	0,6963823	56,65%	0,6471352	58,40%	0,5894775	57,23%

Source: Produced by the author in August, 2022

Apendix L – Results on potential domain 4 - 2014

Table 105 – Results of first experiment on potential domain 4 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6081263	51,50%	0,4894873	56,00%	n/a	n/a
1	0,7295674	59,00%	0,7378895	44,50%	n/a	n/a
2	0,7842637	60,50%	0,6537529	53,00%	n/a	n/a
3	0,6284886	53,00%	0,6381896	56,50%	n/a	n/a
4	0,7585304	53,00%	0,6662907	63,50%	n/a	n/a
5	0,8033347	54,00%	0,7032309	50,50%	n/a	n/a
6	0,8810052	56,50%	0,7585295	48,00%	n/a	n/a
7	0,7965983	46,50%	0,7796916	42,50%	n/a	n/a
8	0,7721705	57,00%	0,6514870	58,50%	n/a	n/a
9	1,1121877	51,50%	0,6026801	58,00%	n/a	n/a
10	0,4652946	50,00%	0,6018631	50,50%	n/a	n/a
11	0,7846976	59,00%	0,6566644	57,00%	n/a	n/a
12	1,1262107	52,50%	0,7103102	63,50%	n/a	n/a
13	0,7651417	58,50%	0,7713085	40,50%	n/a	n/a
14	0,8598181	54,00%	0,7658062	47,50%	n/a	n/a
15	0,5886347	57,00%	0,7965848	51,50%	n/a	n/a
16	0,8168403	54,00%	0,5660474	64,00%	n/a	n/a
17	0,6470305	45,00%	0,7334867	53,00%	n/a	n/a
18	0,9606361	53,00%	0,7148418	59,00%	n/a	n/a
19	0,5162970	56,50%	0,6254769	65,00%	n/a	n/a
Avg	0,7702437	54,10%	0,6811810	54,13%	0,5502236	65,85%

Source: Produced by the author in August, 2022

Table 106 – Results of second experiment on potential domain 4 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,4108724	61,00%	0,7079775	41,50%	n/a	n/a
1	0,5601917	50,00%	1,0422632	38,00%	n/a	n/a
2	0,5036052	60,00%	0,8942875	43,50%	n/a	n/a
3	0,5548995	55,50%	0,7654839	41,00%	n/a	n/a

(Continues...)

Table 106 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
4	0,6782690	51,00%	0,7272406	44,50%	n/a	n/a
5	0,8242638	50,00%	0,7243512	39,00%	n/a	n/a
6	0,6831923	50,00%	0,6961430	43,00%	n/a	n/a
7	0,7560475	49,50%	0,8231269	45,50%	n/a	n/a
8	0,7447639	53,50%	0,6561186	49,50%	n/a	n/a
9	0,5901414	51,50%	0,7113493	56,00%	n/a	n/a
10	0,6631089	52,50%	0,6172255	63,00%	n/a	n/a
11	0,5905310	58,50%	0,6906521	62,50%	n/a	n/a
12	0,6709660	49,00%	0,5564140	59,50%	n/a	n/a
13	0,7651224	54,00%	0,7148115	48,00%	n/a	n/a
14	0,4548088	53,00%	0,6539933	48,50%	n/a	n/a
15	0,6033424	61,50%	0,6716338	62,50%	n/a	n/a
16	0,5506565	58,50%	0,6645976	66,50%	n/a	n/a
17	0,5937871	59,00%	0,7110605	64,00%	n/a	n/a
18	0,6931088	59,00%	0,5525483	61,50%	n/a	n/a
19	1,0566072	59,50%	0,5709635	66,00%	n/a	n/a
Avg	0,6474143	54,83%	0,7076121	52,18%	0,5770270	66,10%

Source: Produced by the author in August, 2022

Table 107 – Results of third experiment on potential domain 4 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8816395	57,00%	0,7321448	61,00%	n/a	n/a
1	0,6541666	56,50%	0,6789111	49,50%	n/a	n/a
2	0,8789368	51,50%	0,6791050	59,00%	n/a	n/a
3	0,4597856	57,00%	0,6815705	52,00%	n/a	n/a
4	0,8721488	53,00%	0,7643512	45,50%	n/a	n/a
5	0,5115945	57,50%	0,6673479	66,00%	n/a	n/a
6	0,7600129	53,50%	0,7782251	49,50%	n/a	n/a
7	0,5078110	56,50%	0,5009474	64,00%	n/a	n/a
8	0,9946162	52,50%	0,7455548	58,50%	n/a	n/a
9	0,8708112	48,00%	0,7687706	62,50%	n/a	n/a
10	0,5224934	56,00%	0,6905036	61,50%	n/a	n/a
11	0,7515533	51,50%	0,7023733	50,50%	n/a	n/a
12	0,6645061	56,00%	0,6457852	46,00%	n/a	n/a
13	0,5011702	54,50%	0,7060850	57,00%	n/a	n/a
14	0,9075029	57,00%	0,6582871	65,50%	n/a	n/a

(Continues...)

Table 107 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
15	0,7996845	55,50%	0,7418781	58,00%	n/a	n/a
16	0,6772741	53,00%	0,7597095	55,50%	n/a	n/a
17	0,6615399	51,00%	0,7062473	56,50%	n/a	n/a
18	0,6583624	61,50%	0,7298178	44,50%	n/a	n/a
19	0,6649618	56,00%	0,6538869	54,50%	n/a	n/a
Avg	0,7100286	54,75%	0,6995751	55,85%	0,6441435	52,44%

Source: Produced by the author in August, 2022

Table 108 – Results of fourth experiment on potential domain 4 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6434939	53,00%	0,6473128	57,00%	n/a	n/a
1	0,6457874	53,50%	0,6876739	54,50%	n/a	n/a
2	1,4283057	58,00%	0,5034631	63,50%	n/a	n/a
3	0,7057511	61,00%	0,6879058	57,00%	n/a	n/a
4	0,7789262	57,00%	0,6597033	64,00%	n/a	n/a
5	0,7489160	56,50%	0,5775138	59,50%	n/a	n/a
6	0,7631441	56,00%	0,5934583	58,50%	n/a	n/a
7	0,8367339	59,00%	0,6680048	58,00%	n/a	n/a
8	0,4128653	59,50%	0,6911176	65,50%	n/a	n/a
9	0,7261723	49,00%	0,7967438	58,50%	n/a	n/a
10	0,3402473	60,00%	0,8564289	61,00%	n/a	n/a
11	0,7798655	58,50%	0,5193041	61,00%	n/a	n/a
12	0,7399564	55,00%	0,6535321	65,50%	n/a	n/a
13	0,5455220	59,00%	0,7078407	66,50%	n/a	n/a
14	0,4992998	60,00%	0,7993696	62,00%	n/a	n/a
15	0,5697935	55,00%	0,7273147	62,50%	n/a	n/a
16	0,5631607	44,00%	0,5594235	58,50%	n/a	n/a
17	0,8687685	56,00%	0,7076764	55,50%	n/a	n/a
18	1,0249491	54,50%	0,6336782	62,50%	n/a	n/a
19	0,7774323	45,50%	0,6803459	55,00%	n/a	n/a
Avg	0,7199545	55,50%	0,6678906	60,30%	0,6774329	53,41%

Source: Produced by the author in August, 2022

Table 109 – Results of fifth experiment on potential domain 4 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6745308	54,50%	0,6959525	39,00%	n/a	n/a
1	0,7752257	54,50%	0,9423553	38,00%	n/a	n/a
2	0,6115376	49,00%	0,6955447	42,00%	n/a	n/a
3	0,8258994	52,50%	0,7992941	44,50%	n/a	n/a
4	0,6907203	52,50%	0,7219071	46,00%	n/a	n/a
5	0,6627429	53,00%	0,6286254	52,50%	n/a	n/a
6	0,7700449	61,00%	0,7634001	40,50%	n/a	n/a
7	1,2402425	58,50%	0,5749180	59,50%	n/a	n/a
8	0,7203370	55,50%	0,7913868	49,00%	n/a	n/a
9	0,5079840	57,50%	0,6844956	54,00%	n/a	n/a
10	0,4857838	57,00%	0,6040376	56,00%	n/a	n/a
11	0,6492332	50,00%	0,6895025	57,50%	n/a	n/a
12	0,7741783	55,50%	0,6441666	60,50%	n/a	n/a
13	0,8535045	59,00%	0,7448511	68,00%	n/a	n/a
14	0,6618184	54,50%	0,6701334	64,50%	n/a	n/a
15	0,9286667	57,00%	0,7080055	62,00%	n/a	n/a
16	0,6431135	53,00%	0,7217740	62,50%	n/a	n/a
17	0,5849942	58,00%	0,5807657	64,00%	n/a	n/a
18	0,6344603	54,50%	0,7702511	61,00%	n/a	n/a
19	0,4976467	59,50%	0,7483859	62,50%	n/a	n/a
Avg	0,7096332	55,33%	0,7089876	54,18%	0,7759141	66,59%

Source: Produced by the author in August, 2022

Table 110 – Results of sixth experiment on potential domain 4 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6816723	56,00%	0,7524964	54,00%	n/a	n/a
1	0,9873501	48,50%	0,7981317	45,50%	n/a	n/a
2	0,8040397	50,00%	0,6514973	57,50%	n/a	n/a
3	0,7900180	58,00%	0,7161444	59,00%	n/a	n/a
4	0,3793272	59,00%	0,6447614	50,50%	n/a	n/a
5	0,6285372	48,50%	0,7443836	57,00%	n/a	n/a
6	0,6481736	54,50%	0,7753706	45,50%	n/a	n/a
7	0,4428838	59,50%	0,8122407	59,00%	n/a	n/a
8	0,9074064	56,00%	0,6215337	48,00%	n/a	n/a
9	0,8727436	57,50%	0,7678112	63,50%	n/a	n/a

(Continues...)

