Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Inferência de similaridade de sentenças judiciais na Justiça do Trabalho

Guilherme Dantas Bispo

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador
Prof. Dr. Marcelo Ladeira

Brasília
2022

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

Bispo, Guilherme Dantas
BB622i Inferência de similaridade de sentenças judiciais na Justiça do Trabalho / Guilherme Dantas Bispo; orientador Marcelo Ladeira. -- Brasília, 2022.
70 p.

Tese (Doutorado - Mestrado Profissional em Computação Aplicada) -- Universidade de Brasília, 2022.

1. Recuperação de Informação. 2. Similaridade de Documentos Jurídicos. 3. nDCG. I. Ladeira, Marcelo, orient. II. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Inferência de similaridade de sentenças judiciais na Justiça do Trabalho

Guilherme Dantas Bispo

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof. Dr. Marcelo Ladeira (Orientador)
CIC/UnB

Prof. Dr. Rafael Ferreira Leite de Mello
UFRPE

Prof. Dr. Thiago de Paulo Faleiros
CIC/UnB

Prof. Dr. Luis Paulo Faina Garcia
CIC/UnB

Prof. Dr. Marcelo Ladeira
Coordenador do Programa de Pós-graduação em Computação Aplicada

Brasília, 6 de Fevereiro de 2022

Dedicatória

À minha amada esposa que compartilhou comigo todas minhas angustias e sempre esteve ao meu lado me incentivando a seguir em frente. Aos meus queridos filhos que mesmo com pouca idade souberam compreender as minhas ausências em alguns momentos familiares. Aos meus familiares e amigos que sempre me apoiaram. Ao meu orientador pelos ensinamentos e pela compreensão em toda jornada.

Agradecimentos

Primeiramente, a Deus pelo dom da vida e por me permitir a realização desse grande sonho. Aos meus familiares, em especial a minha esposa, por serem grandes incentivadores e por sempre acreditarem no meu potencial. Aos professores, em especial ao meu orientador, por compartilharem seus conhecimentos. Aos colegas de turma e de trabalho que de alguma forma puderam contribuir com o sucesso deste trabalho.

Resumo

Esse trabalho propõe um aperfeiçoamento da funcionalidade de minutar e analisar sentenças do sistema Processo Judicial Eletrônico (PJe) da Justiça do Trabalho permitindo ao magistrado uma pesquisa mais refinada de sentenças similares demonstrando, inclusive, o percentual de similaridade das sentenças encontradas com o processo em questão. Para viabilizar a pesquisa são utilizadas técnicas de mineração de texto para identificar similaridades de sentenças na Justiça do Trabalho. Inicialmente, a performance de modelos induzidos via técnica sintáticas e modelos induzidos via técnicas semânticas é avaliada. Para avaliação são considerados três algoritmos: LDA, Doc2Vec e BM25. Os algoritmos são treinados e avaliados com as sentenças do Tribunal Regional do Trabalho da 10ª Região. Baseado nas métricas de P@K e nDCG, o algoritmo BM25 apresentou o melhor desempenho se comparado aos outros algoritmos de análise sintática e, também, de análise semântica. Para a avaliação, foram escolhidos cinco temas do Direito do Trabalho e para cada tema foram elaboradas duas *queries* de pesquisa. As *queries* foram submetidas aos modelos e posteriormente as primeiras 25 sentenças de maior similaridade encontradas foram avaliadas por especialista de negócio levando em consideração a sua relevância. Nesse experimento o BM25 teve 0.8019 como média para índice nDCG, um resultado quase 20% superior ao segundo colocado (LDA250). Após a avaliação, o modelo que usa o BM25 foi integrado a funcionalidade de minutar e analisar sentenças do Processo Judicial Eletrônico (PJe) permitindo aos usuários identificarem de forma fácil quais são as sentenças similares do caso em questão. A solução proposta é uma alternativa de pesquisa durante a elaboração de uma nova sentença permitindo o reaproveitamento de algum texto já desenvolvido em um caso similar anterior, se assim o magistrado desejar.

Palavras-chave: Recuperação de Informação, Similaridade de Documentos Jurídicos, nDCG

Abstract

This work proposes an improvement in the functionality of drafting and analyzing sentences of the Electronic Judicial Process (PJe) system of the Labor Court, allowing the magistrate a more refined search for similar sentences, even demonstrating the percentage of similarity of the sentences found with the process in question . To make the research feasible, text mining techniques are used to identify similarities of sentences in the Labor Court. Initially, the performance of models induced via syntactic techniques and models induced via semantic techniques is evaluated. Three algorithms are considered for evaluation: LDA, Doc2Vec and BM25. The algorithms are trained and evaluated with the judgments of the Regional Labor Court of the 10th Region. Based on P@K and nDCG metrics, the BM25 algorithm showed the best performance when compared to other parsing and semantic analysis algorithms. For the evaluation, five themes were chosen and for each theme two research queries were elaborated. The queries were submitted to the models and the first 25 sentences with the greatest similarity found were evaluated by a business expert taking into account their relevance. In this experiment, the BM25 had 0.8019 as an average for the nDCG index, a result 20% higher than the second place (LDA250). After the evaluation, the model that uses the BM25 was integrated with the functionality to draft and analyze sentences of the Electronic Judicial Process (PJe) allowing users to easily identify which are the similar sentences in the case in question. The proposed solution is a research alternative during the elaboration of a new sentence, allowing the reuse of some text already developed in a previous similar case, if desired by the magistrate.

Keywords: Information Retrieval, Similarity of Legal Documents, nDCG

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

BM25 *Best Match 25*.

BOW *Bag of Words*.

BPM *Business Process Management*.

CBoW *Continuous Bag-Of-Words*.

CNJ Conselho Nacional da Justiça.

CRISP-DM *Cross Industry Standard Process for Data Mining*.

CSJT Conselho Superior da Justiça do Trabalho.

CTPJE Coordenação Técnica do Processo Judicial Eletrônico.

LDA *Latent Dirichlet Allocation*.

LGPD Lei Geral de Proteção de Dados.

LSI *Latent Semantic Indexing*.

mAP *mean Average Precision*.

nDCG *Normalized Discounted Cumulative Gain*.

NLP Processamento de Linguagem Natural.

PJe Processo Judicial Eletrônico.

SVM *Support Vector Machines*.

TEMAC Teoria de Enfoque Meta Analítico Consolidado.

TF-IDF *Term Frequency - Inverse Document Frequency*.

TRT10 Tribunal Regional do Trabalho da 10ª Região.

TRTs Tribunais Regionais do Trabalho.

TST Tribunal Superior do Trabalho.

TUP Tabelas Unificadas Processuais.

Capítulo 1

Introdução

Nesta seção serão apresentados o contexto do trabalho, a definição do problema abordado, a justificativa e objetivos do projeto, as hipóteses de pesquisa e a contribuição esperada.

1.1 Definição do Problema

Segundo o último relatório da Justiça em Números, produzido pelo Conselho Nacional da Justiça (CNJ), a Justiça do Trabalho no ano de 2021:

- proferiu 2,1 milhões de sentenças em processos do 1º grau;
- teve o ingresso de 3 milhões de processos novos;
- teve o acervo passivo de casos pendentes em torno de 4,6 milhões de processos;

Muitos desses processos tratam de assuntos recorrentes e por muitas vezes trazem temas já pacificados no entendimento jurídico. Os magistrados muitas das vezes lavram sentenças muito similares em casos parecidos [1].

Todos os processos judiciais possuem um ou mais assuntos abordados, esses seguem uma antologia definida pelo Conselho Nacional da Justiça (CNJ) nas Tabelas Unificadas Processuais (TUP). Portanto, uma sentença traz a decisão de pedidos de diversos assuntos.

Os processos seguem um rito processual onde várias etapas devem ser concluídas antes do magistrado proferir essa decisão. Quando todas essas etapas anteriores são finalizadas, o processo é definido como concluso para o magistrado, e, então, começam as atividades para confecção da sentença [2].

Inúmeras sentenças são produzidas diariamente como resultado da análise de processos trabalhistas. Uma das etapas necessárias na confecção de uma nova sentença é a busca por decisões similares tanto de autoria do próprio magistrado quanto de autoria dos demais magistrados buscando entender outros posicionamentos. A funcionalidade de minutar

sentenças do PJe não contempla essas buscas o que dificulta o reaproveitamento de trechos de sentenças já proferidas.

Na área da computação, pode-se considerar que essa problemática é resolvida por soluções de recuperação de informação. Um sistema de recuperação de informação se baseia no uso de técnicas de mineração de texto para encontrar a informação desejada a partir de um texto de busca em uma base de dados [3].

Diante desse cenário fica claro que é possível utilizar técnicas de recuperação da informação para encontrar sentenças similares e foi com esse foco que a pesquisa foi desenvolvida.

1.2 Contextualização

A problemática de inferir a similaridade de sentenças judicias envolve a Justiça do Trabalho, o Processo Judicial Eletrônico (PJe) e o relatório de Justiça em Números produzido pelo Conselho Nacional da Justiça (CNJ). Nessa seção é apresentado uma descrição de cada uma dessas entidades.

1.2.1 Justiça do Trabalho

O Poder Judiciário é dividido em ramos: Justiça Federal, Justiça Comum, Justiça do Trabalho, Justiça Eleitoral e Justiça Militar. A Justiça do Trabalho é responsável por conciliar e julgar as ações judiciais decorrentes da relação de trabalho entre trabalhadores e empregadores, bem como as demandas que tenham origem no cumprimento de suas próprias sentenças, inclusive as coletivas. Os órgãos que compõem a Justiça do Trabalho são: o Tribunal Superior do Trabalho (TST), os Tribunais Regionais do Trabalho (TRTs) e os Juízes do Trabalho. Os Juízes do Trabalho atuam nas Varas do Trabalho e formam o 1º grau da Justiça do Trabalho [4].

O Conselho Superior da Justiça do Trabalho (CSJT) tem como suas responsabilidades: exercer, na forma da lei, a supervisão administrativa, orçamentária, financeira e patrimonial da Justiça do Trabalho de primeiro e segundo graus, como órgão central do sistema, cujas decisões terão efeito vinculante [5]. A Figura 1.1 [1] apresenta a organização hierárquica do Poder Judiciário.

O Conselho Nacional da Justiça (CNJ), com objetivo de divulgar a realidade dos tribunais brasileiros com detalhamento da estrutura e litigiosidade, elabora um relatório estatístico chamado Justiça em Números. Esse relatório é lançado anualmente apresentando indicadores e análises essenciais para subsidiar a Gestão Judiciária brasileira [6].

[1] https://www.cnj.jus.br/primeira-instancia-segunda-instancia-quem-e-quem-na-justica-brasileira/

Figura 1.1: Organograma do Poder Judiciário.

Com base nas informações apresentadas nesse relatório estatístico, é possível perceber a dimensão do Poder Judiciário, com suas despesas e arrecadações.

1.2.2 Processo Judicial Eletrônico (PJe)

O PJe é um sistema computacional cujo objetivo é a unificação de todos os sistemas responsáveis pela tramitação eletrônica dos processos judiciais no Poder Judiciário brasileiro. No que tange a sua implementação na Justiça do Trabalho, todos os órgãos desse ramo passaram a utilizá-lo [7]. O gerenciamento do desenvolvimento do sistema e todo o controle da disponibilização das versões do PJe, na Justiça do Trabalho, ficaram a cargo do Conselho Superior da Justiça do Trabalho (CSJT). Em dezembro de 2012, o PJe alcançou a totalidade das Varas do Trabalho, ou seja, todas as Varas do Trabalho passaram a utilizar o PJe como seu sistema de tramitação processual.

1.2.3 Funcionalidade Minutar e Analisar Sentenças

O PJe é um sistema de acompanhamento processual que possui diversas funcionalidades, dentre ela a funcionalidade de Minutar e Analisar Sentenças. Essa funcionalidade permite aos usuários (servidor ou magistrado) elaborar uma nova sentença para um determinado processo.

A funcionalidade, apresentada na Figura 1.2, é consolidada em apenas uma tela com as seguintes opções:

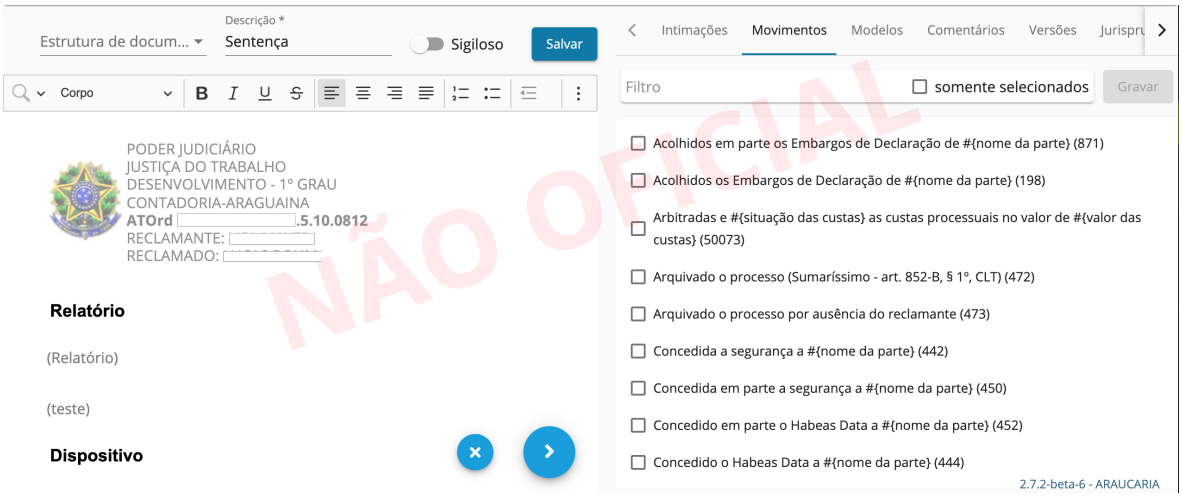


Figura 1.2: Funcionalidade de Analisar e Minutar Sentença.

- Identificação do documento: permite ao usuário definir a estrutura do documento, a descrição e a informação se é sigiloso;
- Editor de texto: permite ao usuário elaborar o texto da sentença conforme a estrutura selecionada na seção de identificação do documento;
- Intimações: permite aos usuários além de confeccionar a sentença, também, expedir expedientes para intimações das partes se for necessário;
- Movimentos: permite aos usuários o lançamento de movimentos processuais quando a sentença for assinada;
- Modelos: permite aos usuários a seleção de algum modelo para auxílio na elaboração da sentença;
- Comentários: exibe os comentários realizados na revisão do documento;
- Versões: apresenta todas as versões salvas da sentença, dessa forma, é possível acompanhar a evolução da sentença.

- Jurisprudência: permite pesquisa na base de jurisprudência do tribunal;

A principal limitação dessa funcionalidade é não permitir aos magistrados buscar por sentenças já proferidas de casos similares. Com essa busca o magistrado poderia buscar argumentações já utilizadas em sentenças anteriores, facilitando a elaboração de uma nova sentença.

1.2.4 Objetivo

Propor um modelo baseado na aplicação de técnicas de mineração de textos para recuperar informações de sentenças judiciais similares no 1º grau da Justiça do Trabalho. Essa busca será baseada em uma pergunta que conterá os principais temas de interesse. As sentenças serão apresentadas de forma ranqueada levando em consideração métricas que utilizam a similaridade baseada nos cossenos da representação vetorial dos documentos e da pergunta.

Os seguintes itens são a definição dos objetivos específicos:

- Comparar os modelos induzidos com base em técnicas sintáticas e em técnicas semânticas para a escolha daquele com melhor performance;
- Avaliar se há variação na performance dos modelos induzidos com a consideração de citações jurídicas;
- Desenvolver uma prova de conceito da aplicação utilizando dados do Tribunal Regional do Trabalho da 10ª Região (Brasília) que integrará o modelo proposto com a funcionalidade de analisar e minutar sentença do PJe.

1.3 Hipótese de Pesquisa

- O uso de técnica de avaliação semântica de texto aumentará a performance dos modelos a serem propostos para a identificação da similaridade entre sentenças judiciais do 1º grau da Justiça do Trabalho comparados ao uso de técnicas de avaliação sintática (léxica);

1.4 Justificativa

Espera-se contribuir com o uso da ferramenta a ser disponibilizada para facilitar encontrar textos de sentenças publicadas de casos similares. O uso da ferramenta tem o potencial de facilitar a redação de sentenças e de contribuir para que casos similares tenham julgamentos similares.

1.5 Contribuição esperada

O foco desse trabalho é apresentar uma solução de aplicação de mineração de textos para a recuperação de informações de sentenças judiciais similares. As sentenças judiciais são documentos jurídicos de alta complexidade que envolvem tanto os fundamentos jurídicos utilizados para embasar a decisão como a própria decisão do magistrado. A maioria das sentenças segue uma estrutura de modelo padrão, porém todos os pedidos levantados na petição inicial do processo trabalhista devem ser contemplados pela sentença, portanto, as sentenças costumam ser documentos de várias páginas decidindo sobre diferentes temas. Além da complexidade inerente das sentenças, as sentenças têm o agravante do entendimento jurídico que muita das vezes pode não ser pacificado o que dificulta ainda mais encontrar padrões.

