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

Classificação automatizada de reclamações abertas pelos clientes e usuários do Sistema Financeiro Nacional

João Laterza

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

Orientador
Prof. Dr. Thiago de Paulo Faleiros

Brasília
2023

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

Laterza, João
LL351c Classificação automatizada de reclamações abertas pelos clientes e usuários do Sistema Financeiro Nacional / João Laterza; orientador Thiago de Paulo Faleiros. -- Brasília, 2023.
132 p.

Dissertação(Mestrado Profissional em Computação Aplicada) - Universidade de Brasília, 2023.

1. Processamento de Linguagem Natural. 2. Classificação de Texto. 3. Aprendizagem Profunda. 4. Modelo de Linguagem. 5. Reclamações. I. Faleiros, Thiago de Paulo, orient. II. Título.

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

Classificação automatizada de reclamações abertas pelos clientes e usuários do Sistema Financeiro Nacional

João Laterza

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

Prof. Dr. Thiago de Paulo Faleiros (Orientador)
CIC/UnB

Prof. Dr. Luís Paulo Faina Garcia
CIC/UnB

Profa. Dra. Nádia Félix Felipe da Silva
UFG

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

Brasília, 10 de fevereiro de 2023

Dedicatória

Dedico este trabalho a todos aqueles que se aventuram no vasto universo da ciência de dados buscando conhecimento e aprendizagem.

Agradecimentos

Agradeço ao Professor Thiago de Paulo Faleiros pela orientação, paciência e dedicação dispensada neste trabalho. Agradeço à minha esposa Priscila, pelo apoio em todos os momentos. Agradeço ao meu colega Eric, do mestrado, e aos meus colegas Urias e Renê, do Banco Central do Brasil, pelos conselhos, contribuições e encorajamento ao longo desta jornada.

Resumo

Compete ao Banco Central do Brasil (BCB) atender aos clientes e usuários do Sistema Financeiro Nacional (SFN) na apresentação de reclamações contra produtos e serviços oferecidos pelas entidades supervisionadas, e monitorar o atendimento de suas demandas. O processo de tratamento dessas reclamações é feito manualmente, e consiste no seu enquadramento como "procedente" ou "improcedente", a partir da análise do relato do cidadão, da resposta da entidade reclamada, de documentos anexados e de julgamento sobre se houve ou não indício de descumprimento da regulamentação vigente. Contudo, com o recente aumento no volume de demandas, não é possível, com os recursos disponíveis, analisar a integralidade das remessas. Nesse contexto, foi construída, pelo BCB, solução automatizada para filtrar as reclamações com maior possibilidade de serem procedentes, direcionando, assim, os esforços de análise e julgamento promovidos pelos seus servidores. Destarte, esta pesquisa teve como objetivo desenvolver um classificador para a tarefa em questão com desempenho superior ao do modelo vigente. Para isso, foram exploradas abordagens de aprendizagem profunda, em contraponto ao método tradicional empregado na solução do BCB. Como resultado, foi desenvolvido um classificador baseado em uma estrutura hierárquica combinando um *Bidirectional Encoder Representations from Transformers* (BERT) e uma *Bidirectional Long short-term memory* (BiLSTM) para gerar representações únicas dos textos da reclamação do cidadão e da resposta da entidade reclamada. Essa solução alcançou, na validação cruzada, um desempenho médio - calculado pela Área sob a curva Precisão-Revocação (PRAUC) - de 71,41%, superando em 0,96 pontos percentuais o modelo reproduzido do BCB, e desempenhando 71,92% nos dados de teste. Estimou-se que, caso o classificador proposto fosse utilizado para nortear a tarefa de tratamento das reclamações, seria possível cobrir 90,23% daquelas de fato procedentes avaliando-se apenas 60% do total de demandas. Por fim, foram ainda propostas estratégias multimodais para a combinação das representações textuais descritas com outras variáveis tabulares que compuseram a solução original. No entanto, o ganho obtido pelas estratégias multimodais frente à proposta anterior não foi estatisticamente significante.

Palavras-chave: BCB, SFN, BERT, LSTM, classificação de texto, reclamações

Abstract

The Central Bank of Brazil (BCB) is responsible for assisting customers and users of the National Financial System (SFN) with complaints against products and services offered by its supervised entities and ensuring that all demands are appropriately addressed. Each concern is manually handled and classified as "proceeding" or "unfounded" based on a preliminary analysis of the facts described by the customer, the alleged entity's reply, the attached documents, and according to its compliance with current regulations. However, dealing with an increasing demand with the available resources has turned out to be an unprecedented challenge, being currently impossible to handle all issues accordingly. In this context, the BCB has developed an automated solution to filter complaints more likely to be classified as "proceeding", thus driving the human activities of examination, analysis and judgment. The present study aims to propose a new classifier for this task with better performance than the current model. To this end, deep learning approaches were explored, differing from the traditional method previously employed. As a result, an experimental classifier was tested, based on a hierarchical structure, and combining Bidirectional Encoder Representations from Transformers (BERT) with Bidirectional Long Short-Term Memory (BiLSTM) to generate unique contempt representations of both the citizen's complaint and the entity's reply. In cross-validation, this solution reached an average performance of 71.41%, based on the area under the precision-recall curve (PRAUC), exceeding the current BCB model by 0.96 percentage points, and performing 71.92% on the test dataset. If the proposed classifier was adopted to drive the task of handling complaints, we estimate that approximately 90.23% of the proceeding demands could be identified by evaluating only 60% of the total amount. In addition, multimodal strategies were tested to combine the described textual representations with tabular features of the original classifier. However, when compared to the previous proposed solution, the achieved gain with the multimodal strategies did not reach a statistically significant outcome.

Keywords: BCB, SFN, BERT, LSTM, text classification, complaints

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AUC Área sob a curva.

BCB Banco Central do Brasil.

BERT Bidirectional Encoder Representations from Transformers.

BiLSTM Bidirectional Long short-term memory.

BOW Bag-of-Words.

Deati Departamento de Atendimento Institucional.

DL Deep Learning.

DNN Deep Neural Networks.

GBDT Gradient Boosting Decision Tree.

GPU Graphics Processing Unit.

IA Artificial Intelligence.

IF Instituição Financeira.

IPDR Índice de Procedência de Reclamações.

LGPD Lei Geral de Proteção de Dados Pessoais.

LightGBM Light Gradient Boosting Machine.

LM Language Model.

LSTM Long short-term memory.

ML Machine Learning.

NLP Natural Language Processing.

PR Precisão-Revocação.

RDR Sistema de Registro de Demandas do Cidadão.

ROC Característica de Operação do Receptor.

SFN Sistema Financeiro Nacional.

TF-IDF Term Frequency–Inverse Document Frequency.

XGBoost Extreme Gradient Boosting.

Capítulo 1

Introdução

Na sociedade moderna, dados textuais podem ser encontrados em diversas fontes diferentes, a exemplo de postagens em redes sociais, conversas em aplicativos de mensagem instantânea e publicações em fóruns de notícias. Na Administração Pública Brasileira, essa tendência também é observada, comportando desde e-mails corporativos e documentos oficiais até textos de reclamações encaminhadas pelos cidadãos. Embora consistam em uma fonte rica de informações, os textos - diferentemente de dados tabulares, que podem ser organizados em uma tabela, a exemplo de registros categóricos e numéricos - devem ser estruturados para então produzirem conhecimento útil [1, 2, 3].

Com a explosão de informações vivenciada nos últimos anos, o processamento manual de textos tem se tornado dispendioso e demorado, além de sujeitar o processo a riscos decorrentes do fator humano, como fadiga, negligência e imperícia. Quando executada por servidores públicos, essa tarefa pode consumir recursos que poderiam ser alocados em serviços mais produtivos para a sociedade. Nesse contexto, foram exploradas alternativas visando a sua automatização, de modo a produzir resultados mais rápidos, confiáveis e efetivos. Assim, técnicas de Processamento de Linguagem Natural, ou *Natural Language Processing* (NLP) em inglês, passaram a ser utilizadas para permitir que computadores analisassem e estruturassem dados textuais, liberando tempo para humanos realizarem tarefas mais complexas e criativas [4].

Originalmente, métodos tradicionais de NLP foram utilizados, envolvendo o processamento inicial do texto para obter representações numéricas, seguido da extração de variáveis - ou atributos - a serem passadas para algoritmos clássicos de Aprendizado de Máquina - em inglês *Machine Learning* (ML) - treinados para a tarefa de interesse. Nessa abordagem, costumam ser adotadas representações textuais mais simplificadas, geralmente pautadas na frequência com que as palavras aparecem no texto. Ademais, a seleção das variáveis demanda uma robusta engenharia de atributos realizada por especialistas no domínio [5].

Na última década, todavia, houve uma migração para métodos baseados em Aprendizagem Profunda, ou *Deep Learning* (DL) em inglês, nos quais modelos de redes neurais são treinados para aprender a extrair e compor as variáveis relevantes automaticamente a partir dos dados [6, 7]. Atualmente, têm sido muito adotadas estratégias voltadas para a aplicação de técnicas de transferência de aprendizado a partir de um Modelo de Linguagem - em inglês *Language Model* (LM) - pré-treinado em uma grande fonte de dados textuais, seguida do ajuste do modelo para a tarefa desejada [8, 9]. Dessa forma é possível explorar representações textuais mais enriquecidas, como os *embeddings* contextuais, que, além de considerarem aspectos semânticos e sintáticos da língua, ainda permitem a desambiguação de palavras homônimas, isto é, cujo significado pode variar de acordo com o contexto [10, 11].

O *Bidirectional Encoder Representations from Transformers* (BERT), modelo proposto em 2018 [12], viabilizou a utilização dessa estratégia em diversos domínios e línguas para diferentes tarefas de NLP. Em âmbito nacional, a título de exemplo, esse modelo foi pré-treinado em um enorme conjunto de dados em português, conhecido como brWaC, dando origem ao BERTimbau [13], que obteve desempenho de estado-da-arte nas tarefas de similaridade semântica, inferência textual e reconhecimento de entidades nomeadas para dois grandes *benchmarks* em português, quais sejam o ASSIN2 e o First HAREM/MiniHAREM [14].

Contudo, a aplicação dessa solução para resolver problemas do mundo real ainda tem se mostrado um grande desafio, principalmente em tarefas envolvendo o processamento de grandes entradas de texto [15, 16, 17]. Adicionalmente, a literatura é ainda escassa quanto à eficácia de técnicas multimodais que permitam a utilização de *embeddings* contextuais junto de outras modalidades de variáveis - como as tabulares - que podem também se mostrar relevantes para o problema em questão [1, 3]. Diante do exposto, esta pesquisa buscou implementar um modelo de classificação das reclamações abertas pelos clientes e usuários do Sistema Financeiro Nacional (SFN), por meio da utilização de métodos baseados em DL, com desempenho superior ao da solução automatizada atualmente vigente, desenvolvida pelo Banco Central do Brasil (BCB) com base no método tradicional, apresentada nas seções seguintes.

1.1 Contextualização de negócio

Compete ao BCB exercer a fiscalização das instituições financeiras e outras entidades supervisionadas no âmbito do SFN, ficando estas - referidas, na presente dissertação, apenas como Instituição Financeira (IF) - obrigadas a fornecer, na forma por ele determinada, os dados ou informes julgados necessários para o fiel desempenho de suas atribuições [18].

Nesse contexto, e diante da responsabilidade do Banco Central, delegada regimentalmente ao Departamento de Atendimento Institucional (Deati), de atender o cidadão na solicitação de informações correlatas e na apresentação de reclamações contra produtos e serviços oferecidos pelas aludidas instituições, bem como de monitorar o atendimento de suas demandas [19], foi implementado o Sistema de Registro de Demandas do Cidadão (RDR), destinado ao registro e ao tratamento de denúncias, reclamações e pedidos de informações [20].