Table 110 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
10	0,6337296	55,50%	0,6221394	66,50%	n/a	n/a
11	0,5793408	57,00%	0,7948360	49,00%	n/a	n/a
12	0,7852867	48,50%	0,7958535	56,00%	n/a	n/a
13	0,6555230	54,50%	0,6991802	68,00%	n/a	n/a
14	0,4271784	55,00%	0,5085115	63,00%	n/a	n/a
15	0,9666556	49,00%	0,7902486	51,50%	n/a	n/a
16	0,6962595	53,50%	0,6665695	61,50%	n/a	n/a
17	0,6127196	55,50%	0,6272154	59,00%	n/a	n/a
18	0,8242623	56,50%	0,6648738	57,00%	n/a	n/a
19	0,5627125	57,50%	0,6993772	53,50%	n/a	n/a
Avg	0,6942910	54,50%	0,7076588	56,23%	0,6276647	50,24%

Source: Produced by the author in August, 2022

Table 111 – Results of seventh experiment on potential domain 4 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7081461	44,00%	0,7275500	53,50%	n/a	n/a
1	0,4894513	56,00%	0,7264339	55,50%	n/a	n/a
2	0,9023012	62,50%	0,6817836	54,50%	n/a	n/a
3	1,2651412	54,00%	0,6873950	55,50%	n/a	n/a
4	0,5298447	62,00%	0,7622785	56,00%	n/a	n/a
5	0,5612643	55,50%	0,7256879	57,00%	n/a	n/a
6	0,7130365	59,50%	0,7148120	60,50%	n/a	n/a
7	0,7753314	55,50%	0,7967044	60,00%	n/a	n/a
8	0,6783276	55,50%	0,7333826	59,50%	n/a	n/a
9	0,5502323	61,00%	0,8887258	51,00%	n/a	n/a
10	0,9100274	57,00%	0,7341874	59,00%	n/a	n/a
11	0,4201835	55,50%	0,6778898	51,50%	n/a	n/a
12	0,6100092	57,50%	0,6911986	59,50%	n/a	n/a
13	0,6642389	56,50%	0,6281767	60,00%	n/a	n/a
14	0,8278635	56,50%	0,6297758	55,00%	n/a	n/a
15	0,8437483	59,00%	0,8614466	47,00%	n/a	n/a
16	0,7181112	48,50%	0,7631263	51,00%	n/a	n/a
17	0,8935160	59,00%	0,5340034	59,50%	n/a	n/a
18	0,7187541	55,00%	0,6377102	50,50%	n/a	n/a
19	0,5212077	61,50%	0,7132099	53,50%	n/a	n/a
Avg	0,7150368	56,58%	0,7157739	55,48%	0,6073157	55,12%

(Continues...)

Table 111 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc

Source: Produced by the author in August, 2022

Table 112 – Results of eighth experiment on potential domain 4 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7490720	55,50%	0,7306303	56,00%	n/a	n/a
1	0,5359434	55,00%	0,7270893	56,00%	n/a	n/a
2	0,6155018	58,50%	0,7814893	57,50%	n/a	n/a
3	0,7494229	58,00%	0,6793118	49,50%	n/a	n/a
4	0,5579625	49,00%	0,6249387	61,50%	n/a	n/a
5	0,6194897	60,00%	0,5332508	64,50%	n/a	n/a
6	0,5985855	61,50%	0,7852109	55,50%	n/a	n/a
7	0,5576696	56,50%	0,8991957	56,00%	n/a	n/a
8	0,7596680	51,50%	0,6571461	61,50%	n/a	n/a
9	0,7681270	52,00%	0,6279223	62,50%	n/a	n/a
10	0,7412938	51,50%	0,7047834	60,50%	n/a	n/a
11	0,9781997	62,50%	0,5677103	61,00%	n/a	n/a
12	0,9185913	53,00%	0,5736167	59,50%	n/a	n/a
13	0,6204582	52,50%	0,8261514	62,00%	n/a	n/a
14	1,1427054	59,00%	0,8388900	64,00%	n/a	n/a
15	0,5827481	51,00%	0,7105053	59,50%	n/a	n/a
16	0,7865866	56,50%	0,7931696	62,00%	n/a	n/a
17	0,8090694	48,00%	0,7100203	56,00%	n/a	n/a
18	0,6066269	55,00%	0,7209507	58,00%	n/a	n/a
19	0,6446764	50,50%	0,7144163	48,50%	n/a	n/a
Avg	0,7171199	54,85%	0,7103200	58,58%	0,2738509	48,54%

Source: Produced by the author in August, 2022

Table 113 – Results of ninth experiment on potential domain 4 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6011857	60,50%	0,7280989	40,50%	n/a	n/a
1	0,6419076	62,00%	0,6796900	47,00%	n/a	n/a
2	0,6200502	55,00%	0,7250043	55,00%	n/a	n/a
3	0,9401734	56,50%	0,6487193	53,00%	n/a	n/a
4	0,6376168	58,00%	0,6566908	53,00%	n/a	n/a
5	0,3953367	56,50%	0,6542581	47,50%	n/a	n/a

(Continues...)

Table 113 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	0,7254937	55,50%	0,6233989	61,00%	n/a	n/a
7	0,5315281	59,50%	0,6498781	54,00%	n/a	n/a
8	0,9202583	50,00%	0,7808449	44,50%	n/a	n/a
9	0,5973383	57,00%	0,7615020	44,50%	n/a	n/a
10	0,7021551	51,50%	0,7111621	54,50%	n/a	n/a
11	0,6191141	59,50%	0,7249511	46,00%	n/a	n/a
12	1,0268692	59,50%	0,8662406	48,50%	n/a	n/a
13	0,6322786	53,50%	0,7817467	51,50%	n/a	n/a
14	0,6640462	55,50%	0,8276460	49,50%	n/a	n/a
15	1,0801089	57,00%	0,5491423	55,00%	n/a	n/a
16	0,8285168	63,50%	0,8664450	51,00%	n/a	n/a
17	0,6122122	63,00%	0,5928596	54,00%	n/a	n/a
18	0,7543010	55,50%	0,9549810	45,50%	n/a	n/a
19	0,6178685	56,00%	1,0609951	45,00%	n/a	n/a
Avg	0,7074180	57,25%	0,7422127	50,03%	0,7784818	43,90%

Source: Produced by the author in August, 2022

Table 114 – Results of tenth experiment on potential domain 4 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7409875	53,50%	0,7438707	42,00%	n/a	n/a
1	0,6629638	52,50%	0,7718638	45,00%	n/a	n/a
2	0,7145148	54,00%	0,7085813	46,50%	n/a	n/a
3	0,6146868	52,50%	0,7242041	44,00%	n/a	n/a
4	1,2062471	57,00%	0,7069325	51,00%	n/a	n/a
5	0,8049478	54,00%	0,7964951	42,50%	n/a	n/a
6	0,9030702	64,50%	0,5967480	54,00%	n/a	n/a
7	0,5255176	59,00%	0,7053201	49,50%	n/a	n/a
8	0,7374040	65,00%	0,6927061	50,50%	n/a	n/a
9	0,8047169	56,00%	0,7197477	58,50%	n/a	n/a
10	0,7206758	55,50%	0,8282294	49,00%	n/a	n/a
11	0,8278399	46,50%	0,6090266	52,50%	n/a	n/a
12	0,6746604	59,00%	0,7385415	57,00%	n/a	n/a
13	0,7733864	51,00%	0,6952333	51,00%	n/a	n/a
14	1,2218906	58,00%	0,7905595	58,50%	n/a	n/a
15	0,8184932	55,50%	0,5399909	62,50%	n/a	n/a
16	0,7445478	61,50%	0,5308466	58,00%	n/a	n/a

(Continues...)

Table 114 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
17	0,7091805	64,00%	0,7965087	55,50%	n/a	n/a
18	0,8699702	55,00%	0,7630116	55,00%	n/a	n/a
19	0,7793208	53,50%	0,5926084	49,00%	n/a	n/a
Avg	0,7927511	56,38%	0,7025513	51,58%	0,8012449	47,56%

Source: Produced by the author in August, 2022

Apendix M – Results on potential domain 5 - 2014

Table 115 – Results of first experiment on potential domain 5 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7540673	44,00%	0,7580329	49,00%	n/a	n/a
1	0,5904794	43,50%	0,6933156	53,50%	n/a	n/a
2	0,5250906	53,50%	0,8740427	55,00%	n/a	n/a
3	0,7249568	49,50%	0,5008098	49,00%	n/a	n/a
4	0,7778542	53,50%	0,6679236	56,50%	n/a	n/a
5	0,5554676	48,50%	0,6822451	54,00%	n/a	n/a
6	0,8194582	50,50%	0,7834065	55,00%	n/a	n/a
7	0,7062719	53,50%	0,7443408	49,00%	n/a	n/a
8	0,5986171	47,50%	0,7696520	59,00%	n/a	n/a
9	0,5327481	51,00%	0,7874384	57,50%	n/a	n/a
10	0,6795456	55,00%	0,7385384	58,50%	n/a	n/a
11	1,0133518	59,00%	0,6459127	55,50%	n/a	n/a
12	0,6119487	49,50%	0,5912776	57,00%	n/a	n/a
13	0,6240003	54,50%	0,7355720	61,50%	n/a	n/a
14	0,5829911	55,50%	0,7030429	50,00%	n/a	n/a
15	0,6288728	51,00%	0,6801553	53,50%	n/a	n/a
16	0,5411990	49,00%	0,6816000	58,00%	n/a	n/a
17	1,0694207	54,50%	0,6563545	55,50%	n/a	n/a
18	0,5583066	58,00%	0,6641638	55,50%	n/a	n/a
19	0,5452586	61,50%	0,6067063	48,50%	n/a	n/a
Avg	0,6719953	52,13%	0,6982265	54,55%	0,7081883	50,31%

Source: Produced by the author in August, 2022

Table 116 – Results of second experiment on potential domain 5 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6194897	46,00%	0,7271953	52,00%	n/a	n/a
1	0,5871117	48,00%	0,7121474	48,00%	n/a	n/a
2	0,5685880	53,50%	0,5748706	50,50%	n/a	n/a
3	0,7353604	48,00%	0,8082964	52,00%	n/a	n/a

(Continues...)

Table 116 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
4	0,7988375	46,50%	0,6226605	49,00%	n/a	n/a
5	0,7432112	51,50%	0,7508185	46,50%	n/a	n/a
6	0,7262624	47,00%	0,7107857	48,00%	n/a	n/a
7	0,6233178	54,50%	0,7295302	52,50%	n/a	n/a
8	0,7175604	52,50%	0,7612151	50,50%	n/a	n/a
9	0,6010510	53,00%	0,4943933	52,50%	n/a	n/a
10	0,7308680	43,50%	0,6625400	46,00%	n/a	n/a
11	0,7985026	46,50%	0,7659709	53,00%	n/a	n/a
12	0,8350683	56,00%	0,5817958	53,00%	n/a	n/a
13	0,8383471	50,00%	0,5986947	55,00%	n/a	n/a
14	0,5816098	48,50%	0,7591491	50,50%	n/a	n/a
15	0,5493878	57,00%	0,6503148	49,50%	n/a	n/a
16	0,6774108	46,50%	0,7344564	50,50%	n/a	n/a
17	0,8743953	53,00%	0,7113332	55,50%	n/a	n/a
18	0,6853073	52,00%	0,8121715	56,50%	n/a	n/a
19	0,5577509	51,00%	0,6334536	55,00%	n/a	n/a
Avg	0,6924719	50,23%	0,6900896	51,30%	0,8076079	56,78%

Source: Produced by the author in August, 2022

Table 117 – Results of third experiment on potential domain 5 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8862469	46,50%	0,6255115	51,00%	n/a	n/a
1	0,6900964	52,00%	0,7297321	55,50%	n/a	n/a
2	0,9665200	56,50%	0,7422290	53,00%	n/a	n/a
3	0,7046117	53,00%	0,6172128	52,50%	n/a	n/a
4	0,7248212	56,50%	0,8556183	43,00%	n/a	n/a
5	0,7118222	49,00%	0,7524053	47,00%	n/a	n/a
6	0,6248403	56,50%	0,7358120	50,00%	n/a	n/a
7	0,7182568	51,00%	0,6778639	56,50%	n/a	n/a
8	0,5225994	48,00%	0,7311713	59,50%	n/a	n/a
9	0,8149899	52,00%	0,6452336	57,50%	n/a	n/a
10	0,8795680	49,50%	0,7090198	51,00%	n/a	n/a
11	0,8403386	52,50%	0,6986931	48,50%	n/a	n/a
12	0,9118563	51,50%	0,6615013	56,00%	n/a	n/a
13	0,7133609	50,50%	0,7632456	55,00%	n/a	n/a
14	0,5155479	55,00%	0,7298433	43,50%	n/a	n/a

(Continues...)