Portanto, a busca por similaridade de sentenças não é algo trivial, uma vez que são vários fatores a serem analisados, sendo que para a descoberta de muitos desses fatores se faz necessário a busca textual nos documentos e a compreensão de textos jurídicos.

Esse trabalho propõe uma solução que permite uma análise da base de dados do PJe em busca de sentenças similares para auxiliar o magistrado na confecção de novas sentenças. Para essa análise, as sentenças serão representadas por vetores e a similaridade será calculada pelo cosseno desses vetores. Dessa forma, será possível fazer um ranqueamento das sentenças similares permitindo aos magistrados encontrar sentenças já proferidas mais relevantes como auxílio na elaboração de uma nova sentença. Toda a solução está integrada na funcionalidade de Minutar e Analisar Sentenças evitando a necessidade de acessar outra funcionalidade ou sistema contribuindo para a elaboração de sentenças similares de forma mais eficiente.

Capítulo 2

Fundamentação Teórica

Este capítulo descreve a fundamentação teórica apresentando os principais conceitos e a revisão do estado da arte na recuperação de textos e na detecção automática da similaridade entre textos jurídicos.

2.1 Recuperação da Informação

Segundo Baeza-Yates e Ribeiro-Neto [8], a Recuperação da Informação é uma das áreas da Ciência da Computação que tem ganhado bastante importância devido a disseminação da Web e o aumento significativo na quantidade de documentos eletrônicos. Um Sistema de Recuperação de Informação é estruturado basicamente em três etapas:

- Processo de coleta: etapa de coleta dos documentos pode ser na Web ou em um repositório particular;
- Processo de indexação: os documentos coletados passam por vários processamentos e o resultado é armazenado em um repositório central. Esses processamentos têm como objeto gerar índices que otimizem a busca, atualmente, o índice invertido tem sido a principal solução adotada;
- Processo de recuperação e ranqueamento: consiste na busca dos documentos que satisfaçam a consulta do usuário. Os documentos apresentados como resultado dessa busca são ordenados de acordo com sua relevância;

Baeza-Yates e Ribeiro-Neto ainda definem que o principal objetivo de um Sistema de Recuperação de Informação é recuperar todos os documentos relevantes ao usuário em contrapartida a recuperação da menor quantidade possível de documentos irrelevantes.

Para a construção de um modelo de recuperação de informação, os documentos textuais devem adotar uma representação numérica para serem insumos dos algoritmos de

comparação textual. Os modelos clássicos para recuperação da informação podem ser divididos em três categorias: (i) booleano, (ii) vetorial, e (iii) probabilístico.

O modelo booleano é baseado na teoria de conjuntos e na álgebra booleana. Esse modelo tem como vantagens o formalismo e a simplicidade, por essas razões o modelo foi bastante utilizando no passado pelos primeiros sistemas de recuperação de informação. Todavia, sua principal desvantagem é não permitir o ranqueamento dos resultados das pesquisas.

Na busca por superar a limitação do modelo booleano surgiu o modelo vetorial. Esse modelo atribuiu pesos não arbitrários aos termos dos documentos e da consulta permitindo o cálculo de grau de similaridade dos documentos armazenados e a consulta do usuário. Dessa forma, é possível fazer um ranqueamento dos documentos similares através desse grau de similaridade. Os resultados desse modelo são mais precisos do que o modelo booelano.

O modelo probabilístico leva em consideração um esforço inicial para gerar uma definição probabilística preliminar que será utilizada para recuperar um primeiro conjunto ideal de documentos. A partir desse momento, uma interação com o usuário é iniciada a título de refinar a descrição probabilística seguindo a classificação de relevância dos documentos segundo o usuário. A maior vantagem desse modelo é permitir um ranqueamento dos documentos através da probabilidade de relevância desses documentos. As principais desvantagens são: a necessidade de estimar uma resposta ideal inicial; a frequência do termo no documento não é levada em consideração; a falta de normalização de acordo com o tamanho dos documentos.

2.2 *Bag of Words*

O modelo *Bag of Words* (BOW) é uma representação numérica para textos muito utilizada em algoritmos de processamento de linguagem natural e em sistemas de recuperação de informações. Nesse modelo, um texto (como uma sentença ou um documento) é representado por um conjunto de palavras com sua frequência de aparição no texto. Essa solução desconsidera a gramática, a semântica e até mesmo a ordem das palavras no texto. As palavras mais frequentes são consideradas as mais importantes nessa representação.

2.3 *Term Frequency - Inverse Document Frequency*

Considerar simplesmente a frequência das palavras em um determinado texto para definir a sua importância não é uma boa estratégia, uma vez que palavras que aparecem em muitos documentos não contribui para identificar um documento específico. Por essa razão

o modelo *Term Frequency - Inverse Document Frequency* (TF-IDF) utiliza a frequência das palavras (TF) em um documento ponderado pelo inverso da frequência (IDF) dessas palavras em um conjunto de documentos, denominado corpus. Dessa forma é possível identificar a importância de uma palavra do documento mediante os demais documentos.

Na função representativa do modelo TF-IDF [9], o termo d representa um documento do conjunto de documentos definido como D (corpus). A variável t representa o termo a ser comparado. Portanto, a formula pode ser descrita assim:

$$TFIDF(d,t) = tf(d,t) * log\left(\frac{|D|}{df(t)}\right) \tag{2.1}$$

Onde, a função $tf(d,t)$ representa a frequência do termo no documento. Essa função é multiplicada pela função logarítmica responsável por fazer a ponderação do inverso da frequência do termo nos demais documentos.

2.4 *Best Match 25*

O modelo *Best Match 25* (BM25) é uma alternativa para ao uso do tradicional *Term Frequency - Inverse Document Frequency*. Esse modelo probabilístico é bastante utilizado em motores de busca e em ferramentas de pesquisa textual, como Solr e o Elasticsearch.

Este é um modelo probabilístico que tenta estimar a relevância de um documento baseado nas distribuições dos termos das consultas nos documentos relevantes e não-relevantes [10].

Segue a equação Okapi BM25:

$$\sum_{T \in Q} w^{(1)} \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} \tag{2.2}$$

Onde:

- Q é a *query* contendo T termos;
- $w^(1)$ é a função que determina o peso de T em Q;

$$log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \tag{2.3}$$

- N é o número de itens (documentos) da coleção;
- n é o número de documentos contendo o termo;
- R é o número de documentos conhecidos por serem relevantes para o tópico específico;

- r é o número de documentos relevantes contendo o termo;
- K é $k_1((1-b)+b.dl/avdl)$
- k_1, b e k_3 são parâmetros que dependem das naturezas das consultas e possivelmente dos dados; k_1 e b possuem valores padrões de 1.2 e 0.75 respectivamente, mas valores menores de b as vezes é vantajoso; em consultas longas, $k3$ geralmente é definido entre 7 e 1000 (efetivamente infinito);
- tf a frequência de ocorrência do termo dentro de um documento específico;
- qtf é a frequência do termo dentro do tópico do qual Q foi derivado
- dl e $avdl$ são, respectivamente, o comprimento do documento e o comprimento médio do documento medidos em alguma unidade adequada

Soluções de recuperação de informação adotam com frequência a função BM25. Essa função também possui algumas variações que tem sido fruto de estudos na comunidade científica.

2.5 *Latent Dirichlet Allocation*

Latent Dirichlet Allocation (LDA) é um modelo estatístico, especificamente um modelo de tópico, originalmente usado na área de processamento de linguagem natural para representar documentos de texto [11]. A ideia básica do LDA é que um documento possa ser considerado como conjunto limitado de tópicos e cada palavra significativa no documento pode ser associada a um desses tópicos. Dado um conjunto de documentos, o modelo LDA tenta descobrir o seguinte [12]:

- Um conjunto de tópicos;
- As palavras pertencentes a cada tópico;
- Representa os documentos usando os tópicos como uma base vetorial;

Portanto, o documento d é representado por uma distribuição multinomial θ^d sobre tópicos T, e cada tópico $z_j, j = 1...T$ como uma distribuição multinacional ϕ^j sobre o conjunto de palavras W. Para descobrir o conjunto de tópicos usados e a distribuição desses tópicos em cada documento, é necessário estimar θ e ϕ. Um documento pode ser representado, no espaço de tópicos, como uma combinação linear dos tópicos.

2.6 Doc2Vec

O Doc2Vec é um modelo que utiliza a abordagem não supervisionada para a partir das frases, dos parágrafos ou dos documentos construir vetores de representação numérica (representações distribuídas de sentenças e documento) [13].

O algoritmo do Doc2Vec é uma solução ampliada do algoritmo do Word2Vec, sendo que o principal objetivo é construir um espaço vetorial para cada documento. Com essa abordagem os documentos semelhantes geram vetores próximos uns dos outros [13]. O modelo possui uma janela deslizante, de tamanho definido, que utiliza para avaliar a palavra com seu contexto.

Esse modelo contém duas arquiteturas de funcionamento:

- SkipGram: a partir de uma palavra o modelo tenta prever a probabilidade das palavras que aparecem na janela deslizante, ou seja, seu contexto. Essa foi abordagem utilizada nessa pesquisa;
- *Continuous Bag-Of-Words* (CBoW): essa arquitetura tenta achar qual é a palavra faltante, ou seja, a partir da definição das palavras do contexto encontra-se a palavra faltante;

Para gerar um vetor único para representar o documento foi utilizado a média dos vetores de todos os termos.

2.7 Apache Solr

Atualmente, na Justiça do Trabalho, o Solr é a ferramenta utilizada para recuperar o conteúdo de todos os documentos jurídicos, independentemente se esses estão no formato HTML ou PDF. Portanto, o Solr será responsável por alimentar os modelos que forem propostos com os conteúdos das sentenças judiciais e das petições iniciais.

O Solr é um servidor de pesquisa construído sobre o Apache Lucene, uma biblioteca de recuperação de informações de código aberto, baseada na linguagem de programação Java. Ele foi projetado para impulsionar aplicativos poderosos na recuperação de documentos.

O projeto é Open Source utiliza o Lucene Core como base para indexação e busca. Também fornece APIs REST para seu uso. Essa abordagem do uso de API REST permite ao Solr ser integrado a praticamente qualquer linguagem de programação.

O funcionamento do Solr pode ser resumido nas seguintes etapas:

- Definição de um esquema: o esquema informa ao Solr sobre o conteúdo dos documentos que será indexado. O esquema do Solr é poderoso e flexível e permite adaptar o comportamento do Solr ao sistema desejado;

- Indexação dos documentos: alimente o Solr com os documentos pelos quais os usuários desejarão procurar;
- Realização das pesquisas: como o Solr é baseado em padrões abertos, é altamente extensível. As consultas do Solr são simplesmente solicitações HTTP e a resposta é um documento estruturado: principalmente no formato JSON, mas também pode ser XML, CSV ou outros formatos. Isso significa que uma grande variedade de clientes poderá usar o Solr, incluindo qualquer plataforma capaz de realizar requisições HTTP;

A indexação dos documentos segue um fluxo de processamentos que pode ser programável utilizando três componentes básicos:

- Os analisadores de campo são usados durante a indexação ou no momento da consulta. Um analisador examina o texto e gera um fluxo de *tokens*. Os analisadores podem ser de uma única classe ou podem ser compostos por uma série de classes de *tokenizer* e filtro.
- Os *tokenizers* dividem os dados do texto em unidades lexicais ou *tokens* .
- Os filtros examinam um fluxo de *tokens* e os mantêm, transformam ou descartam. *Tokenizer* e filtros podem ser combinados para formar um fluxo, onde a saída de um é inserida no próximo. Essa sequência de *tokenizers* e filtros é chamada de analisador e a saída resultante de um analisador é usada para corresponder aos resultados da consulta ou construir índices.

O grau de satisfação das respostas de uma consulta do usuário é denominado relevância. A relevância de uma resposta da consulta depende do contexto em que a consulta foi executada. O Solr permite a configuração dos algoritmos de busca de similaridades de documentos, mas por padrão o BM25 é adotado. Portanto, a resposta de uma pesquisa é ordenada pela relevância onde é apresentado uma pontuação. Quanto maior for a pontuação mais similar é o documento a pesquisa realizada.

A escolha por analisar a ferramenta Solr foi definida pelo fato da Justiça do Trabalho já ter em seu parque tecnológico soluções que utilizem essa ferramenta. Por essa razão também iremos comparar o resultado retornado por uma pesquisa com o Solr com o resultado retornados pelos modelos que propomos.

2.8 Trabalhos Correlatos

Apesar do foco desse projeto de pesquisa ser em recuperação de informação e não em classificação, foram considerados alguns artigos que evolvem essa abordagem por fazerem análise de documentos jurídicos.

Foi feita uma pesquisa bibliográfica utilizando a Teoria de Enfoque Meta Analítico Consolidado (TEMAC). Essa teoria é fundamentada na identificação de literatura de impacto sobre os temas pesquisados, baseada em técnicas de bibliometria [14]

Foram feitas pesquisas na base SCOPUS (Elieser) utilizando o seguinte termo de busca: *'Text similarity' OR 'Legal Document Similarity' OR 'Legal Information Retrieval' OR 'Similar judgments' OR ('Clustering' AND 'Legal judgments').*

O período da pesquisa adotado foi de 2015 a 2021. Com esses critérios de busca foram encontrados 63 artigos científicos sendo 30 artigos pertencentes a área Computer Science. Por fim, foi feito a leitura do resumo desses artigos selecionando sete artigos por abordarem os seguintes assuntos (critério de inclusão dos artigos):

- Técnicas para encontrar similaridade de documentos em geral;
- Trabalhos relacionados a documentos jurídicos com detalhamento sobre pré-processamento do texto;
- Critérios de avaliação para sistema de recuperação de informação;

Todos os outros artigos que não abordam os critérios de inclusão foram descartados. Utilizou-se, também, pesquisas no Google Scholar para encontrar artigos referenciados nos artigos selecionados.

Aletras [15] faz uma crítica às visões gerais da literatura sobre o tema de classificação de textos. Segundo os autores, a classificação de texto é um processo sofisticado que envolve não apenas o treinamento de modelos, mas também numerosos procedimentos adicionais, por exemplo, o pré-processamento de dados, a transformação e a redução de dimensionalidade. Nesse sentido, para a descoberta de similaridade de sentenças será necessária uma análise mais aprofundada de todos os procedimentos adicionais da classificação de texto. A pesquisa foi realizada com dados dos pedidos realizados no Tribunal Europeu de Direitos Humanos, e a ideia é predizer se algum pedido infringe algum artigo da legislação vigente do país. Esse artigo é bem interessante para atual pesquisa, pois descreve técnicas de pré-processamento, de transformação e de redução de dimensionalidade em documentos jurídicos.

Mironczuk [16] demonstra os avanços significativos no Processamento de Linguagem Natural e Aprendizado de Máquina capazes de construir modelos preditivos que podem ser usados para desvendar padrões que direcionam decisões judiciais. Segundo esse estudo,

os modelos descritos têm em média 79% de precisão na previsão das decisões dos casos julgados pelo Tribunal Europeu dos Direitos Humanos.

Ko [17] apresenta uma comparação entre os paradigmas de aprendizado de máquina. A classificação de texto por aprendizado, geralmente conhecido como aprendizado supervisionado, possui um grande problema que é a necessidade de um grande número de documentos rotulados para o treinamento. Essa tarefa de rotulagem é realizada manualmente por seres humanos, sendo muito onerosa. Em contrapartida, os documentos não rotulados são abundantes e facilmente coletados. Neste trabalho, é proposto um método de classificação de texto baseado na aprendizagem não supervisionada ou semi-supervisionada, sendo que o classificador usa técnicas de *bootstrapping* e projeção de características. Os resultados dos experimentos mostraram que o método proposto obteve desempenho razoavelmente útil comparado a um método supervisionado. Ambas as abordagens podem ser válidas para a comparação de sentenças.

Shmueli [18] define que a análise preditiva inclui métodos empíricos (estatísticos e outros) que resultam em previsões de dados. Segundo os autores são seis os papéis para a análise preditiva: geração de nova teoria, desenvolvimento de medições, comparação de teorias concorrentes, melhoria de modelos existentes, avaliação de relevância e avaliação da previsibilidade de fenômenos empíricos. Apesar da análise preditiva não ser o foco desse projeto, a argumentação e a comparação de modelos podem ser bem úteis na definição da estratégia de comparação das sentenças.

Uma aplicação prática de descoberta de tópicos em documentos pode ser vista em [19]. Segundo Larsen, o uso de *clustering* para descoberta em grande escala de tópicos de um texto é uma ferramenta muito poderosa. A técnica envolve basicamente duas fases: primeiro, mapear as características de cada documento em um espaço dimensional e, em seguida, utilizar algoritmos de agrupamento para definir os pontos de hierarquia do *clusters*. Dessa forma é possível classificar os documentos a partir de cada *cluster*.