Em 2018 foram registrados 488 mil atendimentos ao cidadão, com predominância dos registros de reclamação representando aproximadamente 60% do total [21], com os mesmos números permanecendo em 2019 [22]. O processo de tratamento dessas reclamações é integralmente feito de forma manual no Deati, e consiste, sinteticamente, no seu enquadramento como "procedente" ou "improcedente", a partir da análise do relato do cidadão, da resposta da Instituição Financeira (IF), de eventuais documentos anexados à demanda e de julgamento sobre se houve ou não indício de descumprimento da regulamentação vigente, envolvendo a escolha da capitulação mais adequada no caso das demandas encerradas como procedentes. Para essas atividades, o Deati aloca 46% da sua força de trabalho e aproximadamente um terço dos custos totais dispendidos no atendimento ao cidadão [23]. No entanto, com o recente aumento do volume de demandas, atualmente só é possível analisar 60% das reclamações contra IFs encaminhadas ao BCB pelos cidadãos.

No âmbito da governança da administração pública federal direta, autárquica e fundacional, atualmente lidando com limitações de recursos e mudança de prioridades, faz-se necessário encontrar soluções tempestivas e inovadoras na busca de resultados para a sociedade [24]. No BCB essa situação se torna ainda mais relevante devido ao atual quadro de redução da força de trabalho da autarquia, em função de aposentadorias e falta de perspectivas de concursos públicos nos próximos anos. Portanto, aprimorar o uso da tecnologia é um movimento necessário e urgente para que o processo de atendimento fique mais rápido e eficiente [25]. Aliado a isso, há um processo em curso de transformação digital do governo que tem como objetivos, dentre outros, reduzir os custos e melhorar a qualidade dos serviços prestados à sociedade [26, 27].

Dessa forma, em um cenário de restrição orçamentária e redução de pessoal, combinado com constantes avanços tecnológicos, e em observância às disposições normativas mandatórias que tratam do assunto, foi instituído o projeto corporativo "Cidadania Digital", iniciado em junho de 2020, para aprimorar a estratégia de transformação digital do atendimento do BCB [25]. Dentre suas entregas, foi previsto o "tratamento automatizado de reclamações", com objetivo de automatizar tarefas do tratamento de reclamações, permitindo o aperfeiçoamento dos processos do Deati e a realocação de parte da sua força de trabalho em outras atividades essenciais realizadas pelo departamento.

1.2 Definição do problema

Diversos projetos de Inteligência Artificial, ou *Artificial Intelligence* (IA) em inglês, foram implementados no âmbito do "tratamento automatizado de reclamações". Contudo, o problema focado nesta dissertação é voltado para a construção de índice - denominado Índice de Procedência de Reclamações (IPDR) pelos especialistas do BCB - para filtrar as reclamações registradas por cidadãos contra IFs com maior possibilidade de serem procedentes, direcionando, assim, os esforços de análise e julgamento promovidos pelos servidores do Deati, diante da atual impossibilidade do departamento em avaliar todas as demandas.

Do ponto de vista de negócio, o problema em questão pode ser definido como a pré-seleção das demandas com maior chance de serem procedentes, ou seja, de se referirem a casos em que houve indício de inobservância de obrigações regulatórias pelas entidades supervisionadas. Sob uma perspectiva mais computacional, trata-se de um problema de classificação, no qual um classificador deve receber dados textuais, quais sejam o relato do cidadão e a resposta da entidade reclamada, e retornar uma pontuação que permita ranquear as demandas levando em consideração os rótulos "procedente" ou "improcedente".

1.3 Hipóteses de pesquisa e objetivos

Por ocasião do projeto Índice de Procedência de Reclamações (IPDR), foi desenvolvido pelo BCB, no ínicio de 2022, um classificador binário automatizado treinado sobre os dados do RDR. Para isso, foi adotado método tradicional de NLP, contando com uma representação textual simplificada, baseada na frequência ponderada das palavras, denominada *Term Frequency–Inverse Document Frequency* (TF-IDF), e um algoritmo de ML com arquitetura de *Gradient Boosting Decision Tree* (GBDT), conceituado na literatura por suas habilidades de processamento de variáveis tabulares (categóricas ou numéricas), conhecido como *Light Gradient Boosting Machine* (LightGBM) [28, 29].

A solução proposta recebe, como entrada, duas variáveis textuais, a reclamação do cidadão demandante e a resposta da IF. Em adição, como resultado da engenharia de atributos baseada em conhecimentos de especialistas de negócio, característica de abordagens tradicionais, foram também definidas 16 variáveis tabulares voltadas para aspectos contextuais - isto é, informações relacionadas ao contexto do cidadão e da entidade supervisionada no momento da reclamação - ou geradas a partir das variáveis textuais retromencionadas.

Como o modelo foi desenvolvido recentemente, ainda não há registros de sua efetividade nos processos de trabalho do BCB. No entanto, inspirado pelos fortes avanços

ocorridos recentemente no campo de NLP com os métodos de DL, principalmente no tocante à seleção automatizada das variáveis pelo modelo e à utilização de representações mais enriquecidas para o texto, o presente estudo buscou a implementação de modelo com melhor desempenho para a tarefa descrita, no intuito de aprimorar ainda mais o tratamento de reclamações do cidadão.

1.3.1 Hipóteses de pesquisa

Tendo como referência o problema apresentado, a solução desenvolvida pelo BCB e as tecnologias atualmente disponíveis, foram levantadas as seguintes hipóteses de pesquisa:

H1: A utilização das variáveis tabulares levantadas pelo BCB em adição às variáveis de texto não contribui para a tarefa de classificação das demandas abertas pelos cidadãos.

Os conhecimentos mais relevantes para a classificação desejada possivelmente se encontram nos atributos de texto, uma vez que compreendem tanto a reclamação do cidadão quanto a resposta da IF, que configuram as principais fontes analisadas no processo de tratamento das demandas. As variáveis tabulares, por outro lado, parecem ter sido selecionadas para fornecer informações acessórias para o modelo, recorrendo a aspectos contextuais ou buscando extrair conhecimentos dos textos. Todavia, a adição dessas variáveis pode atuar em detrimento do aprendizado do modelo caso introduza informações que não agregam para o problema, prejudicando o desempenho do classificador, diferentemente de métodos de DL, em que o próprio modelo aprende a selecionar os atributos mais relevantes durante a etapa de treinamento.

H2: A utilização do *Bidirectional Encoder Representations from Transformers* (BERT) permite a construção de um classificador para as demandas abertas pelos cidadãos com melhor desempenho do que o desenvolvido pelo BCB.

Como o texto é um dado não estruturado, são necessárias transformações de modo a propiciar o aprendizado pelo algoritmo de ML. Portanto, a representação a ser utilizada é crucial para tarefas de NLP. No caso em tela, diante da potencial relevância dos atributos textuais para o tratamento das demandas, essa questão ganha ainda maior destaque. Não obstante, a representação utilizada no classificador do BCB (TF-IDF), considera apenas a frequência das palavras - ou termos - em um documento ponderada por sua recorrência nos demais, apresentando, consequentemente, duas grandes limitações: (i) a ordem sequencial das palavras no texto é descartada, o que resulta na perda de contexto; (ii) não são levados

em consideração os valores semânticos das palavras, de modo que cada termo é assumido como independente dos demais [10].

Em abordagens de DL, é possível capturar informações semânticas e sintáticas das palavras por meio de *embeddings* contextualizados, que permitem ainda a representação de diferentes significados de uma palavra de acordo com o contexto em que ela aparece [5, 6]. Nesse sentido, o BERT - LM pré-treinado baseado em arquitetura de transformadores que alcançou o estado-da-arte em diversas tarefas de NLP, com fácil extensão para outros domínios e línguas - pode produzir representações textuais que levam em consideração tanto a posição quanto o contexto das palavras [12, 13]. Dessa forma, sua utilização na classificação das demandas pode resultar em um modelo com maior desempenho.

1.3.2 Objetivos

A presente pesquisa teve como objetivo geral o desenvolvimento de um modelo de classificação binária - no âmbito do tratamento de reclamações abertas pelo cidadãos contra IFs - com melhor desempenho do que o atualmente vigente. Considerando que a atual solução foi resultado de diversos estudos e experimentos conduzidos pelo BCB abarcando as principais representações textuais e modelos clássicos referenciados na literatura no âmbito de abordagens tradicionais, foram ainda estabelecidos, para esta pesquisa, os seguintes objetivos específicos:

- Estudar e reproduzir o modelo do BCB, a ser usado como linha-de-base e referência para o estudo;
- Desenvolver novos modelos com base nas hipóteses de pesquisa **H1** e **H2**; e
- Comparar os modelos desenvolvidos com o modelo do BCB no intuito de selecionar aquele com melhor desempenho.

1.4 Organização do trabalho

Esta dissertação foi dividida em 8 capítulos. O Capítulo 1 introduz a pesquisa, contextualizando e definindo o problema e estabelecendo as hipóteses e objetivos do trabalho. No Capítulo 2, é apresentada uma fundamentação teórica com os principais conceitos envolvidos e no Capítulo 3 são elencados trabalhos relacionados ao objeto de estudo. O Capítulo 4 contém o entendimento do negócio e a seleção e análise dos dados utilizados nos experimentos. Em seguida, o Capítulo 5 trata da reprodução do modelo do BCB, enquanto no Capítulo 6 são apresentados os modelos propostos. Por fim, o Capítulo 7 discorre sobre os resultados obtidos com os experimentos e o Capítulo 8 conclui o trabalho e aponta estudos futuros.

Capítulo 2

Referencial Teórico

Este capítulo apresenta os principais conceitos relacionados à classificação de texto com foco em abordagens baseadas no aprendizado de máquina, abrangendo tanto modelos tradicionais quanto de aprendizagem profunda. São também apresentadas as principais métricas de desempenho para avaliação dos classificadores.

2.1 Classificação de texto

A classificação de texto, também conhecida como categorização de texto, pode ser definida como a atribuição de uma ou mais classes ou categorias a uma sequência contígua de palavras, como uma frase, um parágrafo ou um documento [30]. Dessa forma, para uma coleção de m documentos (corpus), $D = \{d_1, d_2, ..., d_m\}$, e um conjunto de n categorias, $C = \{c_1, c_2, ..., c_n\}$, a classificação de texto consiste em atribuir, para cada documento em D, uma ou mais classes em C. Tarefas que envolvem apenas uma categoria por documento, chamadas de *single-label*, podem ser divididas em binárias (com apenas dois rótulos) ou *multi-class* (com mais de dois rótulos). Em contrapartida, em problemas de classificação *multi-label*, cada documento pode ser associado a mais de uma categoria [31, 7].

Trata-se de problema clássico de NLP, com uma ampla gama de aplicações, incluindo análise de tópicos, resposta a perguntas, detecção de *spam*, análise de sentimentos e categorização de notícias. A classificação de texto pode ser realizada de duas formas: manual ou automática. A primeira envolve um anotador humano, que interpreta o conteúdo do texto e o categoriza de acordo. Embora esse método possa fornecer bons resultados, é normalmente demorado e custoso. A segunda forma, por sua vez, aplica técnicas guiadas por IA para classificar automaticamente o texto de maneira mais rápida, econômica e precisa [32, 33].

2.2 Aprendizado de máquina

Aprendizado de máquina (ML) é um ramo da IA que estuda métodos computacionais para adquirir novos conhecimentos, habilidades e meios de organizar o conhecimento já existente. Para isso, são aprimorados algoritmos computacionais por meio de aprendizado automatizado decorrente da experiência adquirida a partir de uma série de observações. Esse aprendizado pode ser categorizado em supervisionado, não-supervisionado e por reforço, dependendo da disponibilidade de dados rotulados [34, 35].