Table 117 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
15	0,7641981	53,50%	0,7375122	50,50%	n/a	n/a
16	0,6709603	50,50%	0,6566175	49,50%	n/a	n/a
17	0,7470175	52,50%	0,7630687	59,50%	n/a	n/a
18	0,4786085	49,50%	0,6056011	47,50%	n/a	n/a
19	0,9846410	52,50%	0,6802604	45,00%	n/a	n/a
Avg	0,7435451	51,93%	0,7059076	51,58%	0,5678497	51,15%

Source: Produced by the author in August, 2022

Table 118 – Results of fourth experiment on potential domain 5 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8548963	51,00%	0,6331580	48,00%	n/a	n/a
1	0,3615424	57,50%	0,7186653	49,50%	n/a	n/a
2	0,7602921	46,00%	0,7880324	47,50%	n/a	n/a
3	0,8065146	60,00%	0,8722433	48,50%	n/a	n/a
4	0,6768748	48,50%	0,7912724	50,50%	n/a	n/a
5	0,6732204	47,00%	0,6199889	51,00%	n/a	n/a
6	0,7829081	56,00%	0,6846693	55,50%	n/a	n/a
7	0,6598513	56,50%	0,7050169	59,50%	n/a	n/a
8	0,6015976	49,00%	0,6827433	49,00%	n/a	n/a
9	0,6618092	48,00%	0,6798286	55,50%	n/a	n/a
10	0,8871081	52,00%	0,7291680	51,50%	n/a	n/a
11	0,9038711	53,50%	0,5897577	51,00%	n/a	n/a
12	0,6166061	54,00%	0,6283042	51,00%	n/a	n/a
13	0,7289512	54,00%	0,6882629	52,50%	n/a	n/a
14	0,5915064	48,00%	0,5947884	50,00%	n/a	n/a
15	0,8752702	50,50%	0,6098164	57,50%	n/a	n/a
16	0,6292245	59,00%	0,5934720	52,00%	n/a	n/a
17	0,8391395	46,50%	0,6171964	56,00%	n/a	n/a
18	0,7136066	46,50%	0,6193557	56,00%	n/a	n/a
19	0,7155083	57,50%	0,7647299	57,00%	n/a	n/a
Avg	0,7170149	52,05%	0,6805235	52,45%	0,7613322	54,70%

Source: Produced by the author in August, 2022

Table 119 – Results of fifth experiment on potential domain 5 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5717916	53,00%	0,7571138	51,50%	n/a	n/a
1	0,6181226	51,50%	0,6448163	54,50%	n/a	n/a
2	0,8190701	49,50%	0,7759424	46,00%	n/a	n/a
3	0,6842700	50,50%	0,6345034	50,00%	n/a	n/a
4	0,9795473	44,50%	0,6549673	45,50%	n/a	n/a
5	0,6906469	51,50%	0,6943339	51,50%	n/a	n/a
6	0,6144359	49,50%	0,6922892	49,50%	n/a	n/a
7	0,6856467	52,00%	0,6741787	46,50%	n/a	n/a
8	0,9583260	53,50%	0,5763364	58,50%	n/a	n/a
9	0,6047966	52,50%	0,6951308	56,50%	n/a	n/a
10	0,7907020	55,00%	0,6717728	51,00%	n/a	n/a
11	0,7069914	55,00%	0,7272914	51,50%	n/a	n/a
12	0,5657014	50,00%	0,7514920	50,50%	n/a	n/a
13	0,7170497	54,00%	0,6139340	50,00%	n/a	n/a
14	0,5890820	50,00%	0,6902922	45,50%	n/a	n/a
15	0,5836518	57,00%	0,7121254	49,00%	n/a	n/a
16	0,9873754	46,00%	0,6365097	54,00%	n/a	n/a
17	0,3948560	58,50%	0,6641295	52,50%	n/a	n/a
18	0,8012709	52,50%	0,6967160	51,00%	n/a	n/a
19	1,0575947	56,00%	0,7214463	49,50%	n/a	n/a
Avg	0,7210464	52,10%	0,6842661	50,73%	0,6560815	50,94%

Source: Produced by the author in August, 2022

Table 120 – Results of sixth experiment on potential domain 5 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7310320	48,50%	0,7146537	51,50%	n/a	n/a
1	0,5448997	48,00%	0,6802838	53,50%	n/a	n/a
2	0,8714485	52,00%	0,7037063	53,00%	n/a	n/a
3	0,8294746	48,00%	0,6175922	56,00%	n/a	n/a
4	0,9261746	47,00%	0,6323699	47,00%	n/a	n/a
5	0,6888762	55,00%	0,6846735	48,00%	n/a	n/a
6	0,6323355	51,50%	0,7075694	52,00%	n/a	n/a
7	0,5531017	53,00%	0,7198092	52,50%	n/a	n/a
8	0,6239914	49,50%	0,6367910	51,50%	n/a	n/a
9	0,7430499	55,00%	0,6570532	53,00%	n/a	n/a

(Continues...)

Table 120 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
10	0,3146291	54,50%	0,6880711	48,50%	n/a	n/a
11	0,7012154	51,00%	0,8966377	59,50%	n/a	n/a
12	0,6575916	58,00%	0,5965097	59,00%	n/a	n/a
13	0,6263129	61,50%	0,5998202	54,50%	n/a	n/a
14	0,7286777	50,50%	0,5946127	56,00%	n/a	n/a
15	0,7927108	57,50%	0,6561634	62,00%	n/a	n/a
16	0,7273715	48,00%	0,7417051	53,50%	n/a	n/a
17	0,5665522	48,00%	0,6998398	59,50%	n/a	n/a
18	0,7349272	49,00%	0,6917302	51,00%	n/a	n/a
19	0,4821075	57,00%	0,5702118	58,00%	n/a	n/a
Avg	0,6738240	52,13%	0,6744902	53,98%	0,9006509	53,03%

Source: Produced by the author in August, 2022

Table 121 – Results of seventh experiment on potential domain 5 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7074000	51,50%	0,6914537	48,00%	n/a	n/a
1	1,1195264	53,50%	0,5814027	51,00%	n/a	n/a
2	0,6993254	51,50%	0,7611849	56,50%	n/a	n/a
3	0,8119426	54,50%	0,7199941	60,00%	n/a	n/a
4	0,6245542	50,50%	0,6402454	59,50%	n/a	n/a
5	0,6444410	54,00%	0,6015896	55,00%	n/a	n/a
6	0,5588780	51,50%	0,6870514	56,50%	n/a	n/a
7	0,8339882	52,50%	0,6644937	54,50%	n/a	n/a
8	0,7618073	56,50%	0,6556464	56,00%	n/a	n/a
9	0,6578102	48,00%	0,6511703	51,50%	n/a	n/a
10	0,6439764	53,00%	0,6789047	53,00%	n/a	n/a
11	0,7938975	49,00%	0,6561646	62,50%	n/a	n/a
12	1,0006685	54,00%	0,6078331	47,00%	n/a	n/a
13	0,6203269	50,50%	0,7157099	47,50%	n/a	n/a
14	0,7042044	51,50%	0,7338142	47,00%	n/a	n/a
15	0,8286440	48,50%	0,5733569	48,50%	n/a	n/a
16	0,5517068	53,00%	0,6971701	53,00%	n/a	n/a
17	0,7199406	53,00%	0,7405114	60,50%	n/a	n/a
18	0,8371248	61,00%	0,6519504	53,00%	n/a	n/a
19	0,5886490	57,50%	0,6059523	50,50%	n/a	n/a
Avg	0,7354406	52,75%	0,6657800	53,55%	0,7866319	50,94%

(Continues...)

Table 121 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc

Source: Produced by the author in August, 2022

Table 122 – Results of eighth experiment on potential domain 5 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8507634	50,50%	0,7843846	45,00%	n/a	n/a
1	0,7488607	48,50%	0,8197500	44,50%	n/a	n/a
2	0,9896873	55,00%	0,7830029	53,00%	n/a	n/a
3	0,6061395	49,50%	0,7324453	57,00%	n/a	n/a
4	0,7499442	44,50%	0,6781268	51,50%	n/a	n/a
5	0,5529329	47,00%	0,7048296	52,00%	n/a	n/a
6	0,8443817	52,00%	0,7559984	49,50%	n/a	n/a
7	0,6496975	56,00%	0,8662551	51,00%	n/a	n/a
8	0,8177434	50,50%	0,7028323	49,00%	n/a	n/a
9	0,5951213	56,50%	0,8318824	51,50%	n/a	n/a
10	0,6847336	48,50%	0,7817512	47,50%	n/a	n/a
11	0,8593373	46,00%	0,7144915	50,50%	n/a	n/a
12	0,8560928	47,00%	0,7018263	50,00%	n/a	n/a
13	0,6480983	53,00%	0,7154268	61,00%	n/a	n/a
14	0,7663789	49,50%	0,5473825	51,50%	n/a	n/a
15	0,7187139	54,00%	0,6880830	53,00%	n/a	n/a
16	0,5190924	58,00%	0,6185479	51,50%	n/a	n/a
17	0,8743670	50,50%	0,7135144	52,00%	n/a	n/a
18	0,3727172	53,00%	0,7093384	60,50%	n/a	n/a
19	0,8021188	53,50%	0,6815084	53,00%	n/a	n/a
Avg	0,7253461	51,15%	0,7265689	51,73%	0,6818504	53,65%

Source: Produced by the author in August, 2022

Table 123 – Results of ninth experiment on potential domain 5 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6891657	49,00%	0,6680017	51,50%	n/a	n/a
1	0,5013593	53,00%	0,6787111	47,50%	n/a	n/a
2	0,8062752	49,00%	0,8835089	48,00%	n/a	n/a
3	0,7439861	50,50%	0,8943075	54,00%	n/a	n/a
4	0,4512794	50,00%	0,7422107	55,50%	n/a	n/a
5	0,7237574	57,00%	0,7005202	47,50%	n/a	n/a

(Continues...)