Além dos artigos de classificação, também foram encontrados trabalhos correlatos que utilizam a abordagem de recuperação de informação baseada em perguntas (*query*) para encontrar documentos similares.

Barco Ranera [20] apresenta uma solução para encontrar decisões semelhantes nos casos da Suprema Corte das Filipinas como alternativa para o problema de acúmulo de processos judiciais nessa corte. Foram implementados o Doc2Vec e a similaridade de cosseno das representações vetoriais dos documentos. O modelo mostra ter uma precisão de 80% e apresenta uma forte correlação positiva com as pontuações de similaridade de um especialista do domínio jurídico. Esse artigo é bem relevante para pesquisa pois apresentada uma problemática semelhante e o uso de um dos algoritmos testado.

Novotná [21] também apresenta uma solução para recuperação de decisões judiciais

que tratam de uma questão legal semelhante na Suprema Corte Tcheca. Apesar dos métodos de processamento de linguagem natural atualmente disponíveis, essa pesquisa jurídica ainda é feita principalmente por meio de buscas booleanas ou por recuperação textual. Neste estudo, é verificado experimentalmente se o método Doc2vec, juntamente com a similaridade de cosseno, pode recuperar automaticamente decisões semelhantes.

Kim [22] propõe um sistema de pergunta/resposta de conteúdo jurídico vinculando as respostas com artigos do direito civil do Japão. A solução utiliza três abordagens: um modelo não supervisionado (representação TF-IDF com LDA); um modelo supervisionado - *Support Vector Machines* (SVM); um modelo híbrido (combinado técnicas de análise sintática/semântica com o modelo não supervisionado SVM). Para avaliação da acurácia dos modelos é utilizado a média da Precisão Média - *mean Average Precision* (mAP). Implicação para pesquisa é o uso do modelo não supervisionado.

Raghav [23] propõe uma solução de identificação de semelhança de julgamento da Supremo Tribunal da Índia baseada na análise das citações dos julgamentos. Utilizou-se de técnicas de *Bag of Word* com TF-IDF e do algoritmo de *K-Means*. Para avaliação foram utilizados os índices de precisão, de *recall* e *F-Measure*. Implicação para pesquisa é o uso de citações jurídicas para identificar a semelhança de julgamentos. Acredita-se que essa técnica pode melhorar os resultados da pesquisa a ser desenvolvida.

Mandal [24] propõe uma metodologia de cálculo de similaridade entre dois documentos legais. Foi utilizada a representação TF-IDF com modelo de tópico (LDA) e modelos de redes neurais (Word2Vec e Doc2Vec). Implicação para a pesquisa é o uso de modelos de redes neurais (Word2Vec e Doc2Vec) para calcular a similaridade de dois documentos legais.

Wagh [25] apresenta uma análise de rede para comparar duas abordagens para encontrar similaridade de documentos jurídicos: similaridade de cosseno e similaridade baseada em citações. A implicação para pesquisa é o reforço da similaridade baseado em citações.

Mathai [26] apresenta técnicas de Processamento de Linguagem Natural (NLP) e de mineração de dados para comparar julgamentos. Utiliza as representações SVM, *Latent Semantic Indexing* (LSI) e similaridade de cossenos. A proposta é extrair os principais conceitos dos julgamentos jurídicos por meio do método iterativo do LSI e compará-lo com a ferramenta de resumo e com as notas de cabeçalho existentes no processo. A implicação para pesquisa é a abordagem de tentar sumarizar uma sentença é uma técnica que pode ser válida na tentativa de identificar os principais temas do julgamento.

Kumar [27] propõe abordagens para encontrar julgamentos jurídicos semelhantes, estendendo as técnicas populares usadas na recuperação da informação e nos motores de busca. Os julgamentos jurídicos são de natureza complexa e referem-se a outros julgamentos. Foram analisados todos os termos, termos legais, cocitação e métodos de similaridade

baseados em acoplamento bibliográfico para encontrar julgamentos semelhantes. Os resultados experimentais mostraram que o método de similaridade de cosseno de termo legal funciona melhor do que o método de similaridade de cosseno de todos os termos.

No artigo de Xue [28] é proposto uma solução de recuperação de texto baseado no LDA e Word2Vec. O algoritmo proposto calcula a distância entre documento e tópicos, e então cada documento é representado como um vetor de características, em que cada dimensão denota a distância entre este documento e um tópico específico. Para testar a eficácia do algoritmo proposto, vários métodos relacionados são feitos de comparação de desempenho. O uso do LDA e do Word2Vec é bastante interessante para o estudo proposto.

Conforme os trabalhos correlatos, a presente pesquisa também utilizará de abordagens de recuperação de informação para recuperar as sentenças similares baseadas em um texto de busca. Para análise da similaridade será utilizada a diferença de cosseno entre as representações vetoriais das sentenças com a representação vetorial do texto de busca. Diferentemente dos trabalhos encontrados, essa pesquisa comparará três diferentes algoritmos: *Latent Dirichlet Allocation* (LDA), *Best Match 25* (BM25) e Doc2Vec.

A escolha dos algoritmos LDA e Doc2Vec foi baseada nos resultados apresentados em alguns dos trabalhos correlatos, nesses trabalhos esses algoritmos apresentaram uma boa performance na busca por similaridades. Já o algoritmo BM25 foi escolhido pelo fato de ser utilizado pelo banco textual Solr, esse banco textual já é utilizado na infraestrutura da Justiça do Trabalho.

Os algoritmos serão integrados a funcionalidade do PJe para ajudar os magistrados e os servidores a encontrarem casos similares de sentenças judiciais já proferidas com o intuito de apresentar a argumentação da fundamentação dessas sentenças para auxiliar na argumentação da nova sentença.

Capítulo 3

Método

Este capítulo detalha a metodologia utilizada, descreve os experimentos exploratórios realizados e os resultados preliminares obtidos.

3.1 Modelo de referência

Com o conhecimento de trabalhos correlatos, foi definida que as ações para atingir os objetivos se basearão no modelo *Cross Industry Standard Process for Data Mining* (CRISP-DM) [29]. Segundo esse modelo o processo de mineração de dados segue as seguintes fases: entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação e entrega. Portanto, a metodologia desse trabalho seguirá as etapas vislumbradas na Figura 3.1.

Figura 3.1: Adaptação do modelo CRISP-DM.

A primeira etapa é a fase de entendimento, onde os dados e o negócio devem ser compreendidos para que as informações sejam extraídas com a maior qualidade possível.

Essa fase inicial concentra-se no entendimento dos objetivos e dos requisitos sob uma perspectiva de mineração de dados. Após esse entendimento, será possível definir o problema de mineração de dados e o desenvolvimento de um plano preliminar do projeto a ser desenvolvido. A fase de entendimento dos dados começa com uma coleta inicial de dados e prossegue com as atividades para se familiarizar com os dados, identificar problemas de qualidade dos dados, descobrir as primeiras ideias sobre os dados ou detectar subconjuntos interessantes para formar hipóteses de informações ocultas. A formulação do problema de mineração de dados e o plano do projeto requerem pelo menos alguma compreensão dos dados disponíveis [29].

A etapa de preparação contempla as atividades para construir a partir dos dados originários um conjunto de dados que serão disponibilizados ao modelo desenvolvido. É provável que as tarefas de preparação de dados sejam executadas várias vezes até chegar em um conjunto de dados interessante para submissão ao modelo. As tarefas incluem seleção de tabela, registro e atributo, limpeza de dados, construção de novos atributos e transformação de dados para ferramentas de modelagem. Na modelagem, várias técnicas de modelagem são selecionadas e aplicadas, e seus parâmetros são calibrados para valores ideais. Algumas técnicas requerem formatos de dados específicos o que justificaria a criação de novos atributos. Frequentemente, percebe-se problemas de dados durante a modelagem ou obtém-se ideias para construir novos dados, por essa razão essas etapas são executadas de forma de iterativa, onde uma etapa oferece um refinamento para as demais ou até mesmo a necessidade de executar a etapa anterior de forma diferente.

A etapa de avaliação, como o próprio nome sugere, tem como objetivo avaliar o modelo construído sob uma perspectiva de análise de dados. Antes de seguir para a implantação final do modelo é fundamental uma avaliação detalhada do modelo revisando as etapas de construção do modelo, para garantir que os objetivos do negócio foram atingidos. No final desta fase, os resultados gerados devem ter sido apreciados para garantir a eficácia do modelo.

A última etapa é a entrega dos resultados aos usuários finais. A criação do modelo geralmente não é o fim do projeto. Normalmente, o conhecimento adquirido precisará ser organizado e apresentado de forma que o cliente possa usá-lo. Dependendo dos requisitos, a fase de implantação pode ser tão simples quanto gerar um relatório ou tão complexa quanto implementar um processo repetido de mineração de dados. É importante entender de antemão quais ações precisarão ser realizadas para realmente fazer uso dos modelos criados.

3.1.1 Entendimento do Negócio e dos Dados

A sentença judicial é o ato proferido por um magistrado para sanar o conflito de algum processo judicial. Esse ato é materializado em forma de um documento escrito o qual costuma contemplar as seguintes seções:

- Cabeçalho: contém a identificação do processo, como por exemplo classe judicial e assuntos, e também a identificação dos envolvidos no processo, como nome das pessoas e a definição da participação no processo (por exemplo, reclamante ou reclamado);
- Relatório: essa seção apresenta um resumo dos pedidos realizados na petição inicial, juntamente com todas as contestações apresentadas durante o rito do processo;
- Fundamento ou Fundamentação: apresenta a análise jurídica das questões de fato e de direito. Nessa seção, os fundamentos lógicos que explicam a tomada de decisão são apresentados;
- Dispositivo (Conclusão): contém a decisão do magistrado sobre as questões levantadas no processo;

O processo judicial trabalhista segue um rito processual onde varias etapas devem ser vencidas antes da confecção da sentença. Após essas etapas o processo é definido como concluso para sentença, ou seja, todas as atividades anteriores a sentença foram finalizadas e o juiz pode analisar os seus resultados para proferir sua decisão. Quando o processo está concluso para sentença, o juiz do trabalho realiza resumidamente as seguintes ações para construir sua decisão:

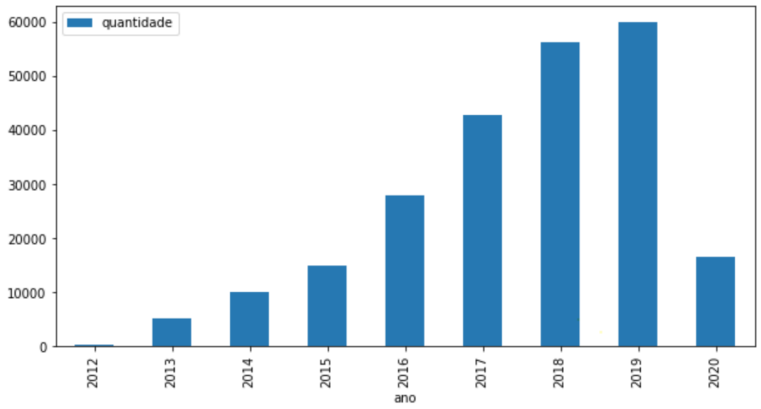
- Realiza a identificação dos assunto e pedidos abordados na petição inicial;
- Se houver contestação, ou seja, peça judicial em que uma das partes apresenta sua defesa ou contraponto, o magistrado realiza o cotejo (confronto) dos pedidos com a contestação apresentada. Dos temas contestados, o juiz examina o conjunto de provas para formar sua convicção;
- Se não houver contestação, o magistrado concede a razão ao reclamante;
- Para fundamentar a decisão, o magistrado busca por suas sentenças anteriores proferidas sobre os determinados assuntos ou pedidos constantes na petição inicial;
- Realiza uma busca no Processo Judicial Eletrônico (PJe) para identificar o entendimento das demais Varas do Trabalho sobre os assuntos constantes na petição inicial;

O modelo a ser proposto contribuirá na realização das duas últimas etapas. A prova de conceito será realizada com a sentenças do Tribunal Regional do Trabalho da 10ª Região. Essa região foi escolhida devida a contribuição de um magistrado com a pesquisa e devido a proximidade geográfica uma vez que os Estados do Distrito Federal e Tocantins são contemplados por essa região.

Os dados foram extraídos do Processo Judicial Eletrônico (PJe), através do banco textual do Solr, no intervalo de abril de 2012 até abril de 2020. O Solr contém o conteúdo textual de todas as peças processuais do PJe. Essa base textual é alimentada através de uma solução de mensageria que realiza a indexação do conteúdo textual das peças processuais em três diferentes momentos:

1. Protocolo do processo: momento onde o advogado preenche todas as informações processuais e submete ao sistema para geração de um número processual. A principal peça processual nesse momento é a petição inicial;
2. Assinatura do documento: durante a movimentação processual é necessário a produção de algumas peças processuais que para serem anexas ao processo é necessário a assinatura. Nesse momento existem diversas peças processuais que podem ser anexas ao processo, porém destacamos as sentenças processuais por ser objeto de pesquisa desse trabalho;
3. Remessa processual: é o recebimento de um processo de uma instância inferior em uma instância superior;

A Tabela 3.1 ilustra as informações contempladas nos dados.



ano	quantidade
2012	248
2013	5210
2014	9951
2015	14932
2016	27878
2017	42635
2018	56235
2019	60004
2020	16502

Figura 3.2: Distribuição da Quantidade de Sentenças por Ano.

Os dados totalizam 233.595 sentenças disponíveis para a construção da prova de conceito. Analisando a distribuição das sentenças ao longo dos anos, é possível perceber

Tabela 3.1: Informações Contempladas.

Descrição	Formato
Identificação do documento	Valor numérico incrementado sequencialmente
Identificação do arquivo binário do documento	Valor numérico (preenchimento opcional)
Código hash do documento	Valor alfanumérico
Extensão do documento	Valor alfanumérico (por exemplo, PDF ou HTML)
Breve descrição informada pelo usuário	Valor alfanumérico
Identificação do tipo do documento	Valor numérico
Código do tipo do documento	Valor alfanumérico
Descrição do tipo do documento	Valor alfanumérico
Nome do responsável pela assinatura	Valor alfanumérico
Data da assinatura do documento	Data
Conteúdo do documento	Texto
Situação de processamento do documento	Valor alfanumérico (valor opcional)
Código do tribunal	Valor alfanumérico (por exemplo, TRT10)
Instância onde o documento foi produzido	Valor numérico (1, 2, 3)
Número do processo formatado	Valor alfanumérico
Identificação do processo	Valor numérico
Identificação do órgão julgador	Valor numérico
Descrição do órgão julgador	Valor alfanumérico
Identificação do órgão julgador colegiado	Valor numérico
Descrição do órgão julgador colegiado	Valor alfanumérico
Identificação se o documento é sigiloso	Valor booleano
Identificação se o processo corre em segredo de justiça	Valor booleano
Identificação do assunto principal do processo	Valor numérico
Descrição do assunto principal do processo	Valor alfanumérico
Descrição da hierarquia dos assuntos	Valor alfanumérico

um aumento gradativo da quantidade de sentenças publicadas Figura 3.2. Isso ocorre por duas razões: (i) os dados foram extraídos do Processo Judicial Eletrônico (PJe) e a instalação desse sistema ocorreu no ano de 2012 nesse regional, por essa razão o número muito baixo de sentenças nesse ano; (ii) o aumento na produtividade dos juízes que, consequentemente, acarretou um aumento nas sentenças publicadas. Com o passar dos anos o PJe foi ganhando força até ser o único sistema de acompanhamento processual de todas a varas do trabalho desse regional;

É importante ressaltar que no ano 2020 só estão sendo contabilizados os meses de janeiro a abril desse ano, provavelmente por essa razão o gráfico não seguiu a tendência de crescimento nesse ano. Com intuito de coletar todos os dados do ano de 2021, o autor fez uma solicitação junto a Coordenação Técnica do Processo Judicial Eletrônico (CTPJE), porém o pedido não foi atendido com a justificativa da mudança de lotação do autor e, também, pela vigência da Lei Geral de Proteção de Dados (LGPD).

A Figura 3.3 apresenta a quantidade de sentenças emitidas por órgão julgador. Observa-se que a grande maioria dos órgãos julgadores publicaram de 5.000 a 10.000 sentenças levando em consideração todo o período dos dados.