Na aprendizagem supervisionada, para cada instância existe um rótulo que indica a qual categoria a observação pertence. Por outro lado, algoritmos de aprendizagem não-supervisionada lidam com dados não rotulados, buscando explorar e aprender suas propriedades de modo a executar tarefas como clusterização e associação. Por fim, no aprendizado por reforço, adota-se a lógica da "tentativa e erro" para que o sistema aprenda quais ações são as "melhores" a serem tomadas [36, 35].

No âmbito da classificação automatizada de textos, a utilização de técnicas baseadas em ML usualmente remete a um aprendizado supervisionado, no qual o algoritmo busca aprender associações inerentes entre textos e seus rótulos, tendo como referência um conjunto de dados (*dataset*) contendo observações pré-rotuladas. No entanto, para que o algoritmo consiga processar as entradas, estas devem estar em um formato apropriado para o aprendizado. Consequentemente, dados não estruturados, como os textos, requerem um pré-processamento - utilizando aplicações de NLP - antes de serem concebidos pelo classificador [30, 32].

A maioria dos modelos clássicos de ML, denominados "Método Tradicionais" em [5], segue um processo de duas etapas para classificação de texto. Primeiramente, variáveis - também chamadas de atributos ou *features*, em inglês - são extraídas manualmente dos documentos. Em seguida, estas são alimentadas no algoritmo para a tarefa de categorização. Portanto, a eficácia desse método está diretamente associada ao processo de obtenção das variáveis, que, todavia, demanda tempo e conhecimento do negócio, dificultando a utilização dos classificadores em outras tarefas.

Buscando superar essas limitações, foram explorados novos métodos baseados em redes neurais que, diferentemente dos classificadores tradicionais, integram a engenharia de variáveis no processo de ajuste do modelo, por meio do aprendizado de um conjunto de transformações não lineares que servem para mapear os atributos diretamente às saídas do classificador. A esses métodos foram aribuídos diferentes nomes, como "Abordagem neural" em [32] e "Método Baseado em representação neural" em [37]. Contudo, as nomenclaturas mais utilizada na literatura remetem a modelos baseados em Aprendizagem Profunda (DL) [38, 39, 9, 6], enquadrados especificamente como "Método de Aprendi-

zagem Profunda" em [5]. A Figura 2.1 apresenta, de forma esquematizada, o fluxo de classificação de texto com esses dois métodos, melhor abordados nas seções seguintes.

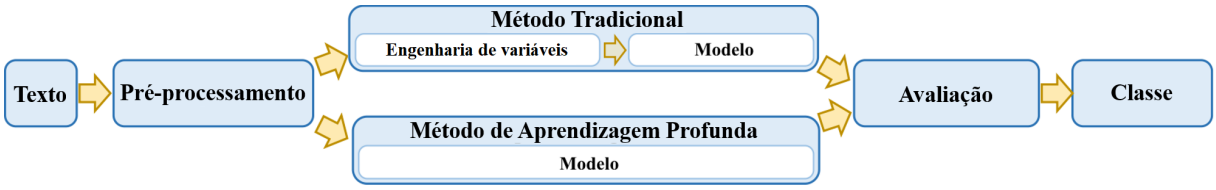


Figura 2.1: Fluxo de classificação de texto (Fonte: adaptado de [5]).

2.3 Método tradicional

Até a década de 2010 os modelos tradicionais prevaleceram para tarefas de classificação de texto. Esse método envolve, primeiramente, o pré-processamento dos dados textuais brutos das entradas, recorrendo a segmentação de palavras, limpeza dos dados e levantamentos estatísticos. Em seguida, é escolhida uma representação textual para expressar o texto pré-processado de forma propícia para o processamento computacional, minimizando a perda de informações [5, 6].

2.3.1 Representação textual

Em métodos tradicionais, normalmente são adotadas representações mais simplificadas do texto, desconsiderando aspectos sêmanticos ou sintáticos da língua. Uma representação amplamente adotada, servindo como linha-de-base, é o saco-de-palavras, *Bag-of-Words* (BOW) em inglês, que consiste em uma matriz na qual os documentos são representados por linhas e suas palavras por colunas, contendo as frequências de incidência desses termos no corpus.

Essa matriz pode ainda ser representada por meio da ponderação TF-IDF (*Term Frequency–Inverse Document Frequency*), na qual as frequências das palavras são normalizadas, penalizando-se aquelas que ocorrem com maior frequência na coleção de documentos [9, 35]. Em outras palavras, é uma medida para indicar a relevância $w_{t,d}$ de um termo t em um dado documento d, sendo calculada por meio da Equação 2.1, em que $tf_{t,d}$ representa a frequência do termo t no documento d, N o número de documentos e df_t a frequência de documentos contendo o termo t [40].

$$w_{t,d} = tf_{t,d}.log\left(\frac{N}{df_t}\right), \tag{2.1}$$

Em que pese essas representações serem bastante eficientes e simples, elas não consideram a ordem sequencial das palavras nem as relações semânticas e sintáticas entre elas. Ademais, ainda que a utilização de n-gramas em vez de unigramas (palavras) mitigue a aludida invariância à ordem dos termos, esse processo acaba produzindo matrizes ainda mais esparsas, agravando o problema da maldição da dimensionalidade, característico dessas representações, em que o aumento da quantidade de atributos resulta na queda do desempenho do modelo [31, 41].

Nas últimas décadas, representações mais modernas foram desenvolvidas. Ao tratar palavras como unidades atômicas associadas a índices de um vocabulário, estas podem ser mapeadas em uma representação vetorial densa, conhecida como *word embedding*, tendo como principais aplicações o Word2Vec, o GloVe e o FastText [42, 43]. No entanto, como esse mapeamento é normalmente feito treinando redes neurais - com os *embeddings* constituindo parâmetros do modelo, sendo, portanto, otimizados e aprendidos durante o processo -, essas representações costumam ser adotadas precipuamente em métodos de apendizagem profunda, tratados na Seção 2.4 [14, 41].

2.3.2 Algoritmos tradicionais para classificação

Uma vez assumindo a representação escolhida, os textos são passados como entradas para o classificador. Nesta etapa, a escolha do algoritmo de ML a ser utilizado vai depender do conjunto de dados rotulados e do problema a ser resolvido [33]. Segundo [44], os classificadores de texto podem ser agrupados em famílias, de acordo com seu algoritmo base. No âmbito do método tradicional, essas famílias enolvem (i) Modelos Lineares, que estimam a linha que melhor se ajusta aos dados; (ii) Máquinas de Vetor de Suporte (SVM), que mapeiam as observações para pontos no espaço e então procuram o hiperplano de separação ideal entre as classes de modo a maximizar a largura do intervalo entre seus pontos adjacentes; (c) Classificadores de vizinhos mais próximos (KNN), que categorizam os documentos com base nos votos de seus vizinhos mais próximos; (d) Classificadores Baeysianos (Naïve Bayes), que adotam o teorema de Bayes, assumindo independência entre as variáveis; (e) Árvores de Decisão (DT), nas quais os atributos são representados por nós em um modelo em forma de árvore; e (f) *Bagging* e *Boosting Ensembles*, que combinam vários estimadores base, também conhecidos como aprendedores fracos (*weak learners*).

Frequentemente, esses algoritmos são implementados em conjunto, no intuito de definir o modelo que melhor se adequa aos dados e à tarefa de classificação desejada, como é o caso em [31], para a categorização de artigos de um jornal, em [35], para a triagem de denúncias e [45], para a automatização de rotinas de relacionamento com o cliente. Contudo, estudos comparativos recentes têm sido conduzidos buscando aferir as vantagens e desvantagens

de cada classificador, avaliando sua performance em diferentes conjuntos de dados de referência (*benchmarks*). De modo geral, essas pesquisas apontam que algoritmos de *ensembles* - principalmente os baseados em *boosting* - apresentam melhor desempenho para aplicações de classificação de texto quando comparados aos demais algoritmos de ML [44, 46, 34, 39].

Nesse contexto, métodos baseados em árvores de decisão impulsionadas por gradiente, ou *Gradient Boosting Decision Tree* (GBDT), em inglês, têm se mostrado bastante populares, devido a sua eficiência, performance e interpretabilidade, alcançando resultados de estado-da-arte em muitas tarefas de aprendizagem de máquina, a exemplo da categorização de textos, e sendo frequentemente empregados em competições como as promovidas pelas empresas Kaggle e Netflix [29, 47, 28]. Diante da relevância desses algoritmos e levando em consideração que o classificador construído pelo BCB - um dos objetos de estudo desta dissertação - é baseado em GBDT, conforme mencionado na Subseção 1.2, este será melhor abordado no presente referencial teórico.

2.3.3 *Gradient Boosting Decision Tree* (GBDT)

GBDT é um modelo de *ensembles* de árvores de decisão. Em outras palavras, utiliza múltiplas DTs como aprendedores fracos para construir um forte. Ao contrário do método de *bagging* (abreviação de *bootstrap aggregating*), no qual os estimadores bases são feitos de modo independente, nas técnicas de *boosting* os aprendedores são construídos em sequência, minimizando iterativamente o erro dos modelos anteriores, emulando, portanto, um aprendizado sequencial [48]. Em GBDTs, o modelo é treinado de forma aditiva, combinando o resultado de M árvores $(f_1, f_2, ..., f_M)$ em etapas consecutivas, como descrito na Equação 2.2, com cada novo estimador sendo ajustado sobre os gradientes negativos da função de perda (também conhecidos como resíduos), o que permite a construção de um forte estimador ao final [29, 47].

$$F(x) = \sum_{m=1}^{M} f_m(x) \tag{2.2}$$

As funções de perda aplicadas podem ser arbitrárias, geralmente ficando a critério do pesquisador. Em tarefas de classificação binária, por exemplo, é normalmente aplicada a entropia cruzada (também conhecida como perda logística ou *log loss*). No entanto, para fornecer uma melhor intuição, caso a função de erro adotada seja o clássico erro quadrático médio (MSE), calculado pela Equação 2.3, em que L é a função de perda que mensura a variação entre os valores reais (y_i) e os preditos $(\hat{y}_i)$ em um cojunto de n observações, o procedimento de aprendizado, considerando a derivada de primeira ordem, resultaria nítidamente em ajustes consecutivos dos resíduos [39, 28].

$$L = \sum_{i}^{n} (y_i - \hat{y}_i)^2 \tag{2.3}$$

A Figura 2.2 esquematiza a construção iterativa de árvores de decisão em um processo de *gradient boosting* para predições de um modelo GBDT. Sob essa estrutura, dois poderosos algoritmos foram propostos recentemente, quais sejam o *Extreme Gradient Boosting* (XGBoost) e o *Light Gradient Boosting Machine* (LightGBM), sendo considerados o estado-da-arte desse grupo, uma vez que permitem, até certa extensão, evitar problemas de sobreajuste (*overfitting*) dos modelos e efetuar seu treinamento com maior eficiência computacional.

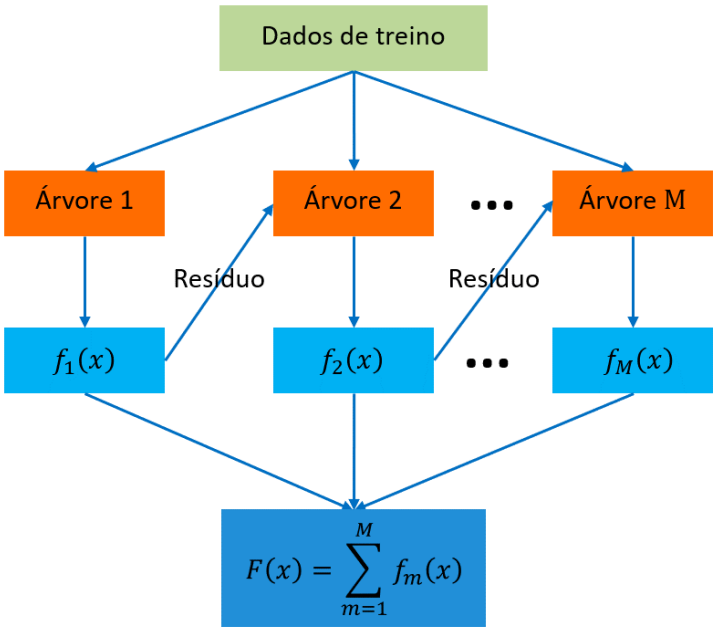


Figura 2.2: Diagrama do algoritmo GBDT (Fonte: adaptado de [48]).