Table 123 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	0,7103330	52,00%	0,6111567	54,50%	n/a	n/a
7	0,7593162	50,50%	0,6932792	53,00%	n/a	n/a
8	0,8326331	45,50%	0,7279583	52,50%	n/a	n/a
9	0,5750144	54,50%	0,6808982	47,50%	n/a	n/a
10	0,5305741	50,00%	0,5629349	55,00%	n/a	n/a
11	0,8694117	51,00%	0,7305223	54,50%	n/a	n/a
12	0,7191555	56,50%	0,6536542	57,50%	n/a	n/a
13	0,6905301	56,00%	0,7209386	58,50%	n/a	n/a
14	0,7186323	45,00%	0,5670030	57,00%	n/a	n/a
15	0,7131288	52,00%	0,5948185	53,00%	n/a	n/a
16	0,8042989	48,50%	0,7406226	54,50%	n/a	n/a
17	0,8254848	50,50%	0,8509480	47,00%	n/a	n/a
18	0,6453524	61,00%	0,5683274	58,50%	n/a	n/a
19	0,7011778	50,50%	0,7169999	56,00%	n/a	n/a
Avg	0,7005433	51,60%	0,6993661	53,15%	0,6238798	56,99%

Source: Produced by the author in August, 2022

Table 124 – Results of tenth experiment on potential domain 5 - 2014

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8124883	48,00%	0,7579418	48,00%	n/a	n/a
1	0,7323043	45,00%	0,6850967	55,00%	n/a	n/a
2	0,6925761	52,00%	0,6717163	50,00%	n/a	n/a
3	0,7237319	49,50%	0,9347386	50,00%	n/a	n/a
4	0,7617073	51,00%	0,8449460	57,00%	n/a	n/a
5	0,7594600	50,50%	0,7571132	54,00%	n/a	n/a
6	0,6418782	55,00%	0,5569863	60,00%	n/a	n/a
7	0,5497659	54,00%	0,9696869	53,00%	n/a	n/a
8	0,7845626	54,50%	0,8450691	55,50%	n/a	n/a
9	0,9581296	54,00%	0,6896876	49,50%	n/a	n/a
10	0,6663966	47,50%	0,6720573	51,50%	n/a	n/a
11	0,7755174	55,50%	0,5546490	51,50%	n/a	n/a
12	0,7406391	57,00%	0,6109614	51,00%	n/a	n/a
13	0,6855889	52,50%	0,6855206	56,50%	n/a	n/a
14	0,5617920	59,00%	0,7113147	49,00%	n/a	n/a
15	0,8725824	54,00%	0,6724721	55,00%	n/a	n/a
16	0,7398628	47,50%	0,7623018	52,00%	n/a	n/a

(Continues...)

Table 124 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
17	0,8005518	50,00%	0,6467765	56,00%	n/a	n/a
18	0,6062400	57,00%	0,6515483	57,00%	n/a	n/a
19	0,7423159	54,50%	0,7303323	58,00%	n/a	n/a
Avg	0,7304046	52,40%	0,7205458	53,48%	0,7831599	49,48%

Source: Produced by the author in August, 2022

Apendix N – Results on potential domain 6 - 2014 and 2015

Table 125 – Results of first experiment on potential domain 6 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8321533	65,00%	0,7736513	65,00%	n/a	n/a
1	0,6291039	66,00%	0,7982594	61,00%	n/a	n/a
2	0,8142059	55,00%	0,5778221	67,00%	n/a	n/a
3	0,5971720	56,00%	0,4444343	68,00%	n/a	n/a
4	0,5674155	67,50%	0,5540680	62,50%	n/a	n/a
5	1,0718991	66,50%	0,6726733	69,50%	n/a	n/a
6	0,7364009	65,50%	0,3819540	64,00%	n/a	n/a
7	0,6026303	69,50%	0,5938124	73,00%	n/a	n/a
8	0,4769607	66,50%	0,8241494	62,00%	n/a	n/a
9	0,7542661	59,50%	0,5313930	64,50%	n/a	n/a
10	0,6234013	59,50%	0,5669930	62,50%	n/a	n/a
11	0,5560945	60,00%	0,4562902	65,00%	n/a	n/a
12	1,0314556	65,00%	0,7910867	68,00%	n/a	n/a
13	0,5614885	62,50%	0,5192348	71,50%	n/a	n/a
14	0,6135522	62,50%	0,5959697	63,50%	n/a	n/a
15	0,6147956	67,00%	0,5403460	70,50%	n/a	n/a
16	0,8269032	58,50%	0,6086374	69,50%	n/a	n/a
17	1,0096679	64,00%	0,8058081	65,50%	n/a	n/a
18	0,8317743	71,00%	0,8300049	63,50%	n/a	n/a
19	0,5172360	70,00%	0,7162151	70,00%	n/a	n/a
Avg	0,7134288	63,85%	0,6291402	66,30%	0,5748977	63,78%

Source: Produced by the author in August, 2022

Table 126 – Results of second experiment on potential domain 6 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,4934905	65,50%	0,5114219	62,50%	n/a	n/a
1	0,7385553	66,00%	0,6524612	61,00%	n/a	n/a
2	0,5917999	64,00%	0,6010629	64,00%	n/a	n/a
3	0,6915146	68,00%	0,5872219	65,00%	n/a	n/a

(Continues...)

Table 126 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
4	0,6156778	56,00%	0,7978951	63,00%	n/a	n/a
5	0,4975334	71,50%	0,5948281	65,50%	n/a	n/a
6	0,5521276	67,00%	0,3603312	64,00%	n/a	n/a
7	0,6511538	63,00%	0,7324753	64,50%	n/a	n/a
8	0,7879463	58,00%	0,5518252	71,00%	n/a	n/a
9	0,7737336	66,00%	0,6974726	63,50%	n/a	n/a
10	0,5316560	70,50%	0,5131831	65,50%	n/a	n/a
11	0,6325590	67,00%	0,6892713	64,00%	n/a	n/a
12	0,5860924	63,50%	0,5849859	65,00%	n/a	n/a
13	0,9821805	65,50%	0,9281651	57,50%	n/a	n/a
14	0,3687284	60,50%	0,7683762	66,50%	n/a	n/a
15	0,8135073	61,00%	0,7628245	62,00%	n/a	n/a
16	0,8723797	66,00%	0,8062225	67,00%	n/a	n/a
17	0,5753573	56,00%	0,6341913	72,00%	n/a	n/a
18	0,7480741	62,00%	0,4553122	66,00%	n/a	n/a
19	0,4905535	64,50%	0,5950440	67,00%	n/a	n/a
Avg	0,6497310	64,08%	0,6412286	64,83%	0,5791090	66,84%

Source: Produced by the author in August, 2022

Table 127 – Results of third experiment on potential domain 6 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8371575	62,50%	0,5726921	67,00%	n/a	n/a
1	0,6075416	69,00%	0,7340399	70,00%	n/a	n/a
2	1,0070509	65,00%	0,5961489	64,00%	n/a	n/a
3	0,7808104	56,00%	0,5744971	68,00%	n/a	n/a
4	0,7617021	60,00%	0,8651090	63,50%	n/a	n/a
5	0,4793542	66,00%	0,5464565	63,50%	n/a	n/a
6	0,7107223	61,50%	0,6063488	66,50%	n/a	n/a
7	1,0154814	66,00%	1,0258023	64,50%	n/a	n/a
8	0,8882489	63,00%	0,5493524	67,00%	n/a	n/a
9	0,9231346	67,50%	0,4118217	62,50%	n/a	n/a
10	0,6202670	66,00%	0,5616413	62,00%	n/a	n/a
11	0,7335783	61,50%	0,5314060	70,00%	n/a	n/a
12	0,7983015	66,00%	0,5017830	64,00%	n/a	n/a
13	0,7186069	64,50%	0,8740571	69,50%	n/a	n/a
14	0,6703324	62,00%	0,7399630	65,50%	n/a	n/a

(Continues...)

Table 127 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
15	0,5100966	68,00%	0,6974051	63,50%	n/a	n/a
16	0,5519881	69,00%	0,4641134	61,50%	n/a	n/a
17	0,9119936	66,50%	0,6963508	65,00%	n/a	n/a
18	0,7054776	57,00%	0,7632543	65,00%	n/a	n/a
19	0,5061051	59,00%	0,4685851	62,00%	n/a	n/a
Avg	0,7368975	63,80%	0,6390414	65,23%	0,6299263	64,29%

Source: Produced by the author in August, 2022

Table 128 – Results of fourth experiment on potential domain 6 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7861530	68,00%	0,4711058	58,50%	n/a	n/a
1	0,6810029	59,00%	0,8020649	56,00%	n/a	n/a
2	0,5657085	62,00%	0,6535977	60,50%	n/a	n/a
3	1,1350232	64,00%	0,4544531	61,00%	n/a	n/a
4	0,7442732	45,00%	0,6500999	59,00%	n/a	n/a
5	0,5645896	61,00%	0,9206113	57,00%	n/a	n/a
6	0,5739813	69,00%	0,6090462	64,00%	n/a	n/a
7	0,2203609	68,00%	0,3831242	51,50%	n/a	n/a
8	0,4273864	69,50%	0,5666980	56,50%	n/a	n/a
9	0,9742475	64,00%	0,9620583	54,50%	n/a	n/a
10	0,8291317	47,50%	0,6961162	63,00%	n/a	n/a
11	1,0572424	65,50%	0,7075562	60,00%	n/a	n/a
12	0,3598958	64,00%	0,8017851	55,00%	n/a	n/a
13	0,6363149	64,00%	0,6768366	58,50%	n/a	n/a
14	0,5318850	65,00%	0,6240584	61,50%	n/a	n/a
15	0,5465351	62,00%	0,6884882	55,50%	n/a	n/a
16	0,4417589	66,00%	0,8992695	60,00%	n/a	n/a
17	0,6630413	59,50%	0,7182842	58,00%	n/a	n/a
18	0,6218537	62,00%	0,7070100	60,00%	n/a	n/a
19	0,4725397	66,50%	0,6826466	55,00%	n/a	n/a
Avg	0,6416462	62,58%	0,6837455	58,25%	0,5149330	67,86%

Source: Produced by the author in August, 2022

Table 129 – Results of fifth experiment on potential domain 6 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6890488	59,50%	0,6697116	67,00%	n/a	n/a
1	0,9080119	55,00%	0,5300789	63,50%	n/a	n/a
2	0,7274370	62,50%	0,6974670	65,00%	n/a	n/a
3	0,6092663	61,00%	0,6221944	67,50%	n/a	n/a
4	0,6520426	58,50%	0,6749852	66,50%	n/a	n/a
5	0,4021018	63,00%	0,7776552	68,50%	n/a	n/a
6	0,5952622	66,00%	0,5780278	63,00%	n/a	n/a
7	0,6545390	54,00%	0,7800400	67,50%	n/a	n/a
8	0,6339939	66,00%	0,6552896	70,50%	n/a	n/a
9	0,4944214	62,50%	0,8401035	68,00%	n/a	n/a
10	0,8458901	56,00%	0,5605755	63,50%	n/a	n/a
11	0,4059299	65,50%	1,2016467	62,00%	n/a	n/a
12	0,5720451	64,50%	0,9730522	67,00%	n/a	n/a
13	0,5851322	65,00%	0,5418415	65,50%	n/a	n/a
14	0,5420436	55,00%	0,6263713	66,50%	n/a	n/a
15	0,5848735	63,00%	0,7547959	62,00%	n/a	n/a
16	1,1292697	57,50%	0,3941185	60,50%	n/a	n/a
17	0,5002721	65,00%	0,7757818	68,50%	n/a	n/a
18	0,3967641	63,50%	0,5968634	66,00%	n/a	n/a
19	0,7175875	63,50%	0,8816407	65,00%	n/a	n/a
Avg	0,6322966	61,33%	0,7066120	65,68%	0,4526347	64,03%

Source: Produced by the author in August, 2022

Table 130 – Results of sixth experiment on potential domain 6 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6843742	68,00%	0,7166967	65,00%	n/a	n/a
1	0,4942955	58,50%	0,7351290	55,00%	n/a	n/a
2	0,5803381	67,50%	0,6369383	55,50%	n/a	n/a
3	0,7121139	67,00%	0,5838130	53,50%	n/a	n/a
4	0,7035142	63,50%	0,6671640	65,00%	n/a	n/a
5	0,3746296	64,00%	0,7201809	67,50%	n/a	n/a
6	1,1457052	67,00%	0,6678171	66,50%	n/a	n/a
7	0,4893154	54,50%	0,5922668	68,50%	n/a	n/a
8	0,1848147	67,50%	0,4868366	68,00%	n/a	n/a
9	0,6212834	68,00%	0,5420704	65,50%	n/a	n/a

(Continues...)