Os assuntos judiciais são organizados em uma estrutura hierárquica mantida pelo Conselho Nacional da Justiça (CNJ), essa estrutura é uma das informações padronizadas pelas Tabelas Processuais Unificadas. Toda ação judicial trabalhista deve conter um ou mais assunto judicial. Analisando os dados sob a ótica do assunto principal das ações, é possível perceber que o assunto 'Aviso Prévio' destoa muito com relação aos demais

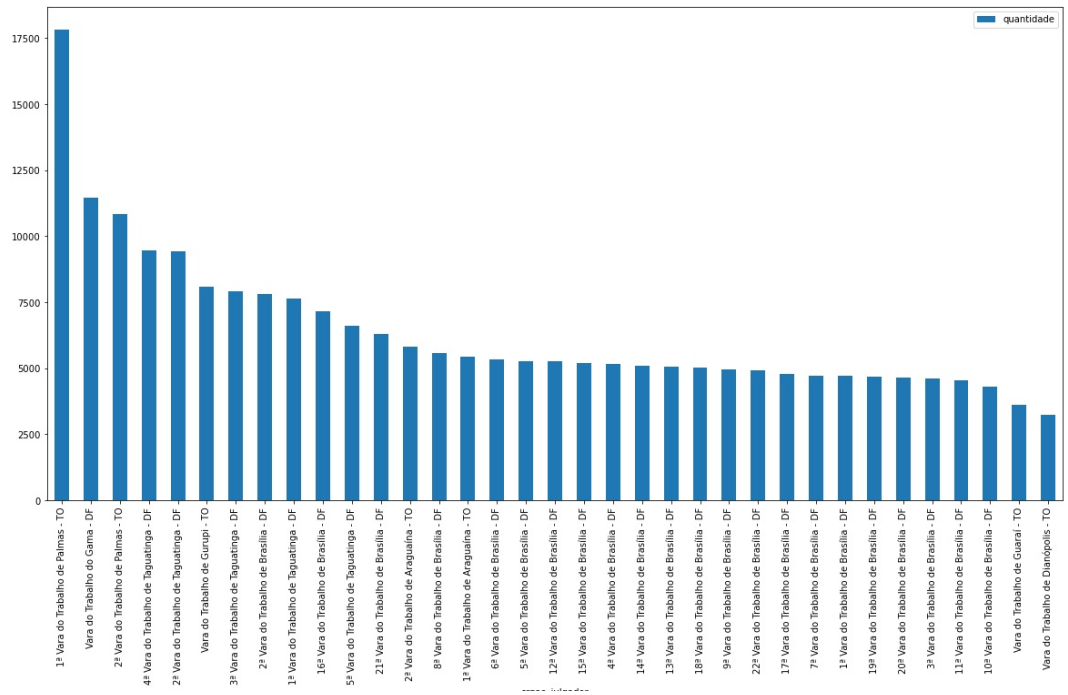


Figura 3.3: Distribuição da Quantidade de Sentenças por Órgão julgador.

assuntos, chegando a quase 23% dos processos desse período dos dados, ou seja, do total de 233.595 processos 52.125 aborda 'Aviso Prévio' conforme demonstrado na Figura 3.4.

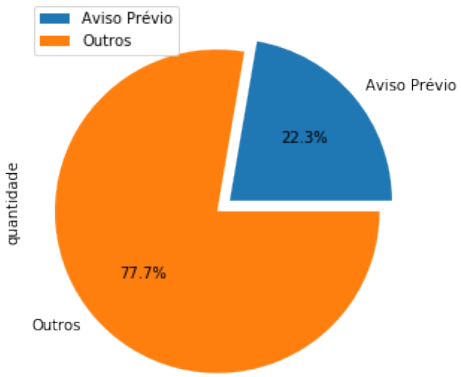


Figura 3.4: Percentual de Processos com o Assunto 'Aviso Prévio'.

Foi estipulado um corte dos assuntos que tem baixa frequência nessa base de dados, com a intenção de diminuir a quantidade de assuntos a serem analisados. Os assuntos que tiveram uma ocorrência inferior a duzentos processos em todo o período foram descartados, essa quantidade representa menos de uma ocorrência do assunto em algum processo por mês em cada vara de trabalho.

Após a realização do corte, a base de dados ficou com um total de 161.881 sentenças abordando 132 assuntos a serem analisados. Isso representa um total de 90% da base de dados original.

Com essa análise preliminar dos dados percebeu-se uma distribuição mais uniforme com relação aos órgãos julgadores, um aumento gradativo com relação aos anos e uma predominância de um assunto específico.

3.1.2 Preparação e Modelagem

Para viabilizar a criação do modelo, é necessário a preparação dos dados e o treinamento dos modelos. A Figura 3.5 apresenta as principais etapas que serão executadas para preparar os dados.

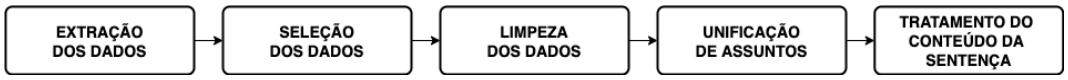


Figura 3.5: Etapas de Preparação dos Dados.

A primeira etapa da preparação dos dados é a extração dos dados. Essa etapa consiste em extrair as sentenças judiciais do PJe, considerando apenas a base de dados do Tribunal Regional do Trabalho da 10ª Região. Para essa extração é necessária a conexão com o banco de dados Postgres do PJe do TRT10.

Após a extração é feito a seleção apenas das sentenças proferidas por algum magistrado no primeiro grau, ou seja, o órgão julgador deve ser as varas do trabalho desse regional. Os campos desejados sobre as sentenças selecionadas são: nome do magistrado, data da publicação, texto da sentença em formato HTML, número do processo, órgão julgador, assunto principal do processo e assunto completo levando em consideração a hierarquia. É importante ressaltar que todas as sentenças judiciais do PJe estão no formato HTML, porém a solução proposta também terá que analisar os temas abordados nas petições iniciais que podem estar no formato PDF. Nesse caso será usado a ferramenta Tesseract para extrair o conteúdo desses arquivos.

Com todas as sentenças selecionadas, então foi necessário remover aquelas que não apresentam valor em algum dos campos, uma vez que todos os campos são obrigatórios. Também, foram removidas as sentenças produzidas, aparentemente, por unidades que não são órgãos julgadores, ou seja, não possuem a prerrogativa para publicar sentenças:

- CEJUSC-JT-TAGUATINGA;
- CONTADORIA-ARAGUAINA;
- CONTADORIA-PALMAS;

- CEJUSC-JT-BRASILIA;
- Coordenadoria de Apoio ao Juízo de Execuções e ao Juízo da Infância e da Juventude;
- Coordenadoria de Apoio ao Juízo de Execuções e ao Juízo da Infância e da Juventude;
- CONTADORIA-TAGUATINGA;
- SEÇÃO DE PRECATÓRIOS;
- CONTADORIA-BSB/GAMA

Para finalizar a etapa de limpeza dos dados, foram removidas as sentenças proferidas por um determinado magistrado, pois esse possui um padrão de produção muito superior aos demais magistrados.

Durante a análise preliminar foi identificada a existência de dois assuntos aparentemente distintos que têm o mesmo significado ('Adicional de Hora Extra' e 'Adicional de Horas Extras'), por essa razão esses assuntos foram unificados. Todas essas estratégias de limpeza e de unificação foram validadas com o magistrado que apoia essa pesquisa.

A última etapa da preparação dos dados é o tratamento do conteúdo da sentença. Essa etapa requer basicamente três passos: (i) conversão do conteúdo, que está em HTML, para texto padrão; (ii) identificação das seções da sentença: cabeçalho, relatório, fundamentação e dispositivo (conclusão); (iii) dividir os textos das seções em parágrafos, pois são esses que serão comparados com os termos de busca.

Com os dados tratados já é possível construir dicionários com representações numéricas desses documentos, para submetê-los ao treinamento dos modelos. A partir dos dados coletados serão induzidos modelos baseados em: BM25, LDA e Doc2Vec. A ideia é fazer uma comparação da performance desses modelos para identificar similaridades de sentenças judiciais. Só será levado em consideração para o treinamento dos modelos a seção de fundamentação das sentenças judiciais. Essa estratégia visa diminuir a quantidade de texto a ser analisado, e por conseguinte diminui as dimensões possíveis na representação numérica do texto.

Uma vez que os modelos estejam treinados e ajustados, o PJe poderá fazer uso desses modelos. A Figura 3.7 ilustra a integração do modelo que for selecionado com a versão proposta para a facilidade Minutar e Analisar Sentença do PJe.

A modelagem definida possui quatro etapas que podem ser executadas de forma cíclica Figura 3.6: edição de nova sentença, montagem do texto de busca, cálculo de similaridade e classificação de resultados. Segue o detalhamento de cada etapa:

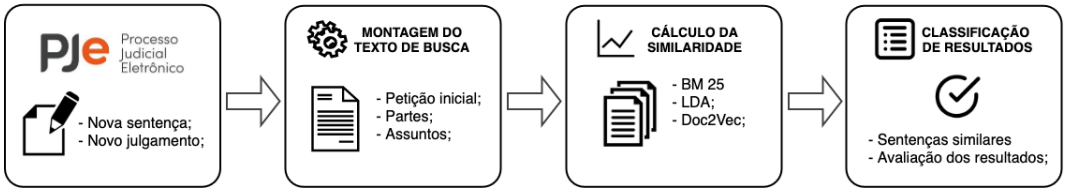


Figura 3.6: Modelagem da solução.

- Edição de uma nova sentença: o magistrado utilizando o PJe acessará a funcionalidade já existente de Minutar e Analisar Sentença para determinado processo judicial. Será disponibilizado nessa tela um mecanismo de busca por similaridade de sentenças já publicadas utilizando as informações do processo judicial em questão. Essa é a única etapa realizada dentro do PJe, as demais farão parte do módulo a ser desenvolvido denominado como Analisador de Sentenças.

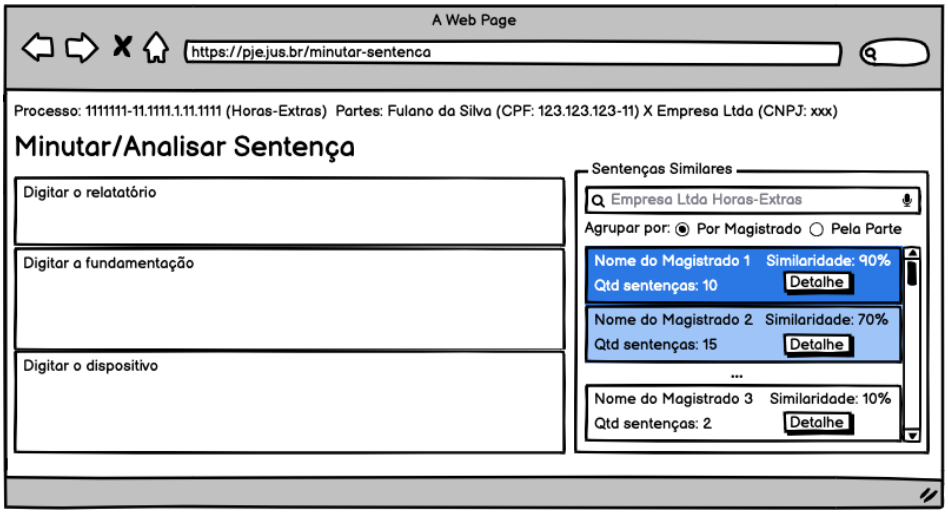


Figura 3.7: Protótipo do Minutar Sentença proposto.

A solução é acrescentar uma funcionalidade na tela de Minutar Sentença que permitirá ao magistrado realizar a busca por sentenças similares. Inicialmente o campo do texto de busca será preenchido com os assuntos e a parte do polo passivo (reclamado) do processo em análise. O resultado poderá ser agrupado por magistrado ou pela parte do polo passivo. Cada agrupamento de resultado apresentará a quantidade de sentenças e o percentual máximo de similaridade encontrado. A ordenação de apresentação dos agrupamentos respeitará a ordem decrescente do percentual máximo de similaridade, também será utilizado uma escala de cores para realçar os resultados mais similares. Ao expandir o agrupamento o magistrado terá a visão detalhada de cada sentença conforme o protótipo ilustrado na Figura 3.8.

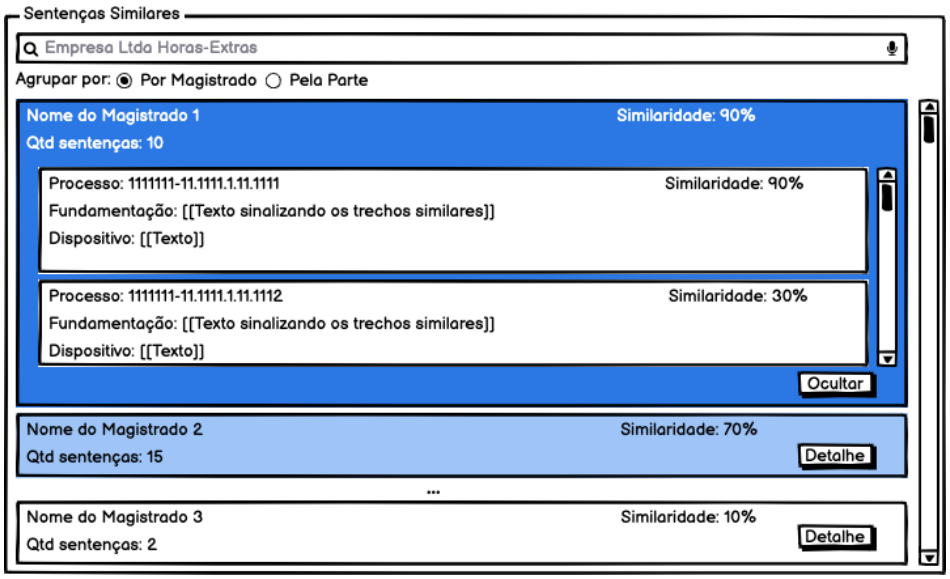


Figura 3.8: Protótipo do Agrupamento Expandido.

- Montagem do texto de busca: a primeira responsabilidade do módulo Analisador de Sentença é analisar a petição inicial, as partes e os assuntos informados no processo para definir um texto de busca. O texto de busca será composto pela identificação da parte que atua no polo passivo e todos os assuntos abordados na petição inicial;
- Cálculo de similaridade: o texto de busca será submetido aos modelos com intuito de encontrar sentenças similares a esse texto, portanto, a comparação será feita do texto de busca com as sentenças. Toda a comparação só levará em consideração os parágrafos da seção de fundamentação da sentença. Para cada parágrafo comparado será obtido uma pontuação de similaridade, denominada score. A pontuação do documento será a máxima pontuação dos seus parágrafos. A pontuação terá um valor entre 0 e 1, sendo 0 a indicação que não há similaridade e 1 a indicação que os documentos são totalmente similares.
- Classificação dos resultados: as sentenças encontradas serão ordenadas pela sua pontuação de forma decrescente, montando assim uma classificação de similaridade de sentenças, onde a primeira sentença terá a pontuação mais elevada. Essa classificação será apresentada na tela de Minutar e Analisar Sentença do PJe. O magistrado poderá sinalizar se as sentenças apresentadas na classificação são similares ou não ao caso em questão. Essa informação poderá ser utilizada para um enriquecimento do modelo.

Uma visão da arquitetura do sistema proposto está apresentada na Figura 3.9. Segue uma breve descrição das responsabilidades de cada camada:

- Camada *Frontend*: contém os componentes visuais de tela que serão executados pelo navegador do usuário. Nessa camada que foi acrescentado uma nova aba (Sentenças Similares) na funcionalidade Minutar e Analisar Sentenças do PJe. As principais linguagens utilizada nessa camada são: Angular, JavaScript, HTML e CSS.
- Camada *Backend*: contém todas as regras de negócio da aplicação. Nessa camada foi necessário criar a lógica para acionar o modelo treinado. Essa camada também possui a responsabilidade de atualizar a base do Pesquisa Textual com auxílio da ferramenta *Tesseract*. Para essa indexação é utilizada uma solução de mensageria que faz o enfileiramento de todos os documentos assinados que são juntados aos processos, baseado nessa ordem é feita a indexação.
- Banco de Dados: é responsável por armazenar todas as informações processuais inclusive os documentos processuais.
- Pesquisa Textual: é uma base de dados textual onde o conteúdo dos arquivos é indexado para permitir uma alta performance na busca.
- Modelo Treinado: camada desenvolvida em *Python* capaz de retornar as sentenças relevantes de acordo com a pesquisa. Utiliza das informações da base de dados do Pesquisa Textual.

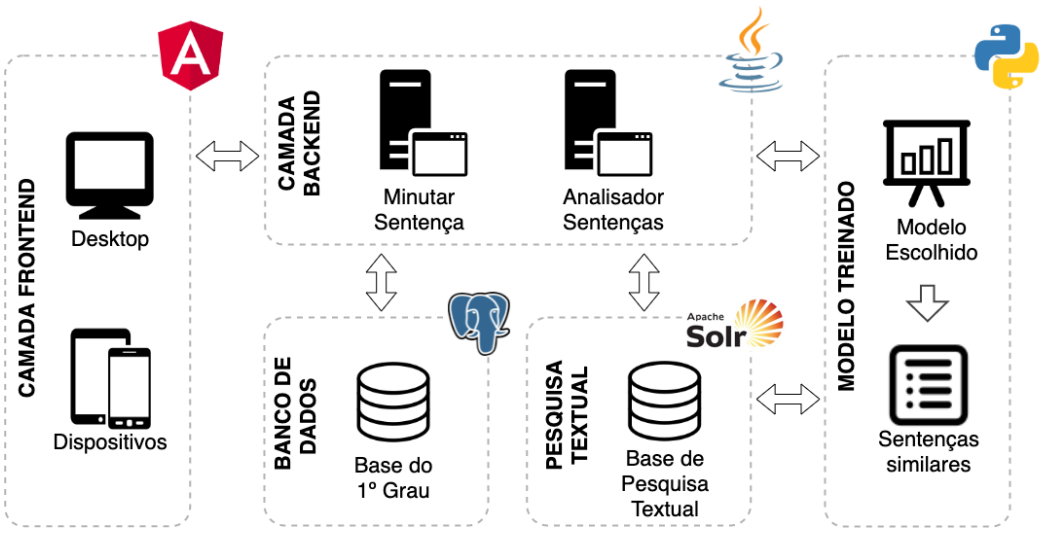


Figura 3.9: Arquitetura da solução.