XGBoost

XGBoost é uma implementação de GBDT, proposta em [47], altamente escalável, flexível e versátil, sendo projetado para explorar as variáveis corretamente e processar grandes conjuntos de dados. Um característica notória desse algoritmo é que ele introduz o termo de regularização na função objetivo para evitar o sobreajuste e pode usar automaticamente a CPU para computação paralela *multi-thread*, selecionando os *cores* disponíveis durante a construção das árvores [47, 49].

LightGBM

Esse algoritmo de GBDT foi desenvolvido em [29], tendo como linha-de-base o XGBoost, buscando reduzir o seu tempo de implementação. Seu grande diferencial reside na orientação vertical de crescimento das árvores (*leaf-wise*), em contraste com os modelos anteriores, que cresciam horizontalmente (*level-wise*), como ilustra a Figura 2.3. Como resultado, obtém-se uma convergência mais rápida durante o treino, porém mais suscetível ao sobreajuste. Para a seleção das partições nos nós das árvores, o LightGBM utiliza um método baseado em histograma, que envolve a ordenação e o agrupamento dos atributos em "caixas" (*bins*). Um algoritmo de amostragem, chamado *Gradient-based One-Side Sampling* (GOSS), é utilizado para indicar a importância das instâncias do *dataset*, priorizando dados com maior gradiente. Adicionalmente, a técnica de *Exclusive Feature Bundling* (EFB) é adotada para reduzir o número de variáveis, por meio do agrupamento de atributos mutuamente exclusivos (isto é, que raramente assumem valores diferentes de zero simultaneamente) [29, 50].

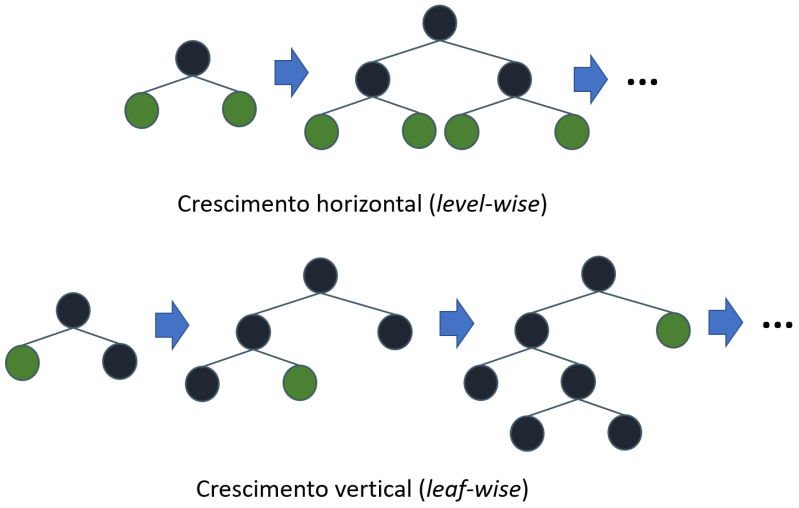


Figura 2.3: Orientação de crescimento das árvores (Fonte: adaptado de [50]).

2.4 Método de aprendizagem profunda

A Aprendizagem Profunda, ou *Deep Learning* (DL) em inglês, é uma subárea do ML voltada para o estudo de algoritmos baseados em redes neurais artificiais, que simulam o cérebro humano para, automaticamente, aprender atributos relevantes a partir dos dados, proporcionando bons resultados em tarefas como compreensão textual. Quando compostas por várias camadas, essas redes podem ser chamadas de Redes Neurais Profundas, em

inglês *Deep Neural Networks* (DNN), resultando em estruturas de alta complexidade objetivando um aprendizado orientado a dados. Na literatura atual, a maioria das pesquisas voltadas para a classificação de texto são baseadas em DNN [35, 5].

A partir da década de 2010, a tarefa de categorização de textos gradualmente migrou de métodos clássicos para métodos de aprendizagem profunda, com os primeiros classificadores de DNN sendo propostos como alternativa aos modelos de duas etapas, que combinam uma pesada engenharia de variáveis com algoritmos tradicionais de classificação. Destarte, enquanto a abordagem anterior requeria um rico conjunto de atributos projetados à mão por especialistas no domínio, o método de DL propõe um pré-processamento mínimo das entradas e o treinamento de um modelo de ponta-a-ponta para aprender a extrair e compor as variáveis relevantes automaticamente a partir dos dados [6, 14].

2.4.1 Representação textual

Na abordadem de DL, um componente crucial para o NLP é a representação dos textos [10]. Em modelos baseados em redes neurais, as entradas geralmente são em forma de *word embeddings*, nas quais palavras são mapeadas de um vocabulário para um espaço vetorial latente em que elementos com significados semelhantes estão próximos, sumarizando, assim, informações sintáticas e semânticas de cada palavra [43]. Nesse contexto, um importante marco na evolução desses modelos foi a inicialização de *word embeddings* com representações pré-treinadas de tarefas não supervisionadas. Ao utilizar vastos conjuntos de documentos para treinar os *embeddings*, estes conseguem melhor capturar aspectos de linguagem, propiciando a inicialização do modelo com representações mais ricas das palavras, o que pode aumentar seu desempenho quando comparado com a inicialização aleatória [14].

Todavia, essa representação textual ainda não considera o significado das palavras no contexto em que se inserem [10]. Assim, atualmente, o foco das pesquisas tem migrado dos tradicionais *word embeddings* para os denominados *embeddings* contextuais, obtidos aproveitando-se os estados internos de Modelos de Linguagem (LM) - que consistem em calcular a probabilidade da ocorrência de um conjunto de termos em uma determinada sequência - para extrair representações mais ricas das palavras no contexto, permitindo, portanto, a desambiguação de termos polissêmicos [51, 11].

Como mencionado anteriormente, no método de DL, o modelo busca extrair as variáveis relevantes do texto, tratando-as como parâmetros a serem otimizados durante o treinamento para uma tarefa específica de NLP. Isso implica ter que aprender do zero ("*from scratch*") todos os parâmetros do modelo usando os dados rotulados disponíveis, o que dificulta a aplicação de grandes modelos em cenários de escassez de observações e pode levar ao sobreajuste no *dataset* de treinamento [7].

Diante desse entrave, uma prática muito empregada na literatura envolve a aplicação de técnicas de transferência de aprendizado, ou *Transfer Learning* em inglês, que consiste em pré-treinar um modelo em uma fonte de dados rica para uma determinada tarefa e efetuar o seu ajuste-fino (*fine-tuning*) na tarefa de interesse. A esse respeito, estratégias envolvendo o ajuste de LMs pré-treinados em grandes corpus têm produzido resultados de estado-da-arte em diferentes tarefas de NLP, com excelente desempenho em classificação de textos. A utilização de técnicas de transferência de aprendizagem permite ainda reduzir a quantidade de observações rótuladas necessárias para o aprendizado supervisionado da tarefa à jusante, ou seja, que de fato se deseja aplicar [13].

2.4.2 Redes neurais profundas para classificação

Inúmeros modelos de DL foram propostos nos últimos anos para tarefas de NLP. Notadamente para a categorização de textos, as aborgadens precursoras apresentaram modelos baseados em Redes Neurais Recursivas (ReNN) - que podem, recursivamente, aprender a semântica do texto e a estrutura em árvore da sintaxe, sem demandar uma engenharia de *features* nas entradas - e em perceptrons multicamadas (MLP) - que envolvem estruturas simples de redes neurais usadas para capturar as variáveis automaticamente, constituídas de camadas de entrada, ocultas (com funções de ativação), e de saída. Posteriormente, outras abordagens mais sofisticadas foram desenvolvidas, como as Redes Neurais Convolucionais (CNN), as Redes Neurais Recorrentes (RNN) e os mecanismos de atenção [32, 5].

As CNNs foram originalmente propostas para a classificação de imagens, com filtros convolutivos para extrair suas características, se estendendo à aplicações de NLP por meio da aplicação simultânea de convoluções definidas por diferentes *kernels* a vários segmentos de uma sequência de texto. RNNs, por sua vez, são amplamente empregadas para processar dados sequenciais por meio de computação recorrente. Assim, para a classificação de texto, LMs baseados em RNN aprendem informações históricas considerando aspectos de localização entre todas as palavras apropriadas para a tarefa. Buscando aumentar a capacidade de interpretabilidade das abordagens anteriores, foram propostos mecanismos de atenção, que permitem que o modelo "preste atenção" às entradas de forma diferenciada. Ou seja, o modelo aprende a mensurar a contribuição das palavras e sentenças para o julgamento de classificação [5, 6].

Em um contexto de forte alavancagem de modelos de DL, o surgimento do *Bidirectional Encoder Representations from Transformers* (BERT), que pode gerar os chamados *embeddings* contextualizados mencionados na Subseção 2.4.1, se mostrou um marco significativo para o desenvolvimento das tecnologias de processamento de linguagem natural. Muitos pesquisadores estudaram modelos baseados no BERT, que alcançaram melhor desempenho do que as abordagens retromencionadas em várias tarefas de NLP, incluindo

classificação de texto. Ademais, com o advento de métodos de transferência de aprendizado, tornou-se possível pré-treinar LMs derivados do BERT em outros corporas, com o aprendizado adquirido sendo disponibilizado para o seu ajuste-fino em tarefas à jusante, alavancando, dessa maneira, pesquisas em línguas com escassez de *datasets* rótulados [52]. A título exemplificativo, [13] treinaram o BERT em um grande conjunto de dados em português, alcançando resultados de estado-da-arte para três tarefas de NLP em *datasets* considerados *benchmark*s da língua.

Como o próprio nome descreve, o BERT é baseado na arquitetura de transformadores, que aplica técnicas de auto-atenção para capturar a influência de cada uma das palavras nas demais. Embora esse processo seja conduzido de forma paralelizada, tornando possível pré-treinar grandes LMs, o mecanismo de atenção pode ser computacionalmente pesado, especialmente ao lidar com textos longos. Consequentemente, o BERT fixa um limite para o tamanho de suas entradas, dificultando o seu uso em categorização de documentos de maior extensão. Para contornar essa limitação, alguns trabalhos recentes, como o Reformer [53] e o LongFormer [16], buscaram adaptar o BERT com técnicas alternativas para o mecanismo de atenção [54]. Todavia, até o momento, não foram encontradas extensões ou derivações desses modelos para a língua portuguesa.

Outras abordagens, menos sofisticadas, para superar esse problema envolvem a utilização de modelos hierárquicos, buscando explorar o relacionamento entre palavras, sentenças e texto. Assim, no primeiro nível, o documento é divido em pedaços (*chunks*) de menor comprimento - com ou sem sobreposição - para serem processados separadamente. Com isso, é possível utilizar *embeddings* contextuais gerados pelo BERT para representar as sentenças, aproveitando-se os conhecimentos adquiridos no pré-treinamento. No segundo nível, as saídas obtidas são processadas para proporcionar uma representação única do documento, a ser empregada para a tarefa de NLP de interesse [16, 15, 37]. Para isso, diferentes modelos podem ser escolhidos. Contudo, uma abordagem intuitiva e bastante popular tem sido a utilização da *Long short-term memory* (LSTM), uma implementação de RNN designada para aprender dependências de longo alcance, de modo a processar sequências levando em consideração o contexto completo das entradas. Embora seu alcance possa não comportar todo o conteúdo do documento a nível de palavras, o mesmo costuma ser suficiente para lidar com *chunks* de sentenças, que ocorrem em menor número [16, 55, 56].