Table 130 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
10	0,4572289	65,50%	0,6828558	64,00%	n/a	n/a
11	0,5662085	60,50%	0,6507350	60,50%	n/a	n/a
12	0,8826888	66,00%	0,5856364	66,50%	n/a	n/a
13	0,7017657	54,50%	0,6209073	63,50%	n/a	n/a
14	1,1367719	66,50%	0,5614845	68,50%	n/a	n/a
15	0,7630742	63,00%	0,7104832	64,50%	n/a	n/a
16	0,6443307	60,00%	0,7159159	47,00%	n/a	n/a
17	0,6372122	63,50%	0,7685173	54,50%	n/a	n/a
18	0,9710930	66,50%	0,8426638	64,50%	n/a	n/a
19	0,8803674	63,00%	0,6033818	66,00%	n/a	n/a
Avg	0,6815563	63,73%	0,6545747	62,48%	0,5362655	63,39%

Source: Produced by the author in August, 2022

Table 131 – Results of seventh experiment on potential domain 6 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7637714	66,50%	0,4699414	65,00%	n/a	n/a
1	0,4326880	71,00%	0,5607634	64,50%	n/a	n/a
2	0,8112221	56,50%	0,6362232	63,00%	n/a	n/a
3	1,0085167	67,00%	0,4386015	68,50%	n/a	n/a
4	1,3012996	62,00%	0,9183431	60,00%	n/a	n/a
5	0,8117237	61,50%	0,7815294	68,50%	n/a	n/a
6	0,5889044	64,00%	0,5895927	56,00%	n/a	n/a
7	0,7665002	61,50%	0,7651545	66,50%	n/a	n/a
8	0,5102606	61,00%	0,6822885	66,00%	n/a	n/a
9	0,4877871	62,00%	0,4177117	62,50%	n/a	n/a
10	0,5095684	68,00%	0,6823256	59,50%	n/a	n/a
11	0,5743450	56,00%	0,5863006	64,00%	n/a	n/a
12	0,7384865	60,00%	0,7032330	63,00%	n/a	n/a
13	0,3874261	59,00%	0,7305996	62,00%	n/a	n/a
14	0,4859697	59,00%	0,8073571	62,50%	n/a	n/a
15	0,7127695	61,00%	0,8439460	63,50%	n/a	n/a
16	0,2200960	66,50%	0,5906761	66,50%	n/a	n/a
17	0,3141185	65,50%	0,5662330	61,00%	n/a	n/a
18	0,5952684	64,00%	0,6403524	64,00%	n/a	n/a
19	0,6657954	62,50%	0,7691841	58,00%	n/a	n/a
Avg	0,6343259	62,73%	0,6590179	63,23%	0,7549970	62,37%

(Continues...)

Table 131 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc

Source: Produced by the author in August, 2022

Table 132 – Results of eighth experiment on potential domain 6 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5560352	65,50%	0,5148154	64,50%	n/a	n/a
1	0,9670594	65,00%	0,7786527	66,50%	n/a	n/a
2	0,3605692	70,50%	0,8017030	60,50%	n/a	n/a
3	0,5845597	57,00%	0,6780425	57,00%	n/a	n/a
4	0,5378383	61,00%	0,5875131	60,50%	n/a	n/a
5	0,6309438	66,50%	0,5540928	60,50%	n/a	n/a
6	0,7880373	64,50%	0,6009105	60,50%	n/a	n/a
7	0,3985397	64,00%	0,6358956	61,00%	n/a	n/a
8	1,0824001	61,00%	0,6264278	62,50%	n/a	n/a
9	0,5751281	66,50%	0,6871637	53,00%	n/a	n/a
10	0,7577429	67,00%	0,6929342	66,00%	n/a	n/a
11	0,4490131	67,00%	0,7184790	62,50%	n/a	n/a
12	0,6859714	70,00%	0,5701971	59,50%	n/a	n/a
13	0,7337943	68,00%	0,6845854	65,50%	n/a	n/a
14	0,5363925	57,50%	0,6573285	53,50%	n/a	n/a
15	0,4028288	61,00%	0,5905522	57,50%	n/a	n/a
16	0,8387069	62,50%	0,6831604	58,00%	n/a	n/a
17	0,5550827	59,50%	0,7367245	67,50%	n/a	n/a
18	0,7860168	61,50%	0,8561425	69,00%	n/a	n/a
19	0,9257799	62,00%	0,6045007	64,50%	n/a	n/a
Avg	0,6576220	63,88%	0,6629911	61,50%	0,6957849	60,08%

Source: Produced by the author in August, 2022

Table 133 – Results of ninth experiment on potential domain 6 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6126031	71,00%	0,5172144	63,00%	n/a	n/a
1	0,3137401	73,50%	0,6194582	62,50%	n/a	n/a
2	0,3493060	66,00%	0,6605810	70,50%	n/a	n/a
3	0,8804145	68,00%	0,8535695	65,50%	n/a	n/a
4	0,8684161	67,00%	0,8071794	63,50%	n/a	n/a
5	0,7298279	66,00%	0,4230445	65,00%	n/a	n/a

(Continues...)

Table 133 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	0,9842135	65,50%	0,6804273	66,00%	n/a	n/a
7	0,4342667	65,50%	0,5803673	66,50%	n/a	n/a
8	0,7469499	61,50%	0,4406444	65,00%	n/a	n/a
9	0,5359296	67,50%	0,5621908	67,50%	n/a	n/a
10	0,3873073	68,50%	0,6219937	64,00%	n/a	n/a
11	0,5142288	58,50%	0,6114553	65,50%	n/a	n/a
12	0,5248454	62,50%	1,2301764	63,50%	n/a	n/a
13	0,6400391	56,00%	0,5694990	65,50%	n/a	n/a
14	0,7029328	64,00%	0,3820788	62,00%	n/a	n/a
15	0,7825501	62,00%	0,6990111	64,50%	n/a	n/a
16	0,8768195	65,50%	0,4982986	67,50%	n/a	n/a
17	0,8301160	57,00%	0,4603405	62,00%	n/a	n/a
18	0,9823374	64,50%	0,8330436	64,50%	n/a	n/a
19	0,8174709	57,00%	0,4131385	61,00%	n/a	n/a
Avg	0,6757157	64,35%	0,6231856	64,75%	0,5007331	64,54%

Source: Produced by the author in August, 2022

Table 134 – Results of tenth experiment on potential domain 6 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8901080	60,00%	0,7372680	53,50%	n/a	n/a
1	0,7071466	68,00%	0,3680088	62,50%	n/a	n/a
2	0,4168555	69,50%	0,6154138	63,50%	n/a	n/a
3	0,8712388	72,00%	0,8080435	63,00%	n/a	n/a
4	0,4439666	59,00%	0,7405765	63,00%	n/a	n/a
5	0,7083458	57,50%	0,7001539	62,00%	n/a	n/a
6	0,8302036	60,00%	0,6964174	57,00%	n/a	n/a
7	0,5604048	64,50%	0,7705504	59,00%	n/a	n/a
8	0,6464202	60,00%	0,7949498	65,00%	n/a	n/a
9	0,6304741	62,00%	0,8294318	60,50%	n/a	n/a
10	0,3984973	66,00%	0,6410859	59,50%	n/a	n/a
11	0,7738637	61,00%	0,6887867	65,00%	n/a	n/a
12	1,0275843	55,50%	0,5262761	64,00%	n/a	n/a
13	0,4570334	62,00%	0,4114969	63,50%	n/a	n/a
14	0,4559122	58,00%	0,8962196	68,50%	n/a	n/a
15	0,2734708	65,50%	0,4327303	62,00%	n/a	n/a
16	0,3209937	62,00%	0,6573378	62,50%	n/a	n/a

(Continues...)

Table 134 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
17	0,6723754	62,50%	0,7249564	56,50%	n/a	n/a
18	0,4634355	65,00%	0,7141708	59,50%	n/a	n/a
19	0,5740249	63,50%	0,6929903	65,00%	n/a	n/a
Avg	0,6061178	62,68%	0,6723432	61,75%	0,5938650	62,24%

Source: Produced by the author in August, 2022

Apendix O – Results on potential domain 7 - 2014 and 2015

Table 135 – Results of first experiment on potential domain 7 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6196790	44,00%	0,6435339	41,00%	n/a	n/a
1	0,5780759	51,50%	0,6803525	59,00%	n/a	n/a
2	0,7975525	58,00%	0,5182673	60,00%	n/a	n/a
3	0,8602009	66,50%	0,6884496	57,00%	n/a	n/a
4	0,8007047	57,00%	0,6083031	60,50%	n/a	n/a
5	0,6723741	64,50%	0,5934452	59,00%	n/a	n/a
6	0,6783049	56,50%	0,7229217	45,50%	n/a	n/a
7	0,5648453	52,50%	0,6926930	55,00%	n/a	n/a
8	0,8569554	62,50%	0,6886312	56,00%	n/a	n/a
9	0,4652832	52,00%	0,5819227	58,50%	n/a	n/a
10	0,5195529	64,50%	0,6426705	60,50%	n/a	n/a
11	0,6073488	56,50%	0,6271397	57,50%	n/a	n/a
12	0,9376185	52,50%	0,6929744	62,50%	n/a	n/a
13	0,6439199	60,50%	0,6241916	62,50%	n/a	n/a
14	0,7801437	57,50%	0,7601740	59,00%	n/a	n/a
15	0,8210191	63,50%	0,5305756	62,50%	n/a	n/a
16	0,7215827	60,50%	0,7726904	59,00%	n/a	n/a
17	0,7322347	66,00%	0,6483305	60,00%	n/a	n/a
18	0,9547302	61,50%	0,7398008	59,00%	n/a	n/a
19	0,7374986	60,00%	0,6557816	50,50%	n/a	n/a
Avg	0,7174812	58,40%	0,6556425	57,23%	0,7341412	48,86%

Source: Produced by the author in August, 2022

Table 136 – Results of second experiment on potential domain 7 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,8249387	54,00%	0,5889630	43,00%	n/a	n/a
1	0,7013257	56,50%	0,7527320	43,50%	n/a	n/a
2	0,7451749	63,00%	0,7010270	59,50%	n/a	n/a
3	0,5534296	60,00%	0,5845642	58,50%	n/a	n/a

(Continues...)