3.1.3 Avaliação

Os resultados apresentados pelos modelos serão avaliados por um especialista que poderá sinalizar a concordância ou não com a similaridade apresentada.

Em uma solução de recuperação de informação existem várias métricas que podem avaliar a qualidade do retorno de uma consulta. Nesta pesquisa a maior preocupação será com o resultado do ranking, então foram selecionadas métricas para avaliar a precisão e o ganho cumulativo de cada resultado retornado após a consulta.

A precisão é uma métrica fundamental em um sistema de recuperação de informação, por essa razão foi escolhido a métrica P@k. Essa métrica mensura a quantidade de alternativas relevantes que estão nas primeiras posições do rank, independente da ordem. O objetivo com essa métrica é saber quais são as sentenças similares que estão na mesma porção (até a k-essima posição) do ranking do especialista, se comparados com a mesma consulta.

Essa métrica é calculada com a Equação 3.1

$$P@k = \frac{n}{k} \tag{3.1}$$

onde n é a quantidade de alternativas em comum que estão nas k primeiras posições dos dois ranks (do especialista e da resposta do modelo).

O resultado de P@k varia apenas entre 0 e 1, sendo o resultado 0 (zero) quando o modelo não conseguiu ordenar nenhuma das alternativas nas k primeiras posições corretamente, e o resultado 1 (um) quando o modelo consegue encontrar o mesmo resultado do ranking do especialista nessas mesmas posições.

Ainda, pensando em avaliar os resultados, também, será utilizada a métrica *Normalized Discounted Cumulative Gain* (nDCG) que leva em consideração o posicionamento de cada resultado. Portanto, os elementos das primeiras posições têm maior relevância que dos elementos inferiores do *rank*. O resultado é normalizado pelo ganho considerado ideal (ranking do especialista).

A Equação 3.2 representa o ganho cumulativo:

$$CG = \sum_{i=1}^{n} rel_i \tag{3.2}$$

onde n é a quantidade de alternativas no ranking e *rel* a relevância de cada item do resultado, sem levar em consideração a sua posição. Porém a premissa do nDCG é penalizar os elementos mais relevantes que estejam ocupando posições mais inferiores no rank. Segundo Wang [30], a Equação 3.3 representa do DCG:

$$DCG = \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{3.3}$$

Para manter a mesma grandeza, o valor de DCG deve ser normalizado independente da quantidade de elementos no resultado. Para isso, basta dividir o DCG calculado com o valor do DCG ideal, ou seja, do ranking fornecido pelo especialista. Dessa forma, essa métrica também irá assumir valores decimais entre 0 (zero) até 1 (um). Dizemos que nDCG é igual a 1 quando ambos os ranks são idênticos.

Equação 3.4 representa o nDCG:

$$nDCG = \frac{DCG_{calculado}}{DCG_{especialista}} \tag{3.4}$$

3.1.4 Implantação

Com base na avaliação, será escolhido o melhor modelo a ser utilizado para a identificação de sentenças similares. O modelo escolhido será integrado ao Processo Judicial Eletrônico (PJe) como parte da solução.

3.2 Proposta de Evolução do Minutar e Analisar Sentença

Esta seção apresenta uma proposta de evolução da funcionalidade de Minutar e Analisar Sentenças do Processo Judicial Eletrônico, baseada no modelo escolhido para encontrar as sentenças similares. Essa proposta de evolução é o grande diferencial dessa pesquisa.

3.2.1 Fluxo Geral Principal

O Processo Judicial Eletrônico (PJe) é um sistema baseado em fluxos que utiliza a tecnologia de *Business Process Management* (BPM) para mapear os estados e as transições percorridas pelos processos judicias. O principal fluxo a ser percorrido pelo processo judicial é denominado como "Fluxo Geral Principal"[1]. Nesse fluxo são definidas várias atividades que podem ser realizadas pelos usuários do sistema, por exemplo: triagem inicial, análise, redistribuição, plantão, cumprimento de providências, encaminhar para o posto avançado, registrar trânsito julgado, remeter ao segundo grau, controle de prazo, concluso para o magistrado, minutar e analisar sentenças, etc.

[1]https://pje.csjt.jus.br/fluxo/fluxo_primeirograu_255/#diagram/4b53852a53cc46d9bea0-5c21f8661235

Portanto, uma das atividades desse fluxo é justamente a tarefa de Minutar e Analisar sentenças. Essa pesquisa propõe uma evolução dessa tarefa, permitindo aos servidores ou aos magistrados pesquisar por sentenças similares ao caso em questão. É interessante ressaltar que o esquema de autorização ou de acesso a essa tarefa não foi alterado, ou seja, a definição dos papéis que podem executar essa tarefa permanece inalterada.

3.2.2 Minutar e Analisar Sentenças revisto

Atualmente a funcionalidade de minutar e analisar sentenças judiciais permite a elaboração de minutas de sentenças utilizando um componente de edição de texto embutido na própria página HTML conforme demonstrado na Figura 3.10. Além desse editor, essa funcionalidade permite a criação das minutas a partir de modelos pré-definidos pelos chefes de gabinetes e, também, permite realizar uma pesquisa no banco de dados de jurisprudência. Contudo, a necessidade de encontrar sentença similares com o caso em questão e a ausência dessa possibilidade na atual funcionalidade de minutar e analisar sentenças fomentaram o desenvolvimento dessa pesquisa.

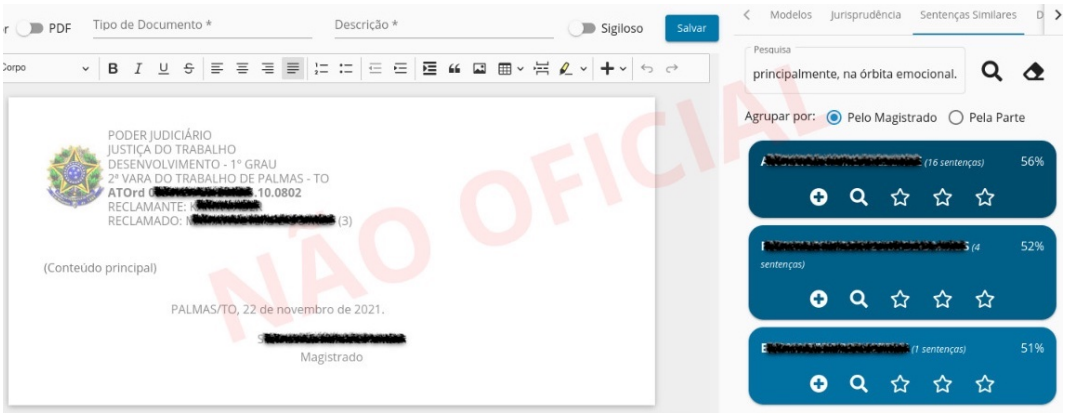


Figura 3.10: Evolução do Minutar e Analisar Sentença.

Portanto, além da escolha do modelo de aprendizado de máquinas para encontrar as sentenças similares, também houve um grande esforço para implementar uma solução que utilizasse esse modelo integrado à funcionalidade de minutar e analisar sentenças. Dessa forma, os usuários não precisarão utilizar de outros meios para realizar essa pesquisa de similaridades que auxiliará na elaboração de uma nova sentença.

A implementação consiste na criação de uma aba do lado direito da tela chamada "Sentenças Similares". Essa aba possui um campo que conterá o texto (*query*) a ser apresentado ao modelo para encontrar as sentenças similares. Os resultados poderão ser agrupados pelo magistrado ou pela parte do processo.

Uma vez que existe o agrupamento das respostas, para cada grupo encontrado é definido um *card* com as seguintes informações:

- O nome do magistrado ou da parte;
- A quantidade de sentenças encontradas do magistrado ou da parte;
- O maior percentual de similaridade encontrado no grupo;

O ranqueamento dos grupos de resultados é feito com base no maior percentual de similaridade encontrado nas sentenças do grupo, seguindo uma ordem decrescente. Também, foi implementado uma gradação de cores para tentar destacar os grupos que apresentaram os melhores resultados.

Além dessa visualização consolidada dos resultados, os usuários poderão avaliar os resultados com relação a sua relevância classificando através dos ícones das estrelas conforme apresentado na Figura 3.11.

Figura 3.11: Exemplo de classificação com relação a relevância dos resultados.

Quanto maior a relevância do resultado mais estrelas o usuário tem que sinalizar. Com essa classificação foi possível realizar o cálculo da métrica nDCG. A orientação para classificação foi a seguinte:

- Três estrelas: os mesmos temas são abordados com as mesmas fundamentações pesquisadas;
- Duas estrelas: os mesmos temas são abordados, porém com uma argumentação diferente na fundamentação;
- Uma ou zero estrela: possuem temas e argumentações diferentes da pesquisada;

Toda essa organização de cores e agrupamento foi implementada para auxiliar os usuários a encontrar as sentenças similares que possam auxiliá-los na confecção de uma nova

sentença. Seguindo essa linha, também foi disponibilizada aos usuários uma funcionalidade para expandir cada grupo de resultado de modo a obter maiores informações sobre os resultados do grupo. Quando acionado essa opção as seguintes informações serão apresentadas: o número do processo e o percentual de similaridade da sentença do processo. Essa lista das sentenças do processo também obedece a ordem decrescente do percentual de similaridade. Nessa lista é possível visualizar os trechos das sentenças que foram encontrados a similaridade, conforme ilustrado na Figura 3.12 e ainda um botão que permite a visualização de todo o teor da sentença.

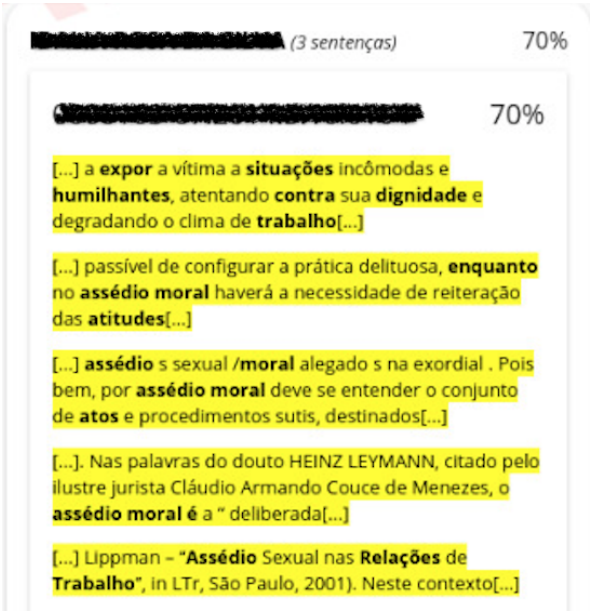


Figura 3.12: Evolução do Minutar e Analisar Sentença.

Essa funcionalidade foi projetada para auxiliar os magistrados e os servidores a encontrarem as argumentações das fundamentações necessárias para produzir uma nova sentença. Acredita-se que a simplicidade e a facilidade da busca embutida no próprio editor da sentença contribuam com a celeridade da elaboração da nova sentença.

3.2.3 Alterações realizadas no PJe

O desenvolvimento dessa proposta de evolução da funcionalidade do Minutar e Analisar Sentenças foi realizada no próprio repositório de código fonte (*Git*) do PJe. O PJe é modularizado em vários projetos que possuem suas responsabilidades bem definidas e por essa razão as alterações se concentrarão em dois projetos:

- *PJE-FRONTEND*: projeto responsável por manter as *interfaces* dos usuários. A principal mudança nesse projeto foi a inclusão da aba "Sentenças Similares";

- *PJE-BACKEND*: projeto responsável por manter as regras de negócio do sistema. A principal alteração nesse projeto foi a inclusão das regras de negócio inerentes a pesquisa das sentenças similares;

Pelo fato da equipe de desenvolvimento do PJe fazer uso do padrão *Gitflow*, as alterações foram realizadas em ramificações (*branches*) derivadas da ramificação principal. Para ser colocado em produção, essas alterações deverão ser integradas à ramificação principal após uma análise da qualidade do código produzido.

3.2.4 Implantação em produção

O primeiro procedimento a ser realizado para implantar a solução em produção é a avaliação das requisições de integração (*merges requests*). Uma requisição de integração para cada projeto foi aberta, sendo que as alterações passaram pelas avaliações automáticas do PJe implementadas por ferramentas como *Sonar* e *Lint*. Apesar da solução ter seguido as definições do Guia de Desenvolvimento Recomendado do PJe, também é necessário a aprovação por parte do revisor de código. Essa é a forma adotada para garantir um mínimo de qualidade de código.

Uma vez integrada a solução na ramificação principal, essa estará disponível para testes no ambiente oficial de testes do PJe, assim o Grupo de Negócio poderá realizar a homologação da aba de Similaridade de Sentenças. Também é importante destacar que a funcionalidade foi testada por um grupo reduzido envolvido nessa pesquisa, e para fazer a publicação dessa solução no ambiente de produção, os testes deverão ser expandidos pelo Grupo de Negócio definido no PJe.

Capítulo 4

Experimentos e Resultados

Este capítulo apresenta o desenvolvimento da pesquisa e os resultados de cada modelo (BM25, LDA e Doc2Vec) na tarefa de recuperação de sentenças similares.

4.1 *Queries*

As *queries* são os textos de pesquisas que foram submetidas aos modelos. Elas foram agrupadas por cinco temas escolhidos pelos especialistas dos dados:

- Reconhecimento ou admissão de vínculo de emprego;
- Indenização por assédio (danos morais);
- Honorários advocatícios sucumbenciais;
- Gratuidade de justiça;
- Horas extras;

Com ajuda dos especialistas dos dados, para cada tema foram elaborados dois textos que descrevem a principal argumentação da fundamentação sobre determinado tema. Com esses textos, foi possível buscar por sentenças similares com a mesma argumentação, em duas execuções dos modelos propostos.

Na primeira execução, as *queries* foram definidas por um magistrado da seguinte forma:

1. **Reconhecimento ou admissão de vínculo de emprego:** "Atente-se que a ilicitude é agravada pela impossibilidade de reconhecimento de vínculo de emprego com a Administração, em função da necessidade de concurso público para tanto, o que implica conferir-se tratamento vantajoso à ilicitude";

2. **Indenização por assédio (danos morais)**: "Quanto ao dano moral, em rápida definição, é aquele dano referente a lesões sofridas pela pessoa em seu patrimônio de valores exclusivamente morais e ideais";

3. **Honorários advocatícios sucumbenciais:** "Os honorários assistenciais são devidos na ocorrência concomitante de dois requisitos: o benefício da justiça gratuita e assistência por sindicato";

4. **Gratuidade de justiça:** "Requerido na forma legal, defiro ao Reclamante os benefícios da assistência judiciária gratuita";

5. **Horas Extras (sobrejornada e supressão do intervalo intrajornada):** "Caracterizada a habitualidade na prestação do serviço suplementar é devida a indenização, sempre que configurada a supressão do labor extraordinário";

Já na segunda execução, foram utilizadas *queries* definidas pelo consenso de três especialista, que atuam em diferentes gabinetes no Tribunal. As seguintes *queries* foram definidas:

1. **Reconhecimento ou admissão de vínculo de emprego:** "O princípio protetor do Direito do Trabalho faz presumir a existência de vínculo de emprego, sempre que houver a prestação de serviços. Assim, era do reclamado o ônus de provar o trabalho autônomo e eventual e desse ônus o réu não se desincumbiu a contento";

2. **Indenização por assédio (danos morais)**: "O assédio moral é o ato de expor o trabalhador a situações humilhantes no ambiente de trabalho, de modo a diminuí-lo em relação aos demais colegas de trabalho ou simplesmente em relação às suas habilidades enquanto profissional. Essa atitude atenta contra a dignidade do trabalhador, afetando sua vida tanto na esfera física e, principalmente, na órbita emocional";

3. **Honorários advocatícios sucumbenciais:** "A concessão de honorários advocatícios está condicionada à constatação de dois fatores, quais sejam: a assistência por parte de sindicato do trabalhador e remuneração inferior ou igual a dois salários mínimos mensais pelos assistidos, ou comprovação de situação econômica tal que impossibilite a demanda judicial sem prejuízo de seu próprio sustento";

4. **Gratuidade de justiça:** "É facultado aos juízes, órgãos julgadores e presidentes dos tribunais do trabalho de qualquer instância conceder, a requerimento ou de ofício, o benefício da justiça gratuita, inclusive quanto a traslados e instrumentos, àqueles que perceberem salário igual ou inferior ao dobro do mínimo legal, ou declararem, sob as penas da lei, que não estão em condições de pagar as custas do processo sem prejuízo do sustento próprio ou de sua família";

5. **Horas Extras (sobrejornada e supressão do intervalo intrajornada):** "A supressão, pelo empregador, do serviço suplementar prestado com habitualidade, durante pelo menos 1 (um) ano, assegura ao empregado o direito à indenização correspondente ao valor de 1 (um) mês das horas suprimidas para cada ano ou fração igual ou superior a seis meses de prestação de serviço acima da jornada normal";

Para permitir a avaliação, utilizando P@K e nDCG, os especialistas montaram um ranking das 25 principais sentenças envolvendo cada tema. A escolha da quantidade de 25 sentenças foi baseada no máximo de sentenças que um analista judiciário, em geral, analisa na busca por uma decisão similar, em sua rotina diária.