Modelos hierárquicos combinando o BERT e a LSTM geralmente apresentam bons desempenhos para tarefas de classificação de texto, com custo-benefício satisfatório. Primeiramente, uma arquitetura de DNN na qual o modelo é treinado de ponta-a-ponta permite extrair e compor os atributos relevantes automaticamente a partir dos dados, dispensando a engenharia de variáveis baseada em conhecimentos profundos do domínio,

típica de abordagens tradicionais. Com o BERT, é ainda possível obter representações textuais enriquecidas - que levam em consideração aspectos sintáticos e semânticos das palavras em seu contexto -, ainda que segmentadas, por conta da limitação de tamanho das entradas. Adicionalmente, ao utilizar um LM já pré-treinado, é possível otimizar o processo de aprendizado para a tarefa à jusante. Por fim, a LSTM consegue capturar informações sequenciais a partir das representações a nível de sentença obtidas do BERT, retornando um *embedding* de todo o documento para ser utilizado para a classificação final. Para facilitar a compreensão desse modelo, as estruturas da LSTM e do BERT são apresentadas nas subseções seguintes.

2.4.3 *Long short-term memory* (LSTM)

Como mencionado anteriormente, as RNNs são as arquiteturas de DNN mais populares para processar dados em sequência, a exemplo de textos. Em comparação com outras redes neurais, a RNN é capaz de processar entradas de comprimento variável, usando seus estados internos. No processo de retropropagação dessas redes, os pesos são ajustados por gradientes, calculados por contínuas multiplicações de derivadas. Caso essas derivadas sejam muito pequenas, os parâmetros do modelo não podem ser atualizados através do gradiente descendente, resultando em um problema de desaparecimento de gradiente [51, 32].

A LSTM, considerada um aprimoramento das RNNs, efetivamente mitiga esse efeito, permitindo que as informações sejam retidas por mais tempo. Para isso, um mecanismo chamado "portões", ou *gates* em inglês, é empregado para controlar a retenção e o descarte de dados. Assim, na estrutura dessas redes, são definidos um portão de entrada ($\mathbf{i}_t$), um de saída ($\mathbf{o}_t$) e um de esquecimento ($\mathbf{f}_t$). Na camada da LSTM, as seguintes equações são computadas [7]:

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f\right) \tag{2.4}$$

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i\right) \tag{2.5}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o\right) \tag{2.6}$$

$$\tilde{\mathbf{C}}_t = tanh\left(\mathbf{W}_C \mathbf{x}_t + \mathbf{U}_C \mathbf{h}_{t-1} + \mathbf{b}_C\right) \tag{2.7}$$

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \tag{2.8}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot tanh\left(\mathbf{C}_t\right) \tag{2.9}$$

Nessa formulação, $\mathbf{x}$ é o vetor de entrada; $\mathbf{h}$ é o vetor do estado oculto; $\mathbf{b}$ o vetor de viés (bias); $\mathbf{C}$ é o vetor do estado da célula ou a "memória" da rede; $\tilde{\mathbf{C}}$ é um candidato a vetor do estado da célula; $\mathbf{W}$'s e $\mathbf{U}$'s são matrizes de pesos das entradas e dos estados ocultos, respectivamente; σ e $tanh$ são, na ordem, funções de ativações sigmóide e tangente hiperbólica; e o símbolo $\odot$ indica uma multiplicação por componentes (ou de Hadamard). Reitera-se que o processamento na LSTM ocorre de forma sequencial. Logo, em uma etapa t a rede recebe, da etapa anterior, informações do estado da célula ($\mathbf{C_{t-1}}$) e do estado oculto ($\mathbf{h_{t-1}}$), bem como uma entrada $\mathbf{x}_t$, retornando, como saída, a memória atualizada ($\mathbf{C}_t$) e o novo estado culto ($\mathbf{h}_t$), conforme ilustrado na Figura 2.4 [4, 7].

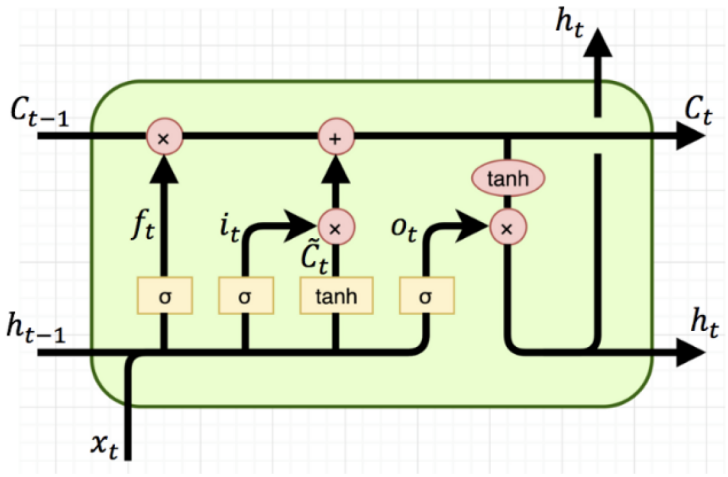


Figura 2.4: Camada de uma LSTM (Fonte: [4]).

Uma variação da LSTM comumente adotada para a classificação de textos é a *Bidirectional Long short-term memory* (BiLSTM), que se mostrou ser bem sucedida na captura de representações de linguagem comum devido à sua capacidade de capturar a ordem de palavras. A BiLSTM concatena as saídas de duas LSTMs para prever cada elemento na sequência. Essas duas redes têm direções de processamento de dados diferentes, com uma processando a sequência da esquerda para a direita (*forward*) e a outra da direita para a esquerda (*backward*). Dessa maneira, qualquer palavra em uma frase - ou, alternativamente, qualquer sentença em um documento, para modelos híbridos - pode ser predita com base nos elementos anteriores e seguintes. A Figura 2.5 representa a estrutura genérica de uma BiLSTM para categorização de documentos, recebendo nas entradas *embeddings* de texto, a nível de palavras ou sentenças, representados por x_0, x_1, ..., x_n [51, 57, 58].

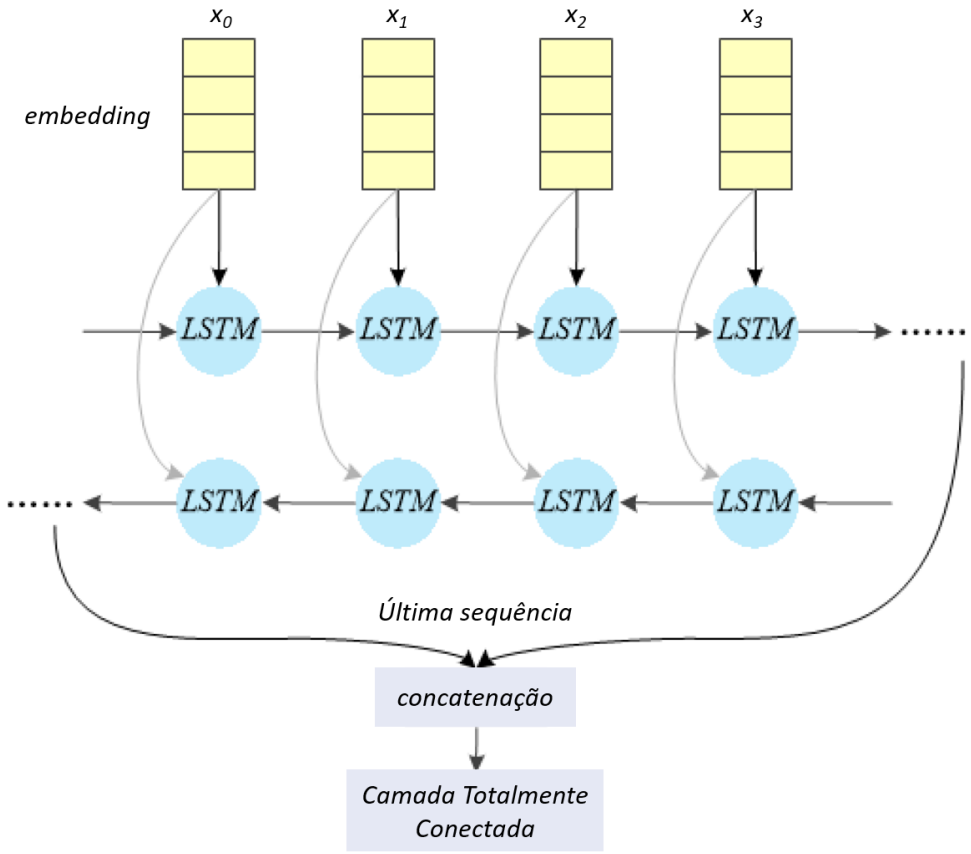


Figura 2.5: Estrutura da BiLSTM para classificação de texto (Fonte: adaptado de [57]).

2.4.4 *Bidirectional Encoder Representations from Transformers* (BERT)

BERT se refere a uma abordagem auto-supervisionada para pré-treinamento de camadas de transformadores, com posterior ajuste-fino em tarefas à jusante. Esse modelo usa duas técnicas não supervisionadas chamadas *Masked Language Modeling* (MLM) e *Next Sentence Prediction* (NSP). No MLM, elementos da sequência de entrada são aleatoriamente mascarados para prever o vocabulário original com base em seu contexto. Desta forma, é possível obter uma representação que funde o contexto à esquerda e à direita, permitindo o pré-treinamento de um transformador bidirecional profundo. O objetivo do NSP, por sua vez, é prever se uma dada sentença B é, de fato, a continuação de uma sentença A, ou se é uma sentença aleatória. Ao fazer isso, o modelo pré-treinado pode capturar relacionamentos à nível de sentença, o que é altamento benéfico para certas tarefas de NLP, a exemplo da classificação de textos [12, 11].

A arquitetura do BERT consiste, basicamente, em um codificador de transformador bidirecional multicamada, similar ao proposto em [59] como alternativa aos modelos con-

volucionais e recorrentes, até então predominantes em tarefas de NLP. Ao confiar apenas em mecanismos de auto-atenção em vez de recorrência, os transformadores podem ver todas as entradas de uma só vez e, portanto, modelar dependências em longas sequências em tempo constante, permitindo uma paralelização muito maior e viabilizando o treinamento de modelos mais profundos [12, 14]. O BERT foi pré-treinado em dois tamanhos diferentes: BERT *Base* ($L = 12$, $H = 768$, $A = 12$, $P = 110M$) e BERT *Large* ($L = 24$, $H = 1024$, $A = 16$, $P = 340M$), em que L é o número de camadas (ou seja, blocos de transformador), H é a dimensão oculta, A é o número de cabeças de auto-atenção e P o total de parâmetros. Além disso, um modelo BERT multilíngue (mBERT) foi também treinado em 104 idiomas usando a arquitetura do BERT *Base* [11].