Table 136 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
4	0,7759782	63,50%	0,6802483	57,50%	n/a	n/a
5	0,6813340	59,50%	0,6652112	50,00%	n/a	n/a
6	0,4208582	58,00%	0,7864886	64,50%	n/a	n/a
7	0,4070632	62,00%	0,6957048	62,50%	n/a	n/a
8	0,6183183	55,00%	0,7392949	57,00%	n/a	n/a
9	0,5628603	63,00%	0,6747178	50,50%	n/a	n/a
10	0,5144910	60,00%	0,7667958	45,50%	n/a	n/a
11	0,5495548	60,50%	0,7563888	46,00%	n/a	n/a
12	0,4572960	61,00%	0,5771099	52,00%	n/a	n/a
13	0,6058429	63,00%	0,5089349	64,50%	n/a	n/a
14	0,6681480	53,00%	0,6743224	46,50%	n/a	n/a
15	0,6177313	63,50%	0,8520353	48,50%	n/a	n/a
16	0,6589693	66,50%	0,7671908	61,00%	n/a	n/a
17	0,3361549	58,50%	0,6926386	60,00%	n/a	n/a
18	1,0376530	67,00%	0,6799350	57,50%	n/a	n/a
19	0,6838753	61,50%	0,5548597	63,50%	n/a	n/a
Avg	0,6210499	60,45%	0,6849582	54,58%	0,5106040	60,15%

Source: Produced by the author in August, 2022

Table 137 – Results of third experiment on potential domain 7 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7622923	59,50%	0,7150106	59,50%	n/a	n/a
1	0,5478115	56,50%	0,7320689	69,00%	n/a	n/a
2	0,3800925	64,00%	0,3891569	61,00%	n/a	n/a
3	0,6899620	52,50%	0,6973301	60,00%	n/a	n/a
4	0,6790342	56,50%	0,8156008	61,50%	n/a	n/a
5	0,5008473	59,50%	0,5476305	66,00%	n/a	n/a
6	0,6631005	65,00%	0,7969579	59,50%	n/a	n/a
7	0,5227607	64,00%	0,7285219	63,00%	n/a	n/a
8	0,7048466	56,00%	0,5005894	64,50%	n/a	n/a
9	0,5235559	57,00%	0,8801853	56,50%	n/a	n/a
10	0,9983236	58,50%	0,6979033	63,50%	n/a	n/a
11	0,6461894	66,50%	0,6159110	63,00%	n/a	n/a
12	0,7022410	57,00%	0,5152844	63,50%	n/a	n/a
13	0,4912731	57,00%	0,6832335	63,00%	n/a	n/a
14	0,5610978	63,50%	0,4435239	63,50%	n/a	n/a

(Continues...)

Table 137 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
15	0,4439010	65,50%	0,5744860	62,00%	n/a	n/a
16	1,0772057	63,00%	0,5248939	56,50%	n/a	n/a
17	0,7959654	62,00%	0,5159434	62,00%	n/a	n/a
18	0,3164125	62,00%	0,4376527	55,50%	n/a	n/a
19	0,7359957	57,50%	0,5560485	65,50%	n/a	n/a
Avg	0,6371454	60,15%	0,6183966	61,93%	0,7091545	65,74%

Source: Produced by the author in August, 2022

Table 138 – Results of fourth experiment on potential domain 7 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5361582	60,50%	0,5988994	51,00%	n/a	n/a
1	0,7872384	56,00%	0,6826614	40,50%	n/a	n/a
2	0,6521091	63,50%	0,7813191	54,00%	n/a	n/a
3	0,8466010	59,50%	0,4998353	66,00%	n/a	n/a
4	0,8738413	59,00%	0,5718996	61,50%	n/a	n/a
5	0,5604590	58,50%	0,7468811	55,50%	n/a	n/a
6	0,9287012	61,00%	0,5663226	63,50%	n/a	n/a
7	0,4489485	61,00%	0,6199771	63,00%	n/a	n/a
8	0,6944972	57,50%	0,6318709	58,50%	n/a	n/a
9	0,7620893	57,50%	0,6726524	43,50%	n/a	n/a
10	0,6743721	62,50%	0,7184120	58,50%	n/a	n/a
11	0,8571487	59,00%	0,5572104	64,50%	n/a	n/a
12	0,4639167	57,50%	0,5009116	63,50%	n/a	n/a
13	0,6638544	63,00%	0,6471735	60,50%	n/a	n/a
14	0,3818473	57,50%	0,5685847	59,00%	n/a	n/a
15	0,6872597	58,00%	0,6865994	60,00%	n/a	n/a
16	0,9352270	61,50%	0,7158908	58,00%	n/a	n/a
17	0,6275698	55,50%	0,6947232	58,00%	n/a	n/a
18	0,5092643	60,50%	0,6079386	65,50%	n/a	n/a
19	0,6528466	59,50%	0,7478098	52,50%	n/a	n/a
Avg	0,6771975	59,43%	0,6408786	57,85%	0,7229948	56,73%

Source: Produced by the author in August, 2022

Table 139 – Results of fifth experiment on potential domain 7 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6524224	64,50%	0,5733979	59,50%	n/a	n/a
1	0,7516099	58,00%	0,6265921	62,50%	n/a	n/a
2	0,4538236	58,00%	0,5242959	58,50%	n/a	n/a
3	0,6409253	55,00%	0,5193928	60,00%	n/a	n/a
4	0,5188276	70,50%	0,3848493	64,00%	n/a	n/a
5	0,9412068	64,00%	0,8589505	58,50%	n/a	n/a
6	0,9652047	67,00%	0,6348647	54,00%	n/a	n/a
7	1,1289101	59,00%	0,6950222	55,00%	n/a	n/a
8	0,4914122	66,00%	0,7207603	66,00%	n/a	n/a
9	0,9525204	61,50%	0,7013033	55,00%	n/a	n/a
10	0,9535505	59,50%	0,6652142	54,50%	n/a	n/a
11	0,7957485	62,00%	0,7932131	51,00%	n/a	n/a
12	0,6720607	54,50%	0,7199212	47,50%	n/a	n/a
13	0,3803084	64,00%	0,6492025	50,00%	n/a	n/a
14	0,6447166	62,00%	0,6856933	40,00%	n/a	n/a
15	0,5121064	57,50%	0,7629324	42,00%	n/a	n/a
16	0,7384416	65,00%	0,7235795	47,50%	n/a	n/a
17	0,7755933	55,50%	0,5790324	57,00%	n/a	n/a
18	0,6495218	62,50%	0,8413217	35,50%	n/a	n/a
19	1,0948172	56,50%	0,7796339	44,00%	n/a	n/a
Avg	0,7356864	61,13%	0,6719587	53,10%	0,5299214	41,12%

Source: Produced by the author in August, 2022

Table 140 – Results of sixth experiment on potential domain 7 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6644914	59,00%	0,7684850	40,00%	n/a	n/a
1	0,7561013	54,00%	0,7253495	53,00%	n/a	n/a
2	0,8275123	59,50%	0,8391448	47,00%	n/a	n/a
3	0,6218300	59,00%	0,7687104	53,50%	n/a	n/a
4	0,4711812	68,00%	0,5792762	43,50%	n/a	n/a
5	0,8677973	64,50%	0,6968843	51,00%	n/a	n/a
6	0,4457664	61,50%	0,6652843	60,50%	n/a	n/a
7	0,3889502	67,00%	0,6976698	60,00%	n/a	n/a
8	1,0111587	59,50%	0,6512386	52,00%	n/a	n/a
9	0,5881572	62,00%	0,6980969	61,00%	n/a	n/a

(Continues...)

Table 140 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
10	0,5216201	59,50%	0,9050109	41,00%	n/a	n/a
11	0,8089966	57,00%	0,7111908	60,00%	n/a	n/a
12	0,3751476	58,50%	0,6710219	73,50%	n/a	n/a
13	0,6610601	56,00%	0,4982882	60,50%	n/a	n/a
14	1,0442201	57,00%	0,5703093	65,00%	n/a	n/a
15	0,7141300	49,00%	0,4352916	68,00%	n/a	n/a
16	0,6099130	57,00%	0,6620741	63,50%	n/a	n/a
17	0,4413726	68,50%	0,6980696	69,00%	n/a	n/a
18	0,4830561	65,00%	0,7422816	52,50%	n/a	n/a
19	0,8484566	65,00%	0,6987711	64,00%	n/a	n/a
Avg	0,6575459	60,33%	0,6841224	56,93%	0,7803552	66,37%

Source: Produced by the author in August, 2022

Table 141 – Results of seventh experiment on potential domain 7 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,6359471	54,00%	0,6626343	57,50%	n/a	n/a
1	0,7350245	48,00%	0,7646791	51,00%	n/a	n/a
2	0,4732555	61,00%	0,6385208	63,50%	n/a	n/a
3	0,9314934	60,00%	0,7673737	43,50%	n/a	n/a
4	0,2721933	54,00%	0,5888881	56,00%	n/a	n/a
5	0,6880327	59,00%	0,6246305	48,50%	n/a	n/a
6	0,2809762	59,50%	0,5071285	53,50%	n/a	n/a
7	0,8225616	50,50%	0,6017272	61,00%	n/a	n/a
8	0,8951764	56,00%	0,5535906	57,00%	n/a	n/a
9	0,5075949	59,00%	0,6013246	57,00%	n/a	n/a
10	0,7383738	62,00%	0,5882870	64,50%	n/a	n/a
11	0,9139845	56,50%	0,6533269	54,00%	n/a	n/a
12	0,5211426	69,50%	0,6434963	59,00%	n/a	n/a
13	0,7295694	59,50%	0,7103108	59,50%	n/a	n/a
14	0,6410527	62,00%	0,7270905	56,00%	n/a	n/a
15	0,7071083	59,00%	0,6609914	67,50%	n/a	n/a
16	0,5667500	64,00%	0,6436492	67,50%	n/a	n/a
17	1,0243521	64,50%	0,6025180	59,50%	n/a	n/a
18	0,4516331	54,50%	0,7143750	53,00%	n/a	n/a
19	0,6142738	66,00%	0,5519377	61,00%	n/a	n/a
Avg	0,6575248	58,93%	0,6403240	57,50%	0,5849164	62,31%

(Continues...)