4.2 *Corpus*

A coleção de busca (*corpus*) é composta por 161.881 sentenças proferidas no intervalo de abril de 2012 a abril de 2020. As sentenças são compostas por cabeçalho, relatório, fundamentação e dispositivo (conclusão), porém para a formação do *corpus* foram consideradas apenas as fundamentações das sentenças e os temas apresentados nas petições iniciais.

Para avaliação foi criado o *Gold Standard Corpus* com 10% da base de dados, dessa forma é possível realizar a comparação entre a representação vetorial das sentenças com a representação vetorial da consulta, conforme ilustrado na Figura 4.1.

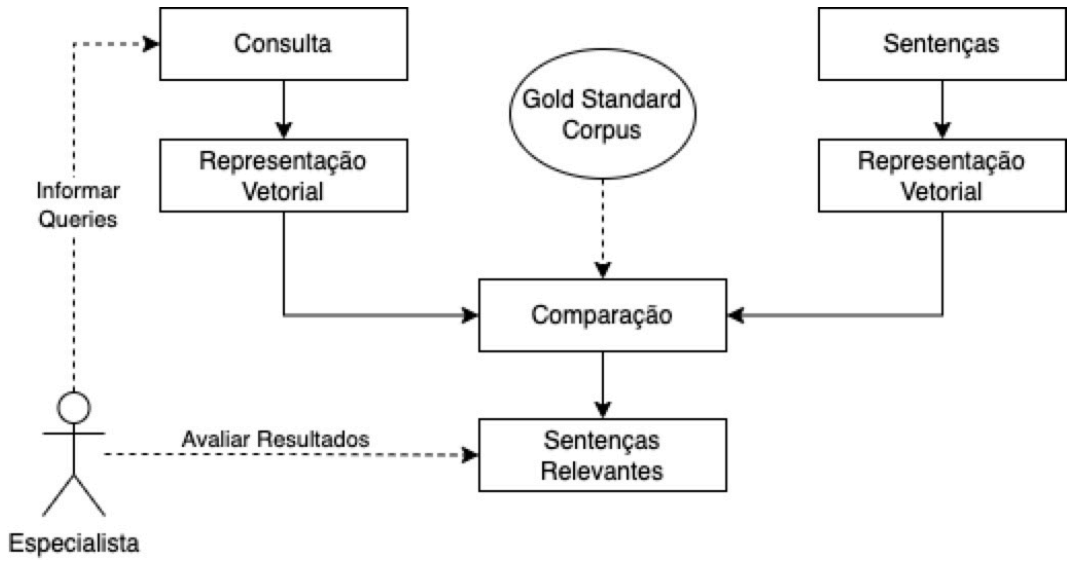


Figura 4.1: Montagem do corpus.

Os seguintes tratamentos foram realizados:

1. Identificação e extração do texto da seção de fundamentação com auxílio de expressões regulares;
2. Remoção das *tags* da linguagem HTML, também com auxílio de expressões regulares;
3. Remoção de caracteres especiais de numeração ordinal;
4. Remoção de acentos;
5. Remoção de *stopwords*. Utilizada a lista do pacote *nltk.corpus.stopwords* no idioma português com adição de algumas palavras comuns em decisões jurídicas (ex.: tribunal, regional, trabalho);
6. Identificação dos parágrafos com auxílio de expressões regulares;
7. Identificação das palavras com auxílio do pacote *Dictionary* do *Gensim*;
8. Remoção de palavras menores de 3 caracteres;
9. Remoção de palavras que aparecem em apenas uma sentença;
10. Identificação dos *tokens* com auxílio da função *word_tokenize* do pacote *NLTK*;
11. Redução para o radical (*stem*), também com auxílio do pacote *NLTK*;

4.3 Detalhes de implementação dos modelos

Nessa seção serão apresentados os detalhes de implementação de cada modelo utilizado com intuito de demonstrar como foram realizados os treinamentos e a forma de execução dos três modelos induzidos (BM25, LDA e Doc2Vec).

4.3.1 *Best Match 25*

A indexação dos documentos é um passo fundamental para a implementação desse modelo. Durante a realização da indexação dos documentos é realizado o cálculo dos pesos dos termos possibilitando a comparação dos documentos a partir desse cálculo. O tamanho do documento e a quantidade de ocorrência dos temos no documento são características levadas em consideração durante a realização do cálculo.

Portanto, foi utilizado o *corpus* descrito na Seção 4.1 para criação do dicionário e para o cálculo dos pesos dos termos. A biblioteca *Gensim* foi a escolhida para a implementação do modelo devido a sua praticidade e vasta documentação.

Os principais passos realizados durante o treinamento desses modelos foram:

1. Realizado os procedimentos de pré-processamento descritos na seção do *corpus*;
2. Criado o índice do *BM25* com auxílio do pacote *rank_bm25* da biblioteca *BM25Okapi* diante do *corpus* gerado;
3. Realizados os procedimentos de pré-processamento nas *queries*:
    - Remoção de acentos;
    - Remoção de *stopwords*;
    - Identificação das palavras com auxílio do pacote *Dictionary* do *Gensim*;
    - Remoção de palavras menores de 3 caracteres;
    - Tokenização;
4. Utilizando o índice do *BM25* criado e os *tokens* da *query* gerados, é realizada a pesquisa para trazer os documentos mais relevantes. Para realização dessa etapa, foi utilizada a função *get_top_n* da biblioteca *BM25Okapi* com o valor 25 atribuído para o parâmetro n (definição da quantidade de documentos relevantes que devem ser retornados pela função). Foram adotados os valores padrões para os parâmetros do BM25 $k = 1{,}5$ e $b = 0{,}75$. O parâmetro k define a suavização da frequência do termo e o parâmetro b define a penalização para o tamanho do documento em relação ao tamanho médio de documentos do corpus;
5. Lista dos 25 documentos (sentenças) mais relevantes, segundo o modelo;

4.3.2 *Latent Dirichlet Allocation*

A implementação desse algoritmo utilizou a abordagem em *batch*.Uma das maiores dificuldades na implementação desse modelo é a definição da quantidade de tópicos. Com intuito de tentar achar a quantidade que melhor atende ao propósito de encontrar a similaridade das sentenças, foram definidos cinco diferentes valores para a quantidade de tópicos (50, 100, 150, 200 e 250 tópicos).

Os dados foram divididos para permitir o treinamento, a validação e o teste do modelo LDA com as quantidades de tópicos definidas. Sendo 80% dos dados destinado ao treinamento, 10% destinados a validação e 10% ao teste.

Para o treinamento foram usados 80% dos dados e os seguintes passos foram implementados:

1. Realizados os procedimentos de pré-processamento descritos na seção do *corpus*;

2. Com intuito de garantir a frequência e a representação dos *tokens* foi utilizada a função *filter_extremes*. Essa função possui três parâmetros e os seguintes valores foram adotados depois de alguns experimentos:

    - *no_below = 50*, filtra os *tokens* que aparecem em menos de 50 documentos;
    - *no_above = 0.4*, filtra os *tokens* que aparecem em mais de 40% dos documentos;
    - *keep_n = 500000*, mantém apenas os primeiros 500.000 *tokens* mais frequentes;

3. Após a definição do *corpus* e do dicionário, o modelo é treinado com auxílio da biblioteca *Gensim* com uso do pacote *LdaMulticore*. O seguinte parâmetro foi ajustado:

    - *num_topics = 50*, define a quantidade de tópicos. O modelo foi inicialmente treinado com 50 tópicos, porém, posteriormente, o treinamento foi realizado com 100, 150, 200 e 250 tópicos;

4. Com o modelo treinado, foram realizados os procedimentos de pré-processamento nas *queries*:

    - Remoção de caracteres especiais de numeração ordinal;
    - Remoção de acentos;
    - Remoção de *stopwords*;
    - Identificação das palavras com auxílio do pacote *Dictionary* do *Gensim*;
    - Remoção de palavras menores de 3 caracteres;
    - Tokenização;

5. Após a definição do *corpus* das *queries*, esse será submetido aos modelos treinados para encontrar os vetores que as representam;

6. Para encontrar a similaridade dos documentos, é utilizado a função de semelhança (*gensim.matutils.cossim*) baseado nos cossenos dos vetores;

7. Lista dos 25 documentos (sentenças) mais relevantes para o modelo LDA;

4.3.3 Doc2Vec

O uso de modelos semânticos na predição possui dois grandes problemas: i) alto custo computacional; ii) realizar treinamento contínuo a cada adição de documento é um grande desafio. Portanto, a utilização de modelos pré-treinados se torna uma solução a ser considerada [31].

Em Hartmann [32], pesquisadores da Universidade de São Paulo (USP) disponibilizam publicamente um modelo pré-treinado de representação de palavras em Português. Nessa

publicação é possível visualizar as etapas executadas para minimizar a quantidade de vocabulários. O CBoW foi a arquitetura utilizada para aprender quais são as palavras subjacentes de uma determinada palavra.

Os dados do corpus da Justiça do Trabalho utilizado nesse projeto de pesquisa foram divididos para permitir o treinamento, a validação e o teste do modelo Doc2Vec. Sendo 80% dos dados destinados ao treinamento, 10% destinados à validação e 10% ao teste. Para a representação das palavras, foi utilizado o modelo word3vec disponibilizado em Hartmann [32].

Portanto, o treinamento dos modelos utilizou 80% dos dados e os seguintes passos foram implementados:

1. Realizados os procedimentos de pré-processamento descritos na seção do *corpus*;
2. Após a definição do *corpus* e do dicionário, o modelo é treinado com auxílio da biblioteca *Gensim* com uso do pacote *Doc2Vec*. Os seguintes parâmetros foram ajustados:
    - *window = 5*, definição da quantidade de termos da janela.
    - *alpha = 0.025*, definição do valor inicial da taxa de aprendizagem (*learning rate*);
    - *min_alpha = 0.0001*, definição do valor mínimo para *alpha* que cairá linearmente durante o treinamento;
    - *vector_size = 400*, tamanho do vetor de *embedding*; *hs = 0*, define a utilização do *Negative Sampling*;
    - *negative = 6*, definição da quantidade de palavras a serem utilizadas como exemplos negativos;
    - *cbow_mean = 1*, definição do uso da média simples dos vetores de todos os termos no algoritmo CBoW para cálculo da representação numérica do documento, com base no valor da representação de cada termo do documento;
3. Com o modelo treinado, foram realizados os procedimentos de pré-processamento nas *queries*:
    - Remoção de caracteres especiais de numeração ordinal;
    - Remoção de acentos;
    - Remoção de *stopwords*;
    - Identificação das palavras com auxílio do pacote *Dictionary* do *Gensim*;
    - Remoção de palavras menores de 3 caracteres;

- Tokenização;

4. Após a definição do *corpus* das *queries*, esse foi submetido ao modelo treinado para encontrar os vetores dos termos que as representam;
5. Realizado o cálculo da média simples dos termos da pesquisa para encontrar o vetor que representa a pesquisa;
6. Para encontrar a similaridade dos documentos, foi utilizado a função de semelhança (*most_similar*) baseado nos cossenos dos vetores;
7. Lista dos 25 documentos (sentenças) mais relevantes para o modelo Doc2Vec;

4.4 Resultados da Avaliação

O primeiro ponto a ser considerado é que os experimentos tiveram duas rodadas de execução. Na primeira rodada, foram consideradas as *queries* elaboradas por um magistrado para cada um dos temas. Na segunda rodada, as *queries* utilizadas nos modelos foram elaboradas por três especialistas do negócio, sendo resultadas de um consenso entre esses especialistas.

Levando em consideração que existem 5 *queries* e que cada *query* resulta em 25 sentenças similares, então para cada modelo existem 125 sentenças a serem avaliadas conforme a similaridade.

Conforme descrito na Seção 3.1.3 Avaliação, os resultados foram avaliados seguindo duas métricas de sistemas de recuperação de informação:

1. **P@K:** métrica de precisão que leva em consideração a proporção de respostas corretas diante de todos os resultados independente da sua posição;
2. **nDCG:** métrica que leva em consideração o posicionamento das respostas corretas;

Para cálculo das métricas inicialmente foram avaliadas apenas as 25 primeiras sentenças similares de cada execução dos modelos. Ressaltando que esse número foi definido baseado na quantidade máxima de sentenças que um servidor, em geral, analisa para elaborar uma nova minuta de sentença.

4.4.1 Métricas P@K

Inicialmente, para cada modelo foi realizado o cálculo da métrica P@K25, posteriormente, variamos o valor do K calculando as métricas P@K1, P@K5, P@K10 e P@K20. Esse cálculo leva em consideração a rodada de execução (*query* do magistrado ou *query* dos

Tabela 4.1: Índice P@K25 da primeira execução.

#	**Modelo**	**Quantidade de Sentenças Similares**	**P@K25**
1	BM25	101 de 125	0.80
2	LDA250	88 de 125	0.70
3	LDA200	85 de 125	0.68
4	Doc2Vec	73 de 125	0.58
5	LDA150	65 de 125	0.52
6	LDA100	63 de 125	0.50
7	LDA50	50 de 125	0.40

Tabela 4.2: Índice P@K25 da segunda execução.

#	**Modelo**	**Quantidade de Sentenças Similares**	**P@K25**
1	BM25	93 de 125	0.74
2	LDA250	92 de 125	0.73
3	LDA200	90 de 125	0.72
4	Doc2Vec	85 de 125	0.68
5	LDA150	83 de 125	0.66
6	LDA100	78 de 125	0.62
7	LDA50	60 de 125	0.48

três especialistas). As sentenças similares encontradas foram consideradas como uma resposta correta quando essa trata do mesmo tema e possui a mesma argumentação na fundamentação. Essa avaliação de similaridade das sentenças foi confirmada com um especialista dos dados. A Tabela 4.1 apresenta o resultado da métrica P@K25 da primeira execução utilizando as *queries* definidas por um magistrado. Com intuito de avaliar a generalidade dos modelos, foram produzidas novas *queries* com os mesmos assuntos, porém com a visão de três analistas de dados diferentes. Cada um dos analistas trabalha em um gabinete diferente e tiveram que entrar em um consenso para a definição das *queries*.

A Tabela 4.2 apresenta o resultado da métrica P@K25 da segunda execução utilizando as *queries* definidas pelo consenso de três analistas dos dados. Com o índice P@K25 de cada execução, foi possível realizar o cálculo desse índice utilizando as duas execuções, para isso foi necessário somar a quantidade de resultados relevantes e quantidade total de possibilidades de cada modelo treinado.

Conforme apresentado na Tabela 4.3, o modelo BM25 com 0.77 foi o que apresentou o melhor desempenho segundo a métrica P@K25, ou seja, segundo essa métrica os resultados apresentados pelo modelo BM25 foram relevantes em mais de 77% dos resultados. Na segunda posição está o modelo LDA com 250 tópicos. Esse modelo obteve o índice de 0.72 de P@K, ou seja, mais de 72% dos resultados foram relevantes para a pesquisa. O terceiro

Tabela 4.3: Média do índice P@K25.

#	**Modelo**	**Quantidade de Sentenças Similares**	**P@K25**
1	BM25	194 de 250	0.77
2	LDA250	193 de 250	0.72
3	LDA200	175 de 250	0.70
4	Doc2Vec	158 de 250	0.63
5	LDA150	148 de 250	0.59
6	LDA100	141 de 250	0.56
7	LDA50	110 de 250	0.44

lugar também foi do modelo LDA, porém agora com 200 tópicos. Para essa configuração o valor do índice ficou em 0.70, ou seja, mais de 70% de relevância nos resultados encontrados. Nas últimas posições da análise da métrica P@K25 ficaram os modelos Doc2Vec, LDA150, LDA100 e LDA50 com 0.63, 0.59, 0.56 e 0.44, respectivamente.

Com intuito de verificar a relevância dos resultados nas primeiras posições, foi refeito o procedimento detalhado acima para calcular a métrica P@K, variando o K entre 1, 5, 10, 20. O gráfico apresentado na Figura 4.2 apresenta os resultados. Percebe-se que o BM25 independentemente do valor do K sempre está mais próximo do resultado ideal (valor 1).