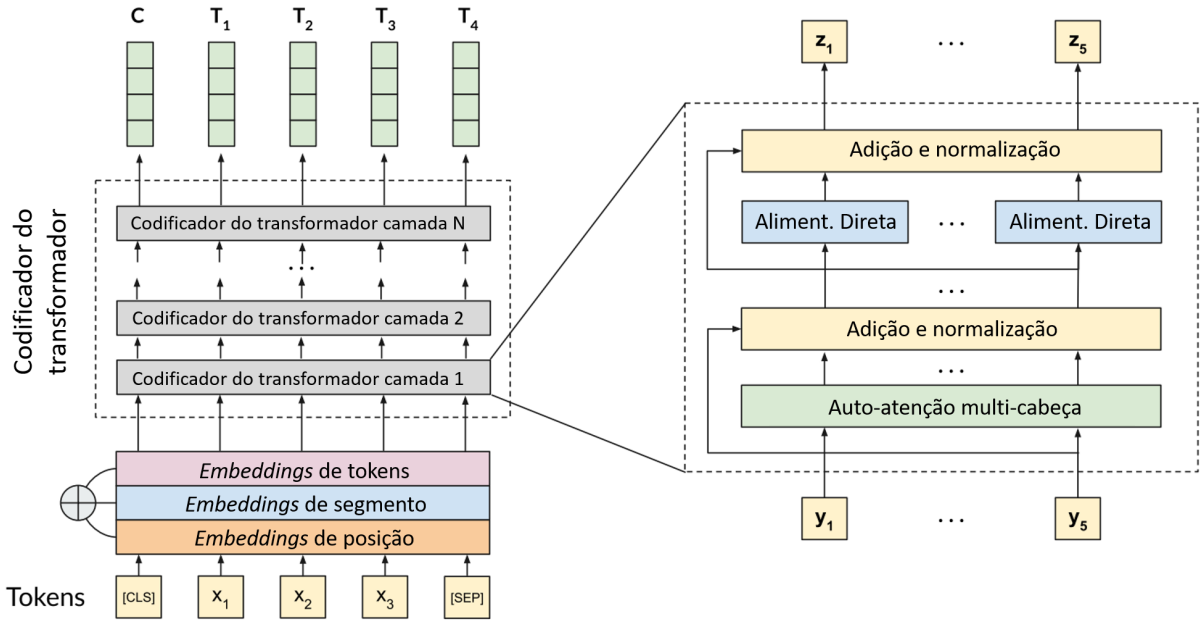


Figura 2.6: Arquitetura do BERT (Fonte: adaptado de [14]).

Para facilitar a sua compreensão, a Figura 2.6 esquematiza a estrutura do BERT considerando um exemplo de entrada de dimensão cinco. A utilização desse modelo demanda, primeiramente, adaptar os dados para uma representação específica, na qual os textos são tokenizados, isto é, divididos em unidades menores, conhecidas como tokens. Especificamente para o BERT, esse processo é granularizado a nível de pedaços de palavras (*wordpieces*), com base em um vocabulário pré-definido que pode incluir caracteres, sequências de caracteres de comprimento variável e até mesmo palavras inteiras. São ainda empregados três tokens especiais: o `[CLS]` e o `[SEP]`, que marcam, respectivamente, o início e o término de cada sequência de entrada, e o `[PAD]`, para fins de preenchimento dos *embeddings* de modo a respeitarem um tamanho pré-fixado (técnica conhecida como *padding*) [13, 7].

Além dos "*embeddings* de tokens" supracitados, o BERT recebe duas outras representações na suas entradas. A primeira, denominada "*embedding* de posição", fornece informações sobre a posição dos tokens, necessária diante da natureza não-sequencial da arquitetura de transformadores, podendo ser formulada por $i \in \{1, 2, ..., S\}$, em que S é o tamanho máximo das entradas. A segunda representação, por sua vez, traz os "*embeddings* de segmento", com o objetivo de desambiguar duas sentenças (A e B) concatenadas nas sequências de tokens, possibilitando o *fine-tuning* do modelo para tarefas específicas de NLP, como a Respostas a Perguntas, ou *Question Answering* (QA), na qual existe um interessante em dissociar o conteúdo das perguntas do das respostas. Como resultado, o *embedding* de entrada para um dado token x_i de um vocabulário $\mathcal{V}$ - de tamanho V - pode ser formalmente definido pela Equação 2.10, em que $LayerNorm$ é uma camada de normalização, H é a dimensão oculta do modelo, $\mathbf{E}_V \in \mathbb{R}^{VxH}$ é a matriz de *embeddings* dos tokens, $\mathbf{E}_{pos} \in \mathbb{R}^{SxH}$ é a matriz de *embeddings* de posição e $\mathbf{E}_{seg} \in \mathbb{R}^{2xH}$ é a matriz de *embeddings* de segmento [14, 12].

$$\mathbf{E}\left(x_i\right) = LayerNorm\left(\mathbf{E}_V^{x_i} + \mathbf{E}_{pos}^{i} + \mathbf{E}_{seg}^{A|B}\right) \in \mathbb{R}^H \tag{2.10}$$

Com as entradas estabelecidas, o BERT mapeia a sequência de tokens $\mathbf{x}$ para uma sequência de representações codificadas, conforme descrito na Equação 2.11, na qual n é a quantidade de tokens, $T_i \in \mathbb{R}^H$ é a representação codificada do i-ésimo token x_i na sequência $\mathbf{x}$ e $\mathbf{c} \in \mathbb{R}^H$ é uma representação agregada de toda a sequência. Para isso, cada entrada passa por um codificador de transformador (*transformer encoder*), que consiste em uma pilha de N camadas idênticas que alternam operações de auto-atenção e de alimentação direta (*feed-forward*) em sub-camadas específicas com conexões residuais seguidas de normalização de camada, conforme ilustrado na Figura 2.6. Nesse processo, entradas e saídas intermediárias, representadas, respectivamente, por $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n\}$ e $\mathbf{z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n\}$ são obtidas [14, 12].

$$(x_1, x_2, ..., x_n) \mapsto (\mathbf{c}, \mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_n) \tag{2.11}$$

Ao final, para tarefas a nível de palavras, a representação codificada T_i de cada token é retirada diretamente da sua posição correspondente no estado oculto final $\mathbf{z}$. Em contrapartida, para o token inicial de cada entrada (`[CLS]`), cujo propósito reside em produzir uma representação codificada agregada $\mathbf{c}$ de toda a sequência, é efetuada transformação em uma camada linear com tangente hiperbólica como função de ativação, conforme descrito na Equação 2.12, em que $\mathbf{W}_c$ contém os pesos e $\mathbf{b}_c$ os bias, e $\mathbf{z}_1$, $\mathbf{b}_c \in \mathbb{R}^H$ e $\mathbf{W}_c \in \mathbb{R}^{HxH}$. Assim, no estágio de pré-treinamento, o modelo é treinado do zero nas tarefas auto-supervisionadas (MLM e NSP) para aprender representações úteis de $\mathbf{c}$ e $\mathbf{T}_i$.

Este estágio é computacionalmente intensivo e pode ser executado apenas uma vez. Na etapa de ajuste-fino, outro modelo específico é então anexado ao BERT pré-treinado, em uma abordagem direta ou híbrida, e todo o sistema é treinado de ponta-a-ponta na tarefa de interesse [13, 12].

$$\mathbf{c} = tanh\left(\mathbf{z}_1\mathbf{W}_c + \mathbf{b}_c\right) \tag{2.12}$$

2.5 Métricas de desempenho

Conforme ilustrado na Figura 2.1, a etapa de modelagem - seja por métodos tradicionais ou de aprendizagem profunda - é sucedida pela avaliação dos classificadores. Essa etapa consiste, basicamente, em testar a efetividade dos modelos implementados. Em outras palavras, trata-se de validação da adequação dos tratamentos aplicados aos dados e da modelagem escolhida [35]. Para avaliar o resultado gerado por classificadores, faz-se necessária a aplicação de métricas para aferir sua capacidade de tomar decisões corretas de classificação.

Em tarefas envolvendo ML, essa avaliação é feita utilizando métricas de desempenho [31], geralmente aplicadas com os seguintes objetivos: (i) selecionar o melhor modelo dentre um conjunto de modelos gerados, sendo a métrica, nesse caso, calculada sobre os dados de validação; e (ii) estimar a capacidade de generalização do classificador, o que envolve a mensuração do desempenho do modelo sobre os dados de teste [60]. No contexto de classificação, as métricas de desempenho podem ser segregadas em três famílias, conforme proposto por [61], a saber: de limiar; de ranqueamento; e de probabilidade.

2.5.1 Métricas de limiar

Métricas de limiar são aquelas que quantificam os erros de previsão da classificação, retornando a proporção de situações em que a classe prevista não corresponde à classe esperada em um dado conjunto de dados [62]. Essas métricas são sensíveis a um valor limite (*threshold*, em inglês) que influencia na escolha da classe a qual uma instância pertence, impactando, assim, o desempenho do modelo. A maioria das métricas de limiar pode ser compreendida em termos de uma matriz de confusão, que, para problemas de classificação binária, como aquele abordado na presente pesquisa, assume a forma apresentada na Tabela 2.1 [63, 64].

Em classificações de dados desbalanceados, isto é, que apresentam categorias com frequências muito diferentes, a classe majoritária é usualmente referida como "negativo" e a minoritária como "positivo". Na tarefa abordada neste estudo, seriam correspondentes

Tabela 2.1: Matriz de confusão para classificação binária

		Real	
		Negativa	Positiva
Predita	Negativa	TN	FN
	Positiva	FP	TP

a demandas improcedentes (70%) e procedentes (30%), respectivamente, conforme discorrido no Capítulo 4, porém mantido neste referencial teórico para fins didáticos. Dessa forma, na matriz de confusão apresentada, temos que TN (verdadeiro negativo) se refere a demandas improcedentes classificadas corretamente; FN (falso negativo) a demandas procedentes classificadas incorretamente como improcedentes; FP (falso positivo) a demandas improcedentes classificadas incorretamente como procedentes; e TP (verdadeiro positivo) a demandas procedentes classificadas corretamente.

Dentre as diversas métricas de limiar existentes, foram ressaltadas, nesta dissertação, a **precisão** e a ***revocação*** (também conhecida como sensibilidade ou taxa de verdadeiros positivos - TPR), que focam em uma classe específica, sendo, portanto, amplamente utilizados em problemas de classificação desbalanceada. A precisão, calculada pela Equação 2.13, quantifica o número de classificações positivas corretas dentre todas as predições positivas efetuadas pelo classificador, podendo ser interpretada como a capacidade do modelo de não categorizar como positiva uma instância negativa. Em contrapartida, a *revocação*, calculado pela Equação 2.14, mensura o número de classificações positivas efetuadas corretamente dentre todas as instâncias de fato positivas; ou seja, avalia quão bom é o modelo em categorizar corretamente as instâncias positivas [60].

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{2.13}$$

$$\text{Revocação} = \text{TPR} = \frac{TP}{TP + FN} \tag{2.14}$$

2.5.2 Métricas de ranqueamento

As métricas de ranqueamento, por sua vez, são mais voltadas para a avaliação dos classificadores com base na sua efetividade em separar as classes, sendo empregadas, usualmente, para selecionar as melhores n instâncias de um conjunto de dados. Essas métricas requerem que o modelo retorne um *score*, ou uma probabilidade das instâncias de pertencerem a uma classe. A partir dessa predição, diferentes limiares podem ser aplicados para testar o desempenho dos classificadores [61, 62].

A métrica de ranqueamento mais comumente utilizada para tarefas de classificação binária é a curva Característica de Operação do Receptor (**ROC**), que descreve o com-

portamento de um modelo calculando a taxa de falsos positivos (FPR) - conforme Equação 2.15 - e a taxa de verdadeiros positivos - descrita na Equação 2.14 - para um conjunto de previsões do classificador sob diferentes limiares. Uma característica importante dessa curva é que, diferentemente das métricas de limiar, ela é insensível a mudanças na distribuição de classes. Logo, se no presente estudo, a título exemplificativo, a proporção entre demandas procedentes e improcedentes variar nos dados de teste, a curva ROC não será afetada [60, 63].

$$\text{FPR} = \frac{FP}{FP + TN} \tag{2.15}$$

A Figura 2.7 (a) ilustra um exemplo dessa curva, com o desempenho de três classificadores: A, B e C. A linha pontilhada, obtida com às predições do modelo C, representa classificações aleatórias das instâncias, características de um modelo sem habilidades (*no skill*). Quaisquer pontos abaixo desta linha indicam piores desempenhos, sendo que, a representação para um classificador perfeito é dada por um ponto no canto superior esquerdo do gráfico. Consequentemente, o classificador A pode ser considerado superior aos modelos B e C [65].