Table 141 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc

Source: Produced by the author in August, 2022

Table 142 – Results of eighth experiment on potential domain 7 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7151908	53,00%	0,6581497	48,50%	n/a	n/a
1	0,4763501	54,50%	0,7289334	57,00%	n/a	n/a
2	0,6086662	59,50%	0,6248206	59,50%	n/a	n/a
3	0,7374710	60,50%	0,7117921	64,00%	n/a	n/a
4	0,5389692	65,50%	0,5747364	58,50%	n/a	n/a
5	0,6132793	57,00%	0,6091292	57,00%	n/a	n/a
6	0,6168887	62,50%	0,5338882	70,50%	n/a	n/a
7	0,8707819	62,00%	0,4137278	62,00%	n/a	n/a
8	0,7117091	49,00%	0,6288579	54,50%	n/a	n/a
9	0,8079976	56,00%	0,7111576	54,00%	n/a	n/a
10	0,3928071	63,50%	0,7011300	55,50%	n/a	n/a
11	0,5915295	66,00%	0,7422625	58,00%	n/a	n/a
12	0,6596057	60,00%	0,7071252	66,00%	n/a	n/a
13	0,7291840	55,00%	0,7822946	59,50%	n/a	n/a
14	0,4107148	55,00%	0,5757881	63,00%	n/a	n/a
15	0,3891855	59,00%	0,8017889	50,50%	n/a	n/a
16	0,7750014	58,50%	0,7132087	56,50%	n/a	n/a
17	1,4083824	60,00%	0,7527993	47,50%	n/a	n/a
18	0,8923612	57,50%	0,7548573	57,00%	n/a	n/a
19	0,5285895	69,50%	0,7780005	51,50%	n/a	n/a
Avg	0,6737332	59,18%	0,6752224	57,53%	0,7372152	46,70%

Source: Produced by the author in August, 2022

Table 143 – Results of ninth experiment on potential domain 7 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,9291680	67,50%	0,7230164	53,50%	n/a	n/a
1	0,6498787	59,50%	0,6565542	42,50%	n/a	n/a
2	0,4422827	63,50%	0,6362661	55,00%	n/a	n/a
3	0,5260746	59,00%	0,6702636	53,50%	n/a	n/a
4	0,7904350	54,00%	0,6957647	47,50%	n/a	n/a
5	0,6003104	63,50%	0,5806499	51,50%	n/a	n/a

(Continues...)

Table 143 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	1,2215570	66,50%	0,4906192	55,50%	n/a	n/a
7	0,8458943	61,00%	0,6135394	60,50%	n/a	n/a
8	0,7172669	64,00%	0,6776458	46,00%	n/a	n/a
9	0,5855922	56,00%	0,6981339	50,50%	n/a	n/a
10	0,6834692	64,50%	0,6829195	60,50%	n/a	n/a
11	0,6428280	65,50%	0,5550711	61,50%	n/a	n/a
12	0,8836454	50,00%	0,6648750	49,00%	n/a	n/a
13	0,6752782	59,50%	0,4780872	58,00%	n/a	n/a
14	0,5614532	54,00%	0,8297753	59,50%	n/a	n/a
15	0,5744873	62,50%	0,6485504	58,50%	n/a	n/a
16	0,8677574	54,00%	0,6426549	59,00%	n/a	n/a
17	0,4564136	60,00%	0,6909930	54,50%	n/a	n/a
18	0,9500393	57,00%	0,7317449	55,00%	n/a	n/a
19	0,7163472	60,50%	0,9098735	53,50%	n/a	n/a
Avg	0,7160089	60,10%	0,6638499	54,25%	0,7833300	57,49%

Source: Produced by the author in August, 2022

Table 144 – Results of tenth experiment on potential domain 7 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5144670	66,50%	0,6907099	63,00%	n/a	n/a
1	0,7225024	61,00%	0,7280877	53,50%	n/a	n/a
2	0,8271616	53,00%	0,7790763	58,00%	n/a	n/a
3	0,4738825	61,00%	0,6848681	56,50%	n/a	n/a
4	0,7839405	58,00%	0,7217796	42,50%	n/a	n/a
5	0,5520795	58,00%	0,7644747	43,50%	n/a	n/a
6	0,7205437	62,00%	0,7431949	39,00%	n/a	n/a
7	0,5923576	62,00%	0,7498356	57,50%	n/a	n/a
8	0,6448808	57,50%	0,5684558	53,50%	n/a	n/a
9	0,7986287	62,50%	0,6779395	52,00%	n/a	n/a
10	0,8478677	55,50%	0,8415062	42,50%	n/a	n/a
11	0,9035163	59,00%	0,7115268	49,50%	n/a	n/a
12	0,7308712	59,50%	0,6208916	58,50%	n/a	n/a
13	0,5863137	64,00%	0,7893853	50,50%	n/a	n/a
14	0,7110220	61,00%	0,5990539	44,50%	n/a	n/a
15	1,1595340	59,00%	0,6596065	51,00%	n/a	n/a
16	0,6481535	60,00%	0,7742217	48,50%	n/a	n/a

(Continues...)

Table 144 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
17	0,6763079	59,50%	0,5460344	58,00%	n/a	n/a
18	0,7014744	58,00%	0,7296030	53,50%	n/a	n/a
19	0,5254517	52,00%	0,6118257	44,50%	n/a	n/a
Avg	0,7060478	59,45%	0,6996039	51,00%	0,7952233	46,07%

Source: Produced by the author in August, 2022

Apendix P – Results on potential domain 8 - 2014 and 2015

Table 145 – Results of first experiment on potential domain 8 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,4033060	63,00%	0,5495318	71,50%	n/a	n/a
1	0,5410410	62,50%	0,8006306	63,00%	n/a	n/a
2	0,5465792	69,50%	0,6264195	67,00%	n/a	n/a
3	0,3086825	69,00%	0,8927639	63,50%	n/a	n/a
4	0,7500145	71,50%	0,6164498	60,50%	n/a	n/a
5	1,2142348	67,50%	0,5733029	65,50%	n/a	n/a
6	0,5465008	69,50%	0,7378806	66,50%	n/a	n/a
7	0,6639783	61,50%	0,6554137	69,50%	n/a	n/a
8	0,5968113	75,00%	0,5341752	65,00%	n/a	n/a
9	0,6410602	72,00%	0,6722234	65,00%	n/a	n/a
10	0,6007463	65,50%	0,8595663	48,00%	n/a	n/a
11	0,7930439	67,00%	0,5321320	66,50%	n/a	n/a
12	0,5655696	70,50%	0,7345694	62,00%	n/a	n/a
13	0,4817815	68,50%	0,5685201	64,50%	n/a	n/a
14	0,7877474	64,50%	0,7014320	66,50%	n/a	n/a
15	0,4047492	60,50%	0,4940856	62,50%	n/a	n/a
16	0,3658415	66,50%	0,5897135	69,50%	n/a	n/a
17	1,0295265	68,00%	0,6545506	68,50%	n/a	n/a
18	1,0240409	65,50%	0,7288537	67,00%	n/a	n/a
19	0,8493975	66,00%	0,8547449	60,50%	n/a	n/a
Avg	0,6557327	67,18%	0,6688480	64,63%	0,6203640	59,93%

Source: Produced by the author in August, 2022

Table 146 – Results of second experiment on potential domain 8 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,2960403	69,50%	0,8400398	64,50%	n/a	n/a
1	0,6288987	66,00%	0,6168912	66,00%	n/a	n/a
2	0,8517300	70,00%	0,6447960	64,50%	n/a	n/a
3	0,9776653	59,00%	0,6641707	70,50%	n/a	n/a

(Continues...)

Table 146 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
4	0,9235811	61,50%	0,7519122	66,00%	n/a	n/a
5	0,5550894	60,00%	0,6095126	62,50%	n/a	n/a
6	0,4749856	73,50%	0,6713378	71,00%	n/a	n/a
7	0,4911188	65,50%	0,7156752	65,00%	n/a	n/a
8	0,7729019	64,50%	0,5034283	65,50%	n/a	n/a
9	0,9451569	73,00%	0,6892066	73,00%	n/a	n/a
10	0,3870944	72,00%	0,6156222	72,00%	n/a	n/a
11	0,6711946	72,50%	0,5422515	70,00%	n/a	n/a
12	0,7846873	65,50%	0,7507819	65,00%	n/a	n/a
13	0,6829230	67,50%	0,4956855	62,00%	n/a	n/a
14	0,7392111	64,00%	0,6162949	64,50%	n/a	n/a
15	0,4504746	67,00%	0,5741035	71,00%	n/a	n/a
16	0,6600130	62,00%	0,3108210	70,50%	n/a	n/a
17	0,3986561	65,50%	0,6840960	62,50%	n/a	n/a
18	0,1677900	71,50%	0,4639731	68,00%	n/a	n/a
19	0,6606622	63,50%	0,4168638	71,00%	n/a	n/a
Avg	0,6259937	66,68%	0,6088732	67,25%	0,6790950	66,46%

Source: Produced by the author in August, 2022

Table 147 – Results of third experiment on potential domain 8 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7158381	65,00%	0,6789904	60,00%	n/a	n/a
1	0,5773216	61,00%	0,7510160	60,00%	n/a	n/a
2	0,5255417	65,50%	0,7167777	64,00%	n/a	n/a
3	0,5882113	73,00%	0,6145821	70,00%	n/a	n/a
4	0,4848502	67,00%	0,8122888	53,50%	n/a	n/a
5	0,5653789	63,00%	0,6346262	62,00%	n/a	n/a
6	0,5325626	65,50%	0,7760161	49,00%	n/a	n/a
7	0,5593430	64,00%	0,6577588	57,50%	n/a	n/a
8	0,6967760	68,00%	0,6612684	67,50%	n/a	n/a
9	0,5311853	70,00%	0,7614968	66,00%	n/a	n/a
10	0,8684754	65,00%	0,7294217	59,00%	n/a	n/a
11	0,7415298	69,50%	0,5816804	69,00%	n/a	n/a
12	0,4368145	55,00%	0,5636813	63,50%	n/a	n/a
13	0,3376490	66,50%	0,5128301	67,50%	n/a	n/a
14	0,3834240	65,50%	0,9998993	59,50%	n/a	n/a

(Continues...)

Table 147 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
15	0,7704723	71,00%	0,6100614	63,00%	n/a	n/a
16	0,9185803	70,50%	0,7386339	61,50%	n/a	n/a
17	0,3658237	72,50%	0,6602441	71,00%	n/a	n/a
18	0,4876342	63,50%	0,8281289	67,50%	n/a	n/a
19	0,5820214	68,50%	0,8147917	67,50%	n/a	n/a
Avg	0,5834717	66,48%	0,7052097	62,93%	0,6292350	68,31%

Source: Produced by the author in August, 2022

Table 148 – Results of fourth experiment on potential domain 8 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,3299998	73,00%	0,6053954	68,50%	n/a	n/a
1	0,9961572	71,00%	0,6161884	73,50%	n/a	n/a
2	0,8306130	65,00%	0,7696198	68,00%	n/a	n/a
3	0,2612224	70,00%	0,8364632	71,50%	n/a	n/a
4	0,5369525	73,00%	0,4524910	71,00%	n/a	n/a
5	0,3533853	64,00%	0,5575522	74,00%	n/a	n/a
6	0,7197973	69,00%	0,4826695	75,00%	n/a	n/a
7	0,6768288	67,50%	0,5218706	69,00%	n/a	n/a
8	0,7067426	55,00%	0,6570404	51,50%	n/a	n/a
9	0,1812372	72,50%	0,3660412	68,00%	n/a	n/a
10	0,9072518	68,00%	0,5102309	69,00%	n/a	n/a
11	0,8678520	71,00%	0,5693285	71,50%	n/a	n/a
12	0,3524323	72,00%	0,4231819	73,50%	n/a	n/a
13	0,6715405	69,50%	0,4968095	74,00%	n/a	n/a
14	0,8759791	65,50%	0,7318077	47,00%	n/a	n/a
15	0,2244382	70,00%	0,6384094	59,00%	n/a	n/a
16	0,7362213	65,50%	0,7039578	56,50%	n/a	n/a
17	0,4185079	65,00%	0,8097686	44,50%	n/a	n/a
18	0,3105648	65,50%	0,6041077	62,50%	n/a	n/a
19	0,4312155	63,50%	0,6667808	59,00%	n/a	n/a
Avg	0,5694470	67,78%	0,6009857	65,33%	0,6756247	50,55%