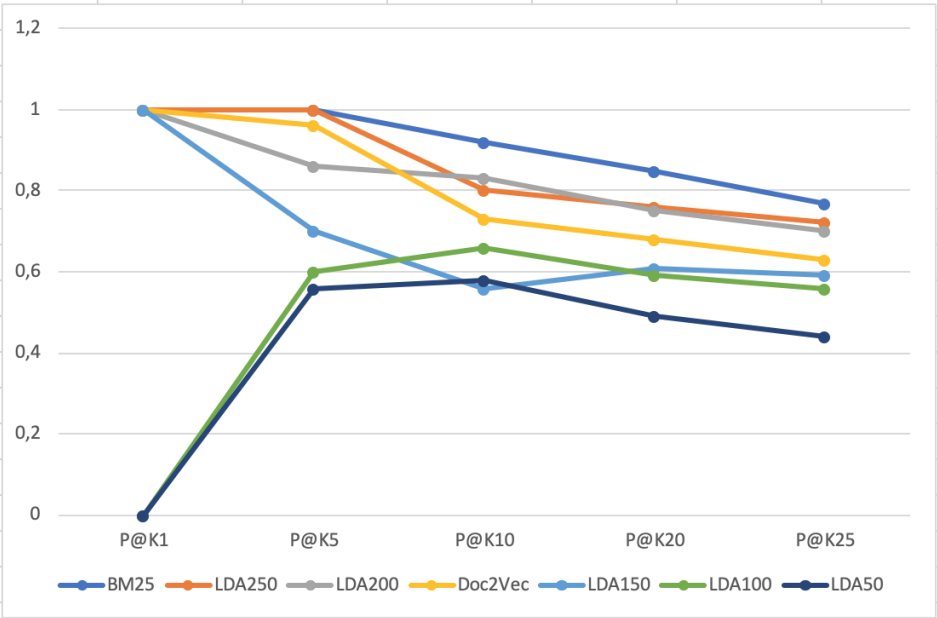


Figura 4.2: Métrica P@K variando o K.

4.4.2 Métricas nDCG

Também foi feito o cálculo da métrica nDCG que leva em consideração o posicionamento das sentenças similares no ranqueamento das respostas. Semelhante ao que foi feito na avaliação do P@K, foram analisadas as 25 sentenças mais similares na resposta de cada modelo.

A eficácia de um sistema de recuperação de informação pode ser aferida com a medida nDCG. A variação dessa média é entre 0 e 1, sendo 1 para a ordenação ideal dos documentos, com os documentos mais relevantes ocupando as primeiras posições do ranqueamento das respostas. Para o cálculo da métrica nDCG, foi necessário a classificação das respostas conforme a sua relevância. Cada resposta foi avaliada com auxílio de um especialista dos dados e foi atribuído um valor de relevância para cada sentença encontrada. Sendo que a classificação da relevância seguiu os seguintes critérios:

- **Irrelevante (valor 0):** a sentença encontrada não aborda o assunto da *query* e também não tem relação com os argumentos da fundamentação pesquisada. Essa resposta é considerada como um erro do modelo;
- **Relevante (valor 1):** a sentença encontrada aborda o assunto, porém os argumentos definidos na fundamentação são diferentes da argumentação da *query*;
- **Muito Relevante (valor 2):** a sentença encontrada aborda o mesmo assunto e os mesmos argumentos da fundamentação da *query*. Essa classificação é considerada como acerto do modelo em encontrar a similaridade da sentença;

A biblioteca apresentada na conferência TREC11[1] foi utilizada para realizar o cálculo da métrica nDCG@25. Para o cálculo, essa biblioteca utiliza dois arquivos de entrada:

- Arquivo ***qrels***: contém as informações da identificação da *query* (Q1, Q2, Q3, Q4 ou Q5), do número do processo da sentença encontrada como resultado da pesquisa, do modelo responsável por ter encontrada a sentença similar e do valor de relevância classificada pelo analista de dados;
- Arquivo ***runs***: contém as informações sobre a execução dos modelos com todas as informações incluindo a posição no ranking: identificação da *query*, modelo, número do processo, posição no ranking e valor de relevância;

Portanto, depois de realizado a primeira execução com as *queries* definidas pelo magistrado foi possível calcular a métrica nDCG@25 dessa execução. A métrica foi calculada

[1] https://github.com/usnistgov/trec_eval

Tabela 4.4: Índice nDCG@25 da primeira execução.

#	**Modelo**	**Média da métrica nDCG@25 por modelo**
1	BM25	0.7885
2	LDA200	0.6352
3	LDA250	0.5876
4	Doc2Vec	0.5421
5	LDA150	0.4656
6	LDA100	0.3459
7	LDA50	0.3361

Tabela 4.5: Índice nDCG@25 da segunda execução.

#	**Modelo**	**Média da métrica nDCG@25 por modelo**
1	LDA250	0.8154
2	BM25	0.7132
3	LDA200	0.6323
4	Doc2Vec	0.5567
5	LDA150	0.5130
6	LDA100	0.5083
7	LDA50	0.4865

para cada *query*, porém para permitir a avaliação da performance do modelo independentemente das *queries* foi calculado a média da métrica por modelo. A Tabela 4.4 apresenta esse resultado.

Os mesmos procedimentos foram realizados para permitir o cálculo da média da métrica nDCG@25 por modelo da segunda execução (*queries* definidas pelo consenso de três analistas de dados). Esse resultado pode ser visto na Tabela 4.5.

Após o cálculo do nDCG para as duas execuções, foi possível calcular a média dos indicadores permitindo a ordenação dos modelos conforme os seus resultados de relevância. A Tabela 4.6 apresenta a ordenação dos modelos conforme essa média do índice nDCG.

Tabela 4.6: Média do índice nDCG@25.

#	**Modelo**	**Média da métrica nDCG@25 por modelo**
1	BM25	0.8019
2	LDA250	0.6742
3	LDA200	0.6099
4	Doc2Vec	0.5494
5	LDA150	0.4893
6	LDA100	0.4271
7	LDA50	0.4113

Tabela 4.7: Média, variância e desvio padrão das métricas.

Métrica Calculada	Média	Variância	Desvio Padrão
P@K1	0,7143	0,2041	0,4518
P@K5	0,8114	0,0308	0,1756
P@K10	0,7257	0,0153	0,1237
P@K20	0,6757	0,0127	0,1129
P@K25	0,6300	0,0107	0,1036
nDCG	0,5661	0,0169	0,1302

Segundo a métrica nDCG, o modelo BM25 também obteve os melhores resultados, apresentando a maior quantidade de sentenças relevantes nas primeiras posições do ranqueamento da resposta. O modelo alcançou o valor de 0.8019 para a média do índice nDCG, esse valor sendo quase 20% superior se comparado com o segundo modelo que apresentou os melhores resultados.

O modelo LDA com 250 tópicos conseguiu alcançar o índice de 0.6742, que caracteriza um bom resultado. Em terceiro lugar também é ocupado pelo modelo LDA, porém com 200 tópicos. Com essa análise é possível perceber que as primeiras posições dessa métrica são ocupadas por modelos que utilizam de técnicas sintáticas.

O modelo Doc2Vec, que utiliza de técnicas de análise semântica, aparece na quarta posição com o valor de 0.5494. Nas últimas posições aparecem os modelos LDA150, LDA100 e LDA50 com 0.4893, 0.4271 e 0.4113, respectivamente.

4.4.3 Média, variância e desvio padrão das métricas

O desvio padrão é uma medida que representa a intensidade de dispersão de um determinado conjunto de dados, ou seja, o desvio padrão indica o grau de uniformidade do conjunto de dados. Para ser mais homogêneo, o desvio padrão têm que ser o mais próximo de 0 (zero).

Com intuito de embasar ainda mais os resultados, foi realizado um estudo da dispersão das métricas calculando a média, a variância e o desvio padrão de cada métrica. Essa análise do cálculo do desvio padrão é fundamental para definição do melhor algoritmo. Na Tabela 4.7 os resultados são apresentados.

Com exceção do desvio padrão da métrica P@K1, todos os demais tem essa medida entre 0,1036 e 0,1756. Considerando esses valores não é possível determinar com exatidão qual é o melhor algoritmo, porém é possível determinar que o modelo Doc2Vec não será o melhor modelo em nenhuma dessas métricas.

Capítulo 5

Conclusão e Trabalhos Futuros

Este capítulo apresenta as conclusões da pesquisa e também relata as oportunidades para trabalhos futuros que poderão dar continuidade a essa pesquisa.

5.1 Conclusão

Esta pesquisa apresentou um estudo dentro do domínio de Recuperação da Informação (RI) para textos jurídicos, especificamente para sentenças judiciais. A análise se concentrou em sentenças judiciais produzidas no Tribunal Regional da 10ª Região no período de 2012 a 2020. Avaliou-se o desempenho de três algoritmos para encontrar as similaridades das sentenças judiciais: BM25, LDA e Doc2Vec. Nesse contexto, foram selecionados 5 temas com duas versões de *queries* para cada um dos temas. Foram utilizadas as métricas P@K e nDCG para avaliar os resultados dos experimentos. Além do estudo sobre a melhor solução para recuperação das sentenças similares, também foi implementada uma proposta de evolução da funcionalidade Minutar e Analisar Sentença do PJe propondo uma pesquisa dessas sentenças na própria tela de edição das sentenças.

Portanto, depois da avaliação dos experimentos e a implementação da proposta de evolução da funcionalidade de edição de sentenças foi possível chegar as seguintes conclusões da hipótese do trabalho:

- **O uso de técnica de avaliação semântica de texto aumentará a performance dos modelos a serem propostos para a identificação da similaridade entre sentenças judiciais do 1º grau da Justiça do Trabalho comparados ao uso de técnicas de avaliação sintática (léxica).**

  Diante dos experimentos executados utilizando a base de dados das sentenças judiciais do Tribunal Regional da 10ª Região, os modelos sintáticos (BM25 e LDA) tiveram os melhores resultados se comparado com o modelo semântico (Doc2Vec).

Com o cálculo estatístico do desvio padrão dessas métricas não foi possível determinar exatamente a ordem dos melhores modelos, porém é possível afirmar que o modelo Doc2Vec (único modelo semântico) não é o melhor modelo, essa disputa fica entre os modelos BM25, LDA250 e LDA200. Os detalhes desses resultados podem ser vistos no Capítulo 4 especificamente na Seção 4.4 Resultados da Avaliação.

5.2 Trabalhos Futuros

Neste trabalho foi apresentado um estudo demostrando que é possível utilizar técnicas de mineração de texto para encontrar similaridades de sentenças judiciais. A pesquisa se concentrou basicamente em duas peças processuais: as petições iniciais e as sentenças. Uma evolução da pesquisa é incluir os documentos do tipo contestação, pois nesses documentos também são apresentados pedidos que deverão ser tratados pelas sentenças judiciais.

Também foi apresentada uma evolução da funcionalidade de Minutar e Analisar Sentenças do PJe, com a qual os usuários do sistema podem avaliar a relevância dos resultados conforme o texto de pesquisa (*query*). Essa avaliação foi utilizada para criar métricas de avaliação dos modelos, e potencializar a melhoria da performance deles.

O modelo BERT tem apresentado resultados promissores em soluções que abordam a problemática de perguntas e respostas (*Question Answering*), portanto esse poderia ser uma outra opção de modelo semântico a ser analisada.

Como o BM25 foi o algoritmo que apresentou os melhores resultados, então uma alternativa para avaliação futura é comparar com algumas variações desse próprio algoritmo. Também é possível o uso de uma abordagem de *learning rate* para otimizar os resultados.

O Solr é uma ferramenta já conhecida para a problemática de recuperação da informação e já é utilizada na Justiça do Trabalho. A sua performance com relação ao tempo de resposta é indiscutível, então uma abordagem que pode ser analisada é o uso da estratégia definida nesse trabalho para refinar os resultados do Solr. Dessa forma, a *query* seria submetida primeiramente para o Solr, que seria o responsável por trazer uma quantidade pré-definida de resultados, e esses seriam reorganizados conforme a similaridade das sentenças.

Por fim, outra possibilidade de trabalhos futuros é a criação de métricas para avaliar o *feedback* dos usuários que pode ser mensurado através da quantidade de cliques necessários para a formulação de uma sentença judicial, a quantidade de tempo ou mesmo a quantidade de sentenças similares que foram classificadas utilizando as estrelas.

Referências

[1] Tartuce, Fernanda, Maria Cec´de Araujo Asperti, Programa de Pós Graduação Lato Sensu e DIREITO G V da FGV: *AS TÉCNICAS DE JULGAMENTO DE CASOS REPETITIVOS E A TRIAGEM DE PROCESSOS E RECURSOS SOB A PERSPECTIVA DO ACESSO À JUSTIÇA INDIVIDUAL*. Revista de Processo| vol, 288(2019):275–299, 2019. 1

[2] Libardoni, Paulo José e Rodrigo Wasem Galia: *EFEITOS DA REFORMA TRABALHISTA NA JUSTIÇA DO TRABALHO: ANÁLISE DA DINÂMICA PROCESSUAL NA SEGUNDA VARA DO TRABALHO DE SANTA MARIA/RS*. Revista Opinião Jur{\'\i}dica (Fortaleza), 19(30):118–148, 2021. 1

[3] Ellis, David: *A behavioural approach to information retrieval system design*. Journal of documentation, 1989. 2

[4] Morel, Regina Lucia M e Elina G Pessanha: *A justiça do trabalho*. Tempo Social, 19:87–109, 2007. 2

[5] Trabalho, Ouvidoria do Tribunal Superior do: *Sobre a Justiça do Trabalho*. `http://www.tst.jus.br/web/acesso-a-informacao/justica-do-trabalho`. 2

[6] Justiça, Conselho Nacional de: *Justiça em Números*. `https://www.cnj.jus.br/pesquisas-judiciarias/justica-em-numeros/`. 2

[7] Trabalho, Ouvidoria do Tribunal Superior do: *Histórico do Processo Judicial Eletrônico da Justiça do Trabalho (PJe-JT)*. `http://www.csjt.jus.br/web/csjt/historico`. 3

[8] Baeza-Yates, Ricardo e Ribeiro Neto Berthier: *Recuperação de Informação: Conceitos e Tecnologia das Máquinas de Busca*. 2013, ISBN 9788582600498. 7

[9] Salton, Gerard e Christopher Buckley: *Term-weighting approaches in automatic text retrieval*. Information Processing and Management, 1988, ISSN 03064573. 9

[10] Robertson, Stephen e Hugo Zaragoza: *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009. 9

[11] Porteous, Ian, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth e Max Welling: *Fast collapsed gibbs sampling for latent dirichlet allocation*. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, páginas 569–577, 2008. 10

[12] Maskeri, Girish, Santonu Sarkar e Kenneth Heafield: *Mining business topics in source code using Latent Dirichlet Allocation*. Proceedings of the 2008 1st India Software Engineering Conference, ISEC'08, páginas 113–120, 2008. 10

[13] Le, Quoc e Tomas Mikolov: *Distributed representations of sentences and documents*. 31st International Conference on Machine Learning, ICML 2014, 4:2931–2939, 2014. 11

[14] Melo, Ari e Mariano Arimariano@unb Br: *Revisão da Literatura: Apresentação de uma Abordagem Integradora*. Em *AEDEM International Conference*, 2017, ISBN 978-84-697-5592-1. 13

[15] Aletras, Nikolaos, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro e Vasileios Lampos: *Predicting judicial decisions of the European court of human rights: A natural language processing perspective*. PeerJ Computer Science, 2016(10):1–19, 2016, ISSN 23765992. 13

[16] Mirończuk, Marcin Michał e Jarosław Protasiewicz: *A recent overview of the state-of-the-art elements of text classification*. Expert Systems with Applications, 106:36–54, 2018, ISSN 09574174. 13

[17] Ko, Youngjoong e Jungyun Seo: *Text classification from unlabeled documents with bootstrapping and feature projection techniques*, 2009. ISSN 03064573. 14

[18] Shmueli, Galit e Otto R. Koppius: *Predictive analytics in information systems research*, 2011. ISSN 02767783. 14

[19] Larsen, Bjornar e Chinatsu Aone: *Fast and effective text mining using linear-time document clustering*. 1999. 14

[20] Barco Ranera, Lorenz Timothy, Geoffrey A Solano e Nathaniel Oco: *Retrieval of Semantically Similar Philippine Supreme Court Case Decisions using Doc2Vec*. Em *2019 International Symposium on Multimedia and Communication Technology (ISMAC)*, páginas 1–6, 2019. 14

[21] Novotná, Tereza e others: *Document similarity of czech supreme court decisions*. Masaryk University Journal of Law and Technology, 14(1):105–122, 2020. 14

[22] Kim, Mi Young, Ying Xu e Randy Goebel: *Legal question answering using ranking SVM and syntactic/semantic similarity*. Em *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9067, páginas 244–258. Springer Verlag, 2015, ISBN 9783662481189. 15

[23] Raghav, K., Pailla Balakrishna Reddy, V. Balakista Reddy e Polepalli Krishna Reddy: *Text and citations based cluster analysis of legal judgments*. Em *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, ISBN 9783319268316. 15

[24] Mandal, Arpan, Raktim Chaki, Sarbajit Saha, Kripabandhu Ghosh, Arindam Pal e Saptarshi Ghosh: *Measuring similarity among legal court case documents*. Em *ACM International Conference Proceeding Series*, páginas 1–9. Association for Computing Machinery, novembro 2017, ISBN 9781450353236. 15

[25] Wagh, Rupali e Deepa Anand: *Application of citation network analysis for improved similarity index estimation of legal case documents : A study*. Em *2017 IEEE International Conference on Current Trends in Advanced Computing, ICCTAC 2017*, volume 2018-Janua, páginas 1–5. Institute of Electrical and Electronics Engineers Inc., janeiro 2018, ISBN 9781509049974. 15

[26] Mathai, Sumi, Deepa Gupta e G. Radhakrishnan: *Iterative concept-based clustering of Indian court judgments*. Em *Advances in Intelligent Systems and Computing*, volume 712, páginas 91–103. Springer Verlag, 2018, ISBN 9789811082276. 15

[27] Kumar, Sushanta, P Krishna Reddy, V Balakista Reddy e Aditya Singh: *Similarity analysis of legal judgments*. Em *Proceedings of the Fourth Annual ACM Bangalore Conference*, páginas 1–4, 2011. 15

[28] Xue, Mu: *A text retrieval algorithm based on the hybrid LDA and Word2Vec model*. Em *2019 International Conference on Intelligent Transportation, Big Data \& Smart City (ICITBS)*, páginas 373–376. IEEE, 2019. 16

[29] Shearer, Colin: *The CRISP-DM model: the new blueprint for data mining*. Journal of data warehousing, 5(4):13–22, 2000. 17, 18

[30] Wang, Yining, Liwei Wang, Yuanzhi Li, Di He, Wei Chen e Tie Yan Liu: *A theoretical analysis of NDCG ranking measures*. Em *Journal of Machine Learning Research*, 2013. 28

[31] Rodriguez, Pedro L e Arthur Spirling: *Word Embeddings: What works, what doesn't, and how to tell the difference for applied research*. The Journal of Politics, 2021. 39

[32] Hartmann, Nathan, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jessica Rodrigues e Sandra Aluisio: *Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks*. (Section 3), 2017. `http://arxiv.org/abs/1708.06025`. 39, 40

Apêndice A

Detalhamento da Avaliação

Este capítulo apresenta uma perspectiva dos resultados por assunto, diferentemente do que foi apresentado durante os capítulos anteriores que apresentou os resultados sob a perspectiva dos modelos.