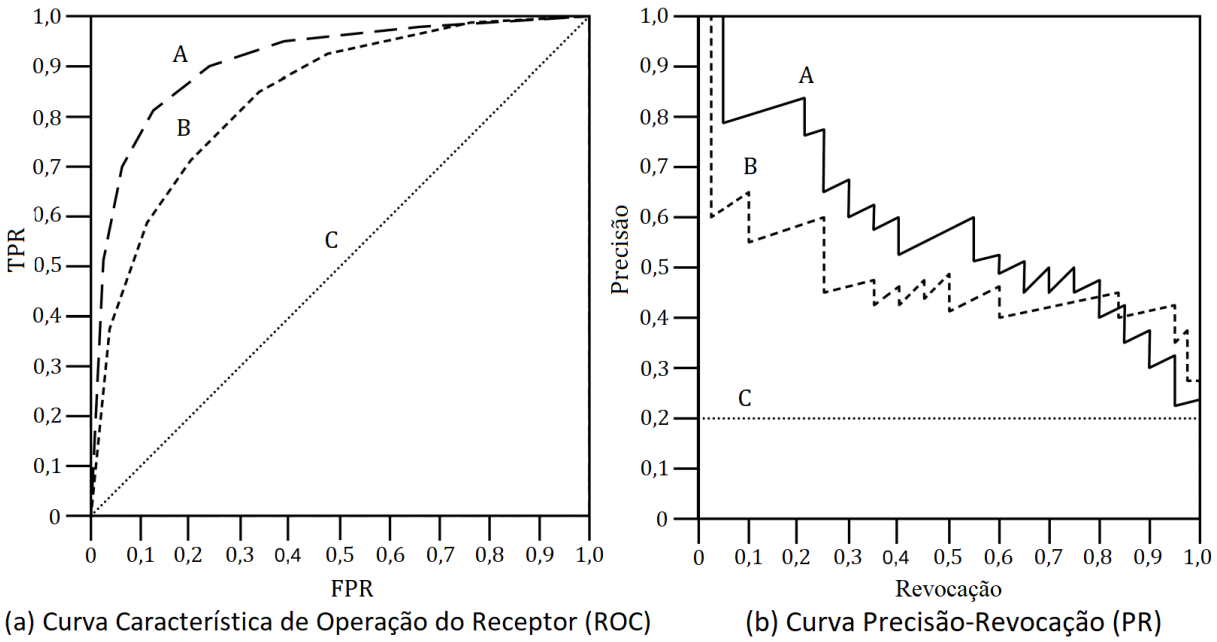


Figura 2.7: Curvas ROC e PR (Fonte: adaptado de [60]).

Outra métrica de ranqueamento adotada como alternativa à curva ROC é a curva de Precisão-Revocação (**PR**), que pode ser utilizada de forma semelhante, porém com foco no desempenho do classificador para a classe minoritária. Para problemas de classificação envolvendo dados severamente desbalanceados, com poucos exemplos da classe positiva, a

curva ROC pode fornecer uma visão excessivamente otimista do desempenho dos modelos, uma vez que pequenos números de previsões corretas ou incorretas podem resultar em grandes mudanças na curva. Em contraste, a curva PR é especificamente adaptada para a detecção de eventos raros, sendo mais recomendada quando o interesse reside na classe minoritária [64, 62].

Essa métrica também considera diferentes limiares para avaliar o desempenho dos classificadores, porém calculando a precisão (Equação 2.13) e a revocação (Equação 2.14). Nesse caso, predições de um modelo sem habilidades correspondem a uma linha horizontal com precisão proporcional ao número de instâncias positivas no *dataset*, e os melhores desempenhos são representados por curvas próximas ao canto superior direito do gráfico. Na Figura 2.7 (b), é apresentado um exemplo da curva Precisão-Revocação (PR), no qual, mais uma vez, o classificador A é superior ao B e ao C, com o desempenho deste último sendo representado pela linha de base (*no skill*) [65].

Para fins de comparação entre o desempenho dos modelos classificadores, é interessante resumir as informações apresentadas pelas curvas ROC e PR em valores escalares. Para isso, calcula-se a área sob essas curvas (AUC), obtendo-se métricas - **ROCAUC** e **PRAUC**, respectivamente - que sumarizam as habilidades dos modelos. Em ambos os casos, a área calculada será sempre um valor no intervalo [0, 1], sendo que, quanto mais próximo de um, melhor o desempenho do classificador. Notadamente para a curva PR, o cálculo da AUC pode ser bastante desafiador, sendo recomendadas, na literatura, diferentes abordagens para aproximar o seu valor, com destaque para a Precisão Média, que considera a média ponderada das precisões em cada limiar, com o aumento do *recall* do limiar anterior usado como fator de ponderação,sendo calculada de acordo com a Equação 2.16, na qual P_n e R_n são os valores da precisão e do *recall* no n-ésimo *threshold* [66, 60].

$$\text{Precisão Média} = \sum_n \left(R_n - R_{n-1}\right) P_n \tag{2.16}$$

2.5.3 Métricas de probabilidade

As métricas de probabilidade são projetadas especificamente para quantificar a incerteza nas previsões de um classificador, sendo normalmente adotadas quando o objetivo é avaliar a confiabilidade dos modelos, não apenas mensurando suas falhas, mas também aferindo se a seleção de classes incorretas foi feita com alta ou baixa probabilidade [61].

Todavia, avaliar um modelo baseado nas probabilidades previstas requer que estas estejam calibradas e, geralmente, algoritmos de ML, sobretudo os não lineares, costumam ser treinados em estruturas não probabilísticas, retornando predições descalibradas das classes. Assim, para a utilização das métricas em epígrafe, é necessário, primeiro,

assegurar que as saídas do classificador estejam devidamente calibradas em relação ao *dataset* [62].

Argumentavelmente, a métrica mais comum para avaliar as probabilidades previstas em problemas de classificação binária é a **entropia cruzada**, também chamada de *LogLoss*, calculada de acordo com a Equação 2.17, em que y corresponde aos valores esperados e p aos valores previstos [61].

$$\text{LogLoss} = ylog(p) + (1 - y)log(1 - p) \tag{2.17}$$

Capítulo 3

Trabalhos Relacionados

Neste capítulo, foram sumarizados trabalhos correlatos ao tema de classificação de textos, abrangendo desde revisões da literatura quanto ao estado-da-arte até experimentos envolvendo aplicações práticas no mundo real. Foram contemplados estudos nacionais e internacionais, envolvendo o processamento de textos longos, a construção de modelos multimodais e a categorização de reclamações em diferentes contextos.

3.1 Revisão do estado-da-arte

Diante da importância da tarefa de classificação de texto no âmbito do processamento de linguagem natural, considerando um cenário de explosão informacional junto de um acentuado desenvolvimento da IA, inúmeras pesquisas foram conduzidas buscando alavancar modelos de categorização automática de documentos, baseados, principalmente, em métodos de aprendizagem profunda. Em [5], foram revisadas técnicas de estado-da-arte desde 1961 a 2021, abarcando tanto modelos tradicionais quanto de DL. Adicionalmente, foram propostas diferentes taxonomias para a tarefa em questão com base nos textos envolvidos e nos modelos empregados para a extração e classificação dos atributos, elencando suas vantagens e desvantagens.

Na aludida pesquisa, os autores ressaltam que, em contraste com métodos tradicionais, que demandam uma custosa engenharia de variáveis antes de treinar o classificador, abordagens baseadas em DL, nas quais os modelos aprendem a extrair os atributos mais relevantes para a tarefa de interesse, apresentam recorrentemente melhores resultados, embora demandem maior carga de processamento computacional. Ainda de acordo com o estudo, os melhores classificadores, de modo geral, são aqueles que adotam arquitetura de transformadores, sendo capazes de treinar, por computação paralelizada, grandes LMs que conseguem capturar aspectos sintáticos e semânticos das palavras de acordo com o seu contexto, proporcionando representações enriquecidas do texto. Contudo, métodos de DL

resultam em modelos caixa-preta, em detrimento da sua interpretabilidade, configurando, nesse aspecto, nítida desvantagem frente aos classificadores tradicionais.

Em extensão, em [32] os autores, assumindo a superioridade no desempenho apresentada por métodos de aprendizagem profunda em diversas tarefas de NLP, realizaram uma revisão abrangente de mais de 150 modelos de DL desenvolvidos recentemente para classificação de textos, efetuando análise quantitativa da sua performance em *benchmarks* populares e dicutindo as tendências atuais de pesquisas na área. Dentre os resultados apresentados, são ressaltadas estratégias populares envolvendo o pré-treinamento de LMs - notadamente o BERT e seus derivados - em grandes corporas previamente ao ajuste-fino em tarefas à jusante. É, todavia, feita ponderação quanto à capacidade de modelos pré-treinados em documentos de domínio amplo para generalizarem em problemas de domínio específico. Ademais, os autores recomendam que a arquitetura das camadas voltadas para a tarefa de interesse seja escolhida de acordo com a sua natureza, levando em consideração, por exemplo, a estrutura linguística do texto a ser capturada. Enfim, assinalam desvantagens desses modelos relacionadas ao grande consumo de memória para o treinamento e, assim como em [5], à sua baixa interpretabilidade.

Na literatura nacional, dada a escassez de dados rotulados em português, os avanços com técnicas de NLP não têm apresentado o mesmo crescimento que na língua inglesa, que conta com vastos e diversificados *datasets* adotados internacionalmente como *benchmarks* [52, 67]. Não obstante, em 2020 foi publicado trabalho de grande relevância que permitiu alavancar diversas tarefas de NLP no Brasil. A pesquisa objetivou o estabelecimento de um robusto LM para a língua portuguesa, conhecido como BERTimbau [13].

Os autores, nesse trabalho, pré-treinaram um BERT em um enorme corpora em português, conhecido como brWaC, contendo mais de 2,68 bilhões de tokens gerados a partir de 3,53 milhões de documentos retirados de *webpages* brasileiras, com alta diversidade de domínios e qualidade de conteúdo, sendo a maior coleção aberta nessa língua até o momento [14]. Um novo vocabulário de cerca de 30 mil unidades de sub-palavras foi gerado para o modelo com base neste *dataset* e frases aleatórias de artigos da Wikipédia em português, sendo posteriormente convertido para o formato de pedaços de palavras, representação de entrada para o BERT.

Para fins de validação, o BERTimbau foi avaliado em três tarefas de NLP à jusante (similaridade semântica, inferência textual e reconhecimento de entidades nomeadas) para os *datasets* ASSIN2 e First HAREM/MiniHAREM em português. Como resultado, o modelo melhorou seu estado-da-arte, superando o BERT multilíngue e abordagens monolinguais anteriores. Enfim, os autores disponibilizaram o BERTimbau para a comunidade (em dois tamanhos, Base e Large, assim como BERT original), de modo a promover bases de comparação para pesquisas futuras na área.

3.2 Aplicações práticas

Em aplicações de classificação de texto para solução de problemas do mundo real, nos quais a disponibilidade de recursos computacionais e de tempo para o treinamento dos modelos acaba se tornando um severo fator limitante, a utilização de métodos tradicionais ainda tem se mostrado bastante popular. Na área de Gestão de Relacionamento com o Cliente (CRM), foi feita, em [45], uma revisão sistemática dos principais algoritmos de ML aplicados em ferramentas correlatas. A pesquisa buscou responder três questões (Quais áreas do CRM atualmente tem mais aplicações com ML? Quais algoritmos de ML são aplicados em processos de CRM? Como os algoritmos de ML melhoram os processos de CRM?), com referência em quatro grande bases de dados internacionais (Ebsco Database, IEEE Xplore, Science Direct, Emerald), sendo selecionados, ao final, 84 estudos mediante aplicação dos critérios de inclusão/exclusão. Especificamente para tarefas de monitoramento de reclamações, que aplicam técnicas de mineração de texto, verificou-se forte predominância de modelos clássicos (90%), principalmente classificadores baseados em *ensemble* (*XGBoost*, *Adaboost* e *Random Forest*).