Source: Produced by the author in August, 2022

Table 149 – Results of fifth experiment on potential domain 8 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,3071323	64,50%	0,7318968	64,50%	n/a	n/a
1	0,3374083	73,50%	0,7464466	71,00%	n/a	n/a
2	0,7168659	67,50%	0,7829952	67,00%	n/a	n/a
3	1,5551053	69,50%	0,6390331	69,00%	n/a	n/a
4	0,5042524	51,00%	0,6411601	49,50%	n/a	n/a
5	0,9707248	68,50%	0,5220809	67,50%	n/a	n/a
6	0,2867502	72,00%	0,6916350	58,00%	n/a	n/a
7	0,7739321	64,50%	0,7709440	54,50%	n/a	n/a
8	0,7193533	61,00%	0,7184542	57,50%	n/a	n/a
9	0,6658925	67,50%	0,7603574	56,00%	n/a	n/a
10	0,5780525	60,00%	0,8086318	44,50%	n/a	n/a
11	0,8437322	63,50%	0,7595078	62,00%	n/a	n/a
12	1,0131148	69,50%	0,6804019	54,50%	n/a	n/a
13	0,6467606	70,50%	0,6592487	59,00%	n/a	n/a
14	0,8486910	67,50%	0,6494557	58,50%	n/a	n/a
15	0,4562461	69,00%	0,7090269	66,00%	n/a	n/a
16	1,1722035	64,00%	0,7230356	60,50%	n/a	n/a
17	0,4383879	65,00%	0,6898574	49,00%	n/a	n/a
18	0,3020391	66,50%	0,6878531	55,50%	n/a	n/a
19	0,2002572	72,50%	0,6499017	69,00%	n/a	n/a
Avg	0,6668451	66,38%	0,7010962	59,65%	0,6555846	66,83%

Source: Produced by the author in August, 2022

Table 150 – Results of sixth experiment on potential domain 8 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5506173	65,00%	0,5621713	58,50%	n/a	n/a
1	1,2536818	61,00%	0,7088603	66,50%	n/a	n/a
2	0,5054504	62,00%	0,5825194	67,00%	n/a	n/a
3	0,5914767	63,50%	0,7493116	66,00%	n/a	n/a
4	0,6150782	66,00%	0,6594579	69,00%	n/a	n/a
5	0,4621011	67,00%	0,9419288	67,00%	n/a	n/a
6	0,5878336	64,50%	0,7147812	64,00%	n/a	n/a
7	0,6961234	66,00%	0,6771064	66,50%	n/a	n/a
8	0,2959630	70,50%	0,4436347	64,50%	n/a	n/a
9	0,4323717	68,50%	0,6903402	70,50%	n/a	n/a

(Continues...)

Table 150 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
10	0,8600757	67,50%	0,3739519	67,00%	n/a	n/a
11	0,6445585	68,50%	0,5842018	71,00%	n/a	n/a
12	0,6125357	61,50%	0,7239649	66,00%	n/a	n/a
13	0,8914042	61,00%	0,7509356	60,00%	n/a	n/a
14	0,9494563	64,00%	0,6106066	63,50%	n/a	n/a
15	0,4057557	66,50%	0,6199710	65,00%	n/a	n/a
16	0,6232728	70,00%	0,5134772	72,00%	n/a	n/a
17	0,4335704	68,50%	0,6821452	52,00%	n/a	n/a
18	0,8286791	63,50%	0,7123649	64,00%	n/a	n/a
19	0,9285197	61,50%	0,5596914	65,00%	n/a	n/a
Avg	0,6584263	65,33%	0,6430711	65,25%	0,7236459	64,73%

Source: Produced by the author in August, 2022

Table 151 – Results of seventh experiment on potential domain 8 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,4426667	67,00%	0,5865011	70,00%	n/a	n/a
1	0,8574502	61,50%	0,6775253	68,50%	n/a	n/a
2	0,5701714	60,50%	0,7205003	66,50%	n/a	n/a
3	0,8473248	61,00%	0,6859474	68,50%	n/a	n/a
4	0,9915224	71,50%	0,8246608	66,50%	n/a	n/a
5	0,5433627	62,50%	0,8418049	61,50%	n/a	n/a
6	0,8115389	68,00%	0,7457868	69,50%	n/a	n/a
7	0,3729701	71,50%	0,5611898	72,00%	n/a	n/a
8	0,4323680	66,00%	0,7640715	61,00%	n/a	n/a
9	0,8982376	75,50%	0,5769632	61,50%	n/a	n/a
10	0,5887375	67,00%	0,5406824	55,50%	n/a	n/a
11	0,8677626	70,50%	0,7055794	41,00%	n/a	n/a
12	0,4730888	67,50%	0,7118819	51,50%	n/a	n/a
13	0,6441766	71,00%	0,8362877	33,50%	n/a	n/a
14	0,8582331	64,50%	0,6488450	37,50%	n/a	n/a
15	0,2606927	70,50%	0,5801250	57,00%	n/a	n/a
16	0,5393409	73,00%	0,6952772	58,50%	n/a	n/a
17	0,7487627	65,00%	0,8479924	42,00%	n/a	n/a
18	1,0064282	75,50%	0,6469054	60,50%	n/a	n/a
19	0,4985195	63,50%	0,5829684	50,50%	n/a	n/a
Avg	0,6626678	67,65%	0,6890748	57,65%	0,5070481	50,80%

(Continues...)

Table 151 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc

Source: Produced by the author in August, 2022

Table 152 – Results of eighth experiment on potential domain 8 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,7887801	71,00%	0,7999184	66,50%	n/a	n/a
1	0,6802069	65,50%	0,3163701	69,00%	n/a	n/a
2	0,7432998	66,00%	0,7373675	69,00%	n/a	n/a
3	1,2228558	66,00%	0,6503026	68,00%	n/a	n/a
4	0,6885635	61,50%	0,7706948	72,00%	n/a	n/a
5	0,5942327	63,50%	0,5561990	68,50%	n/a	n/a
6	0,6731048	73,50%	0,7146633	66,00%	n/a	n/a
7	0,6807680	64,50%	0,8460954	63,00%	n/a	n/a
8	0,4769087	73,50%	0,3733057	67,50%	n/a	n/a
9	0,4939470	71,50%	0,9565020	66,00%	n/a	n/a
10	0,4320267	61,50%	0,4037792	66,00%	n/a	n/a
11	0,4086736	72,50%	0,5748473	71,50%	n/a	n/a
12	0,6485979	66,50%	0,4543660	68,50%	n/a	n/a
13	0,5931785	67,00%	0,5810578	63,50%	n/a	n/a
14	0,3931153	64,50%	0,6137956	64,00%	n/a	n/a
15	0,7610404	65,50%	0,6936419	62,50%	n/a	n/a
16	0,4893121	74,00%	0,8742664	63,50%	n/a	n/a
17	0,3777151	69,50%	0,6901016	66,00%	n/a	n/a
18	0,2507912	69,50%	0,6003918	66,00%	n/a	n/a
19	0,6861396	65,00%	0,6087484	67,00%	n/a	n/a
Avg	0,6041629	67,60%	0,6408207	66,70%	0,6875782	67,08%

Source: Produced by the author in August, 2022

Table 153 – Results of ninth experiment on potential domain 8 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5800109	70,00%	0,5049281	69,50%	n/a	n/a
1	0,6182527	69,00%	0,4324563	73,50%	n/a	n/a
2	0,6900635	66,50%	0,6398363	66,00%	n/a	n/a
3	0,2457535	76,50%	0,4931504	68,00%	n/a	n/a
4	0,7802405	66,00%	0,6233856	61,00%	n/a	n/a
5	0,5939870	68,50%	0,7470633	70,00%	n/a	n/a

(Continues...)

Table 153 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
6	0,6982151	68,50%	0,6841951	64,50%	n/a	n/a
7	0,5398566	65,50%	0,6337255	59,00%	n/a	n/a
8	0,6659039	64,50%	0,6301960	66,00%	n/a	n/a
9	0,8177234	70,50%	0,7356967	64,50%	n/a	n/a
10	0,6869767	60,00%	0,6011571	67,50%	n/a	n/a
11	0,7682482	62,00%	0,6223662	65,00%	n/a	n/a
12	1,0689293	71,00%	0,5062124	64,00%	n/a	n/a
13	0,2951501	74,00%	0,6137387	61,50%	n/a	n/a
14	0,6474404	65,50%	0,7503126	60,50%	n/a	n/a
15	0,6921744	56,00%	0,8643947	59,50%	n/a	n/a
16	0,4554631	64,50%	0,6037737	65,00%	n/a	n/a
17	0,4835127	67,50%	0,5095530	61,00%	n/a	n/a
18	0,3600120	73,00%	0,4389952	65,50%	n/a	n/a
19	0,6255059	65,00%	0,7137418	57,50%	n/a	n/a
Avg	0,6156710	67,20%	0,6174439	64,45%	0,6469291	57,83%

Source: Produced by the author in August, 2022

Table 154 – Results of tenth experiment on potential domain 8 - 2014 and 2015

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
0	0,5901419	75,50%	1,1216563	67,00%	n/a	n/a
1	0,3227776	72,50%	0,4930786	63,00%	n/a	n/a
2	0,7320257	70,00%	0,5028061	68,50%	n/a	n/a
3	0,7185819	69,00%	1,1600634	65,00%	n/a	n/a
4	0,6379088	60,50%	0,5871844	60,00%	n/a	n/a
5	0,9062971	68,00%	0,6904380	69,00%	n/a	n/a
6	0,7949355	73,50%	0,8058761	66,00%	n/a	n/a
7	0,5064362	70,00%	0,5374223	65,50%	n/a	n/a
8	0,6430832	68,50%	0,5652454	62,50%	n/a	n/a
9	0,6654111	65,00%	0,6947567	59,00%	n/a	n/a
10	0,6731379	72,50%	0,7025800	54,00%	n/a	n/a
11	0,7491280	65,00%	0,7737234	48,50%	n/a	n/a
12	0,2311844	71,50%	0,5714322	59,50%	n/a	n/a
13	0,8259130	68,00%	0,7574845	43,50%	n/a	n/a
14	0,7655144	77,00%	0,7826258	61,50%	n/a	n/a
15	0,5076597	63,00%	0,8125774	48,00%	n/a	n/a
16	0,2475382	72,00%	0,9254258	61,50%	n/a	n/a

(Continues...)

Table 154 – Conclusion

Epoch	Train loss	Train Acc	Val loss	Val Acc	Test loss	Test Acc
17	0,7442009	69,50%	0,6845181	57,50%	n/a	n/a
18	0,6298629	60,50%	0,6022630	59,00%	n/a	n/a
19	0,5665163	64,50%	0,7697969	61,00%	n/a	n/a
Avg	0,6229127	68,80%	0,7270477	59,98%	0,7488835	63,26%

Source: Produced by the author in August, 2022