A.1 Apresentação dos resultados por assunto

Esta seção apresenta a avaliação das 25 sentenças mais relevantes para cada *query*, ou seja, para cada tema escolhido. A relevância é classificada em 0 (sem relevância), 1 (mesmo tema porém argumentação diferente) e 2 (mesmo tema e mesma argumentação).

A.1.1 Execução 1: *Query* 1

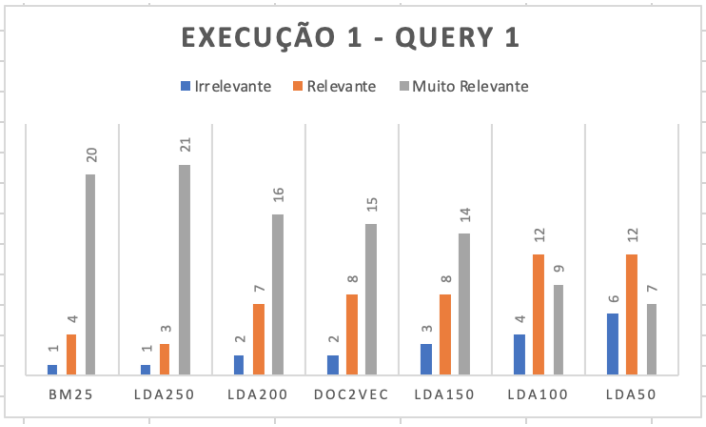


Figura A.1: Resultados da *query* 1 na primeira execução.

Tema: Reconhecimento ou admissão de vínculo de emprego.

***Query*:** "Atente-se que a ilicitude é agravada pela impossibilidade de reconhecimento de vínculo de emprego com a Administração, em função da necessidade de concurso público para tanto, o que implica conferir-se tratamento vantajoso à ilicitude."

A.1.2 Execução 1: *Query* 2

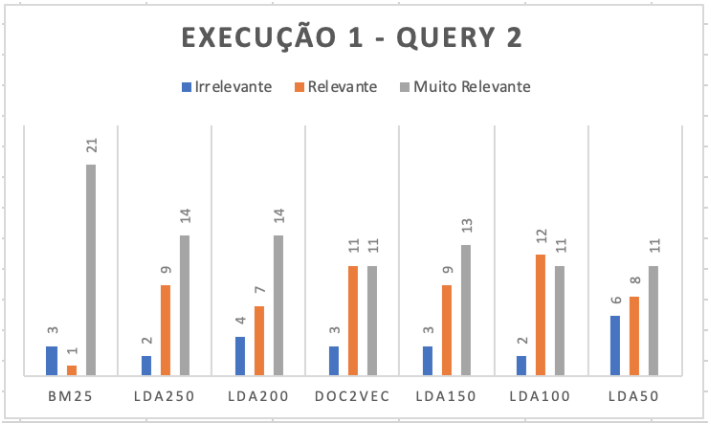


Figura A.2: Resultados da *query* 2 na primeira execução.

Tema: Indenização por assédio (danos morais).

***Query*:** "Quanto ao dano moral, em rápida definição, é aquele dano referente a lesões sofridas pela pessoa em seu patrimônio de valores exclusivamente morais e ideais."

A.1.3 Execução 1: *Query* 3

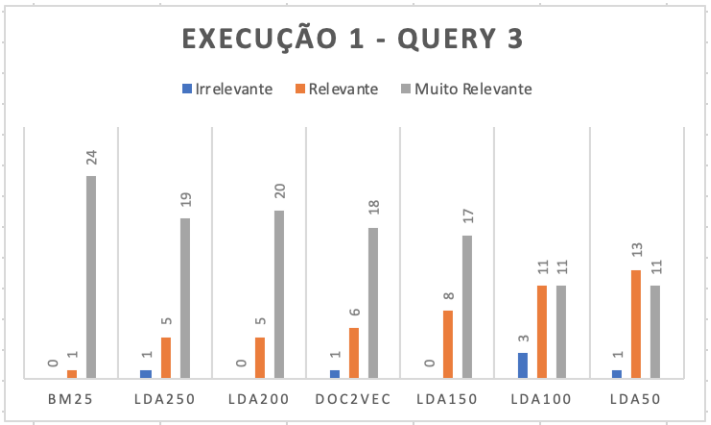


Figura A.3: Resultados da *query* 3 na primeira execução.

Tema: Honorários advocatícios sucumbenciais.

***Query*:** "Os honorários assistenciais são devidos na ocorrência concomitante de dois requisitos: o benefício da justiça gratuita e assistência por sindicato."

A.1.4 Execução 1: Query 4

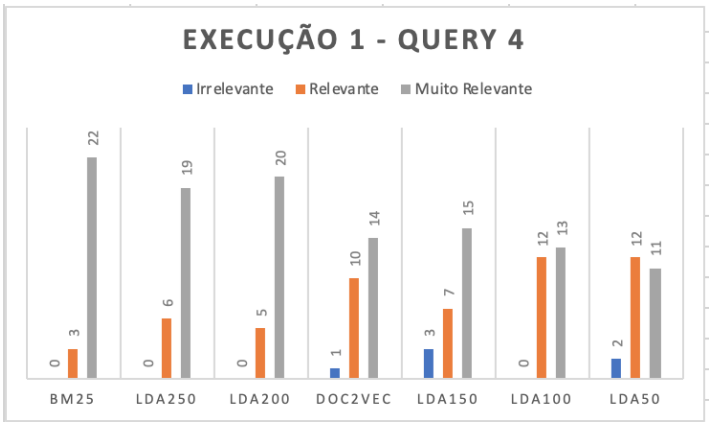


Figura A.4: Resultados da *query* 4 na primeira execução.

Tema: Gratuidade de justiça.

***Query*:** "Requerido na forma legal, defiro ao Reclamante os benefícios da assistência judiciária gratuita."

A.1.5 Execução 1: *Query* 5

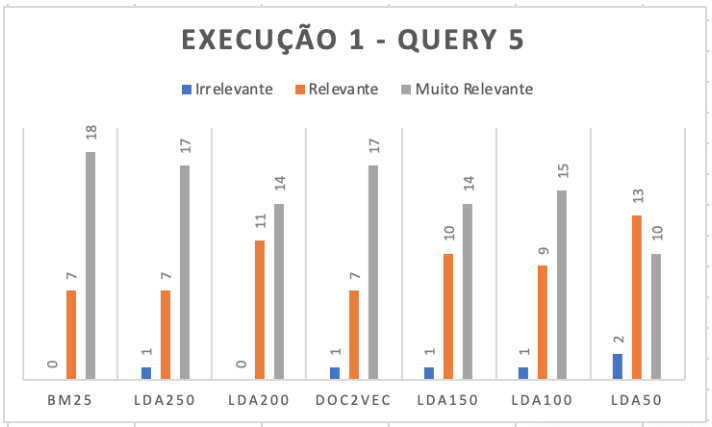


Figura A.5: Resultados da *query* 5 na primeira execução.

Tema: Horas Extras (sobrejornada e supressão do intervalo intrajornada).

***Query*:** "Caracterizada a habitualidade na prestação do serviço suplementar é devida a indenização sempre que configurada a supressão do labor extraordinário. "

A.1.6 Execução 2: *Query* 1

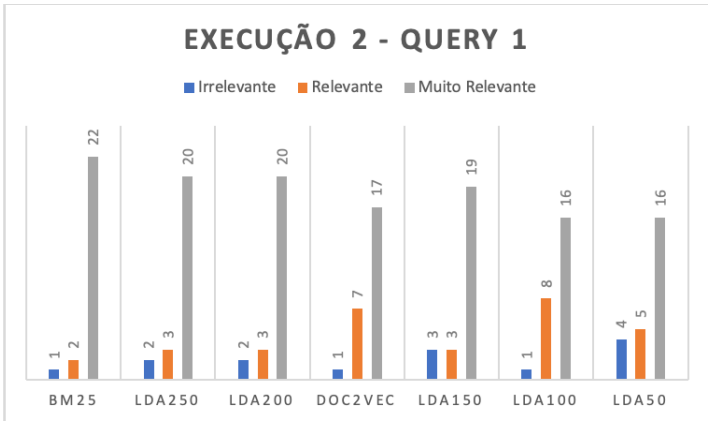


Figura A.6: Resultados da *query* 1 na segunda execução.

Tema: Reconhecimento ou admissão de vínculo de emprego.

***Query*:** "O princípio protetor do Direito do Trabalho faz presumir a existência de vínculo de emprego, sempre que houver a prestação de serviços. Assim, era do reclamado o ônus de provar o trabalho autônomo e eventual e desse ônus o réu não se desincumbiu a contento."

A.1.7 Execução 2: *Query* 2

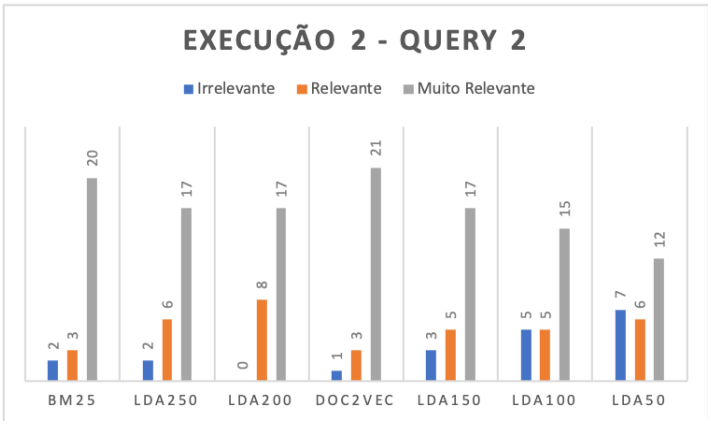


Figura A.7: Resultados da *query* 2 na segunda execução.

Tema: Indenização por assédio (danos morais).

***Query*:** "O assédio moral é o ato de expor o trabalhador a situações humilhantes no ambiente de trabalho, de modo a diminuí-lo em relação aos demais colegas de trabalho ou simplesmente em relação às suas habilidades enquanto profissional. Essa atitude atenta contra a dignidade do trabalhador, afetando sua vida tanto na esfera física e, principalmente, na órbita emocional."

A.1.8 Execução 2: *Query* 3

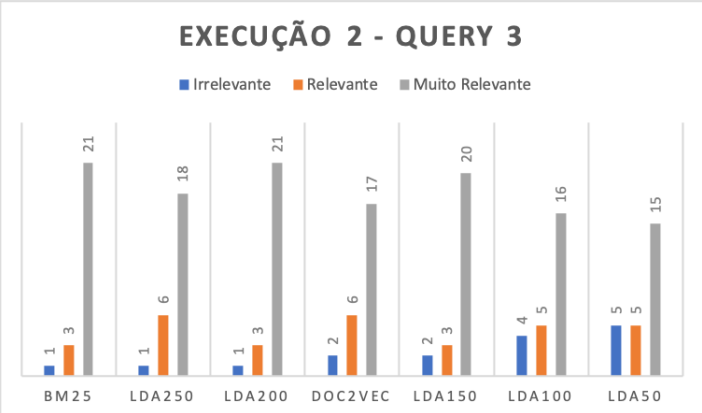


Figura A.8: Resultados da *query* 3 na segunda execução.

Tema: Honorários advocatícios sucumbenciais.

***Query*:** "A concessão de honorários advocatícios está condicionada à constatação de dois fatores, quais sejam: a assistência por parte de sindicato do trabalhador e remuneração inferior ou igual a dois salários mínimos mensais pelos assistidos, ou comprovação de situação econômica tal que impossibilite a demanda judicial sem prejuízo de seu próprio sustento"

A.1.9 Execução 2: *Query* 4

Tema: Gratuidade de justiça.

***Query*:** "É facultado aos juízes, órgãos julgadores e presidentes dos tribunais do trabalho de qualquer instância conceder, a requerimento ou de ofício, o benefício da justiça gratuita, inclusive quanto a traslados e instrumentos, àqueles que perceberem salário igual ou inferior ao dobro do mínimo legal, ou declararem, sob as penas da lei, que não estão em condições de pagar as custas do processo sem prejuízo do sustento próprio ou de sua família"

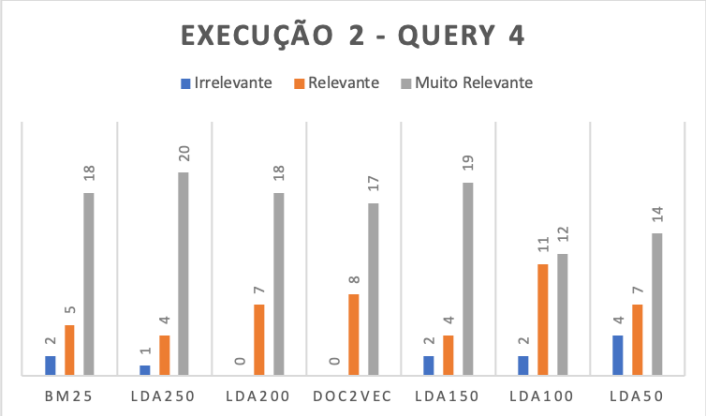


Figura A.9: Resultados da *query* 4 na segunda execução.

A.1.10 Execução 2: *Query* 5

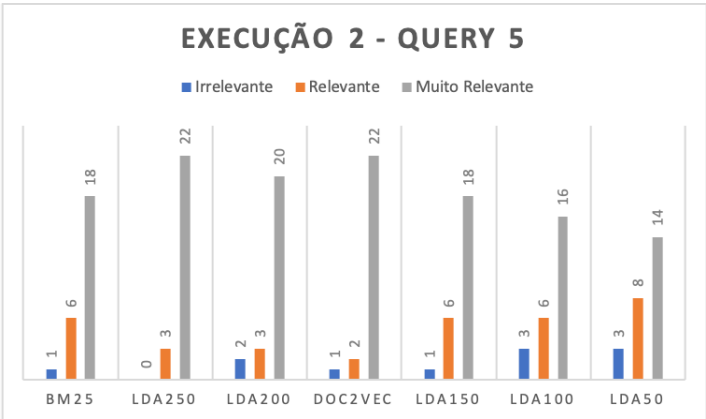


Figura A.10: Resultados da *query* 5 na segunda execução.

Tema: Horas Extras (sobrejornada e supressão do intervalo intrajornada).

***Query*:** "A supressão, pelo empregador, do serviço suplementar prestado com habitualidade, durante pelo menos 1 (um) ano, assegura ao empregado o direito à indenização correspondente ao valor de 1 (um) mês das horas suprimidas para cada ano ou fração igual ou superior a seis meses de prestação de serviço acima da jornada normal."