Por outro lado, ainda que mais escassas, observam-se pesquisas empregando DNNs para a categorização de reclamações, com relatos de desempenho satisfatório obtido com os modelos propostos. A título de exemplo, similar ao objeto de estudo tratado nesta dissertação, em [10] os autores propuseram um categorizador automático para as reclamações de clientes de uma concessionária de energia elétrica. Para isso foi considerada uma abordagem em árvore, na qual cada documento passava por dois processos de classificação distintos, em diferentes níveis de granularidade. Dessa forma, foi desenvolvido um modelo hierárquico incorporando, no primeiro nível, um LM pré-treinado (BERT) para obter os rótulos "rasos", de menor granularidade, seguido de um modelo de Word2Vec para a predição dos rótulos "profundos", que configuram a categorização de interesse. Os autores conduziram experimentos para validar o modelo assinalando sua viabilidade e efetividade. Contudo, pontuam limitações relacionadas principalmente à baixa disponibilidade de dados rotulados, ainda que amenizada pelo uso de técnicas de transferência de aprendizado.

Em âmbito nacional, um trabalho que impulsionou fortemente tarefas de NLP em aplicações práticas foi o projeto VICTOR, apresentado em [7]. O trabalho consistiu na construção de nova base de dados a partir de documentos jurídicos digitalizados do Supremo Tribunal Federal (STF), reunindo mais de 45 mil recursos extraordinários e totalizando cerca de 692 mil documentos, ou 4,6 milhões de páginas. Os dados produzidos nesse projeto foram rotulados, contendo anotações para duas tarefas diferentes: classificação de tipo de documento e identificação de tema de repercussão geral. A primeira tratou da classificação por página, em que cada uma podia pertencer a seis classes disjuntas. A

segunda abordou a classificação por processo, sendo uma tarefa *multi-class* na qual cada processo podia ter mais de um tema de repercussão geral.

Adicionalmente, os autores, nesse projeto, treinaram uma série de modelos no intuito de se estabelecer uma linha-de-base para trabalhos futuros, comparando representações textuais e sequenciais dos dados. Esses modelos contemplaram tanto métodos tradicionais - com representações de saco-de-palavras e três diferentes classificadores, quais sejam NB, SVM e XGBoost - quanto métodos de DL, abrangendo modelos baseados em BiLSTM e CNN. Foi ainda avaliada a possibilidade de aproveitar a natureza sequencial dos dados para melhorar os resultados de classificação de tipo de documento sendo, para isso, treinado um campo aleatório condicional (CRF) de cadeias lineares nas predições de uma CNN treinada nos dados, o que proporcionou visíveis melhorias. A base de dados foi disponibilizada em três versões de diferentes tamanhos e conteúdos (BigVictor, MediumVictor e SmallVictor) para incentivar a exploração de melhores modelos e técnicas.

Ainda no tocante a trabalhos nacionais, um estudo similar ao proposto nesta dissertação foi realizado recentemente, analisando a possibilidade de automatizar a classificação de documentos recebidos pela Administração Pública brasileira a partir de um estudo de caso envolvendo reclamações de passageiros de companhias aéreas encaminhadas à Agência Nacional de Aviação Civil (Anac). A pequisa realizada em [4] teve como justificativa a otimização de processos de trabalho da aludida agência por meio da implementação de um classificador *multi-class* automático das reclamações, no intuito de liberar recursos para outras tarefas, em prol de um melhor atendimento ao cidadão.

Para isso, foi utilizada a metodologia ULMFiT, proposta em [8], que consiste em primeiro pré-treinar um LM baseado na arquitetura da LSTM em um corpus genérico; em seguida efetuar o ajuste-fino desse modelo em um domínio específico de negócio, usualmente sendo o *dataset* selecionado para a tarefa à jusante, empregando técnicas de ajuste-fino discriminativo e taxas de aprendizado triangulado assimétricas; e, por fim, ajustar o modelo para a tarefa de interesse. No estudo em [4], a primeira etapa foi feita a partir de um conjunto de milhares de artigos da Wikipédia portuguesa; e, para as demais etapas, foi considerado um *dataset* contendo aproximadamente 40 mil reclamações de passageiros classificadas manualmente pela Anac.

O modelo construído em [8] foi validado aplicando-se a técnica de validação cruzada, na qual o *dataset* é particionado em subconjuntos mutuamente exclusivos, e avaliações iterativas são efetuadas empregando um subconjunto para validação e os demais para treinamento. Embora o desempenho final do classificador não tenha alcançado a meta proposta, que acabou se mostrando altamente rigorosa, os resultados obtidos foram promissores, superando outros modelos previamente implementados para a tarefa em questão, servindo ainda como referência para trabalhos futuros na área de NLP.

3.3 Classificação de textos longos

Embora a transferência de aprendizado a partir de robustos LMs pré-treinados já tenha se estendido para aplicações práticas de classificação de texto, como esses modelos usualmente são baseados na arquitetura de transformadores, apresentam nítidas limitações para o processamento de grandes entradas (a exemplo de transcrições de relatos ou reclamações mais descritivas), em decorrência da complexidade dos seus mecanismos de atenção. Diante desse entrave, vários pesquisadores passaram a adotar modelos híbridos que permitissem processar os dados de forma segmentada.

Assim, em [15], os autores propõem duas arquiteturas distintas (RoBERT e ToBERT) que permitem a aplicação do BERT em classificação de textos longos. Em ambas as propostas, as sequências de entrada são segmentadas em *chunks* de tamanho fixo com janelas de sobreposição, antes de serem passadas para o BERT, visando a geração dos *embeddings* contextualizados. Essas saídas são concatenadas em uma sequência para então serem processadas por uma LSTM, dando origem ao RoBERT, ou por outro transformador, resultando no ToBERT, propiciando a classificação final do documento. Os autores avaliaram os modelos em duas tarefas de classificação de texto - predição de satisfação do cliente e identificação de tópicos - usando três conjunto de dados tidos como *benchmarks*, e obtiveram melhorias significativas em relação aos classificadores de linha-de-base.

Outro estudo contemplando abordagens hierárquicas de transferência de aprendizagem para classificação de documentos longos foi conduzido recentemente por [56]. Nele, os autores propõem a utilização do BERT e do *Universal Sentence Encoder* (USE) para eficientemente capturar melhores representações textuais. Assim como em [15], as sequências de entradas são segmentadas e passadas individualmente para os LMs, sendo posteriormente propagadas para uma rede neural rasa - no caso uma LSTM ou uma CNN - de modo a gerar *embeddings* a nível de documento, conforme ilustra a Figura 3.1. Os autores avaliaram os quatros arranjos resultantes da associação dos modelos considerando cada nível hierárquico (BERT+LSTM, BERT+CNN, USE+LSTM e USE+CNN) em seis *benchmarks* diferentes, obtendo resultados equiparáveis ou mesmo superiores aos dos modelos de referência.

Os experimentos do estudo confirmam que as quatro configurações descritas podem efetivamente ser empregadas para tarefas de classificação de textos longos, com destaque para o arranjo BERT+LSTM, que apresentou maior desempenho que os demais. Por fim, comentam que a utilização de técnicas mais sofisticadas, como o Longformer, podem proporcionar resultados melhores. No entanto, ressaltam que o seu mecanismo de atenção prejudica a paralelização obtida com a arquitetura de transformadores, além de requererem manuseio especializado. Ademais, conforme frisado em [54], tanto o Longformer quanto o Reformer, ambos modelos que adaptam o BERT para processar grandes en-

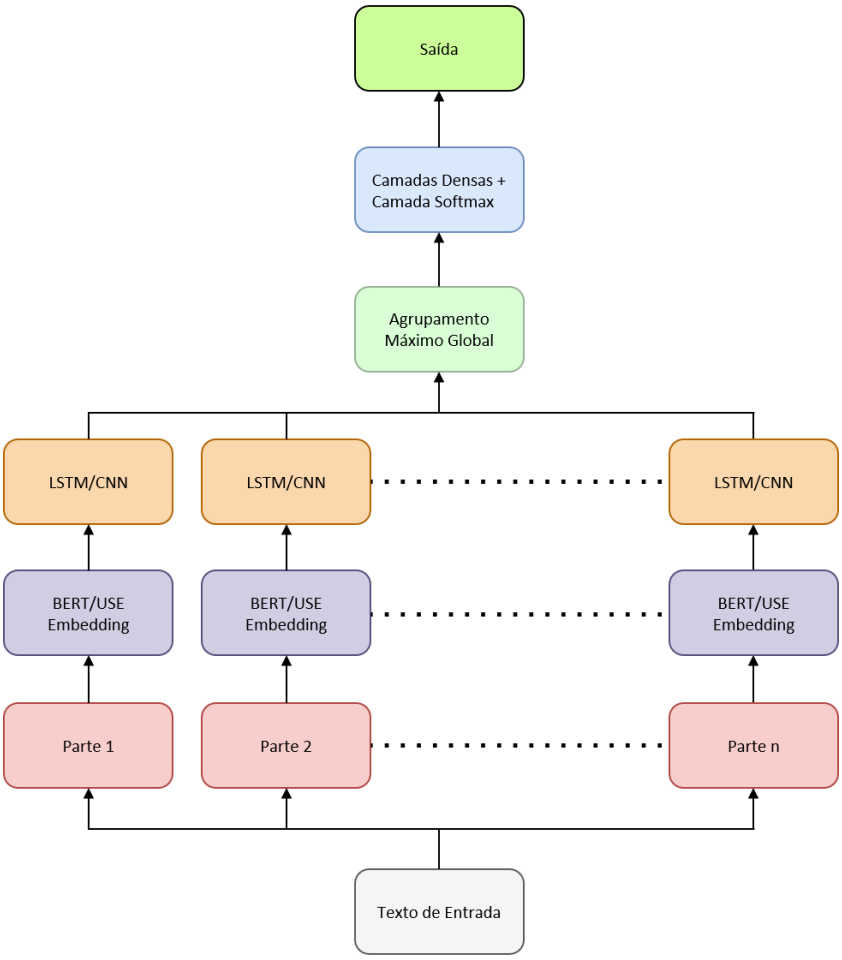


Figura 3.1: Arquitetura proposta para modelo híbrido (Fonte: [56]).

tradas, ainda carecem de publicações disponibilizando pesos pré-treinados para domínios específicos ou outras línguas.

3.4 Classificação multimodal

Uma lacuna que têm sido observada na literatura remonta à utilização de variáveis categóricas e numéricas junto de atributos de texto em DNNs construídas com a arquitetura de transformadores [1, 3]. Como a categorização de documentos pode requerer conhecimentos que nem sempre estão contidos no texto, a incorporação de dados tabulares pode vir a agregar valor para o desempenho desses modelos. Outrossim, dois trabalhos foram publicados no último ano contendo estratégias multimodais para tratar da questão. Em [1] é proposto o Multimodal-Toolkit, um pacote *open-source* de implementação em python para integrar dados textuais e tabulares com transformadores em aplicações à justante.

A ferramenta conta, primeiro, com uma etapa de pré-processamento dos dados que adequa as variáveis para a entrada em um módulo denominado pelos autores de "Transformador com Tabular", conforme ilustrado na Figura 3.2. Esse módulo consiste, basicamente, em um bloco de transformador - com a implementação da Hugging Face[1], para facilitar a interface com o usuário - e um sub-módulo de combinação para associar as variáveis multimodais antes de ingressarem em uma camada totalmente conectada que retorna as predições do modelo. Logo, os atributos de texto são primeiro processados pelo transformador antes de serem combinados com os dados tabulares, opção adotada para manter a praticidade na utilização de pacotes já implementados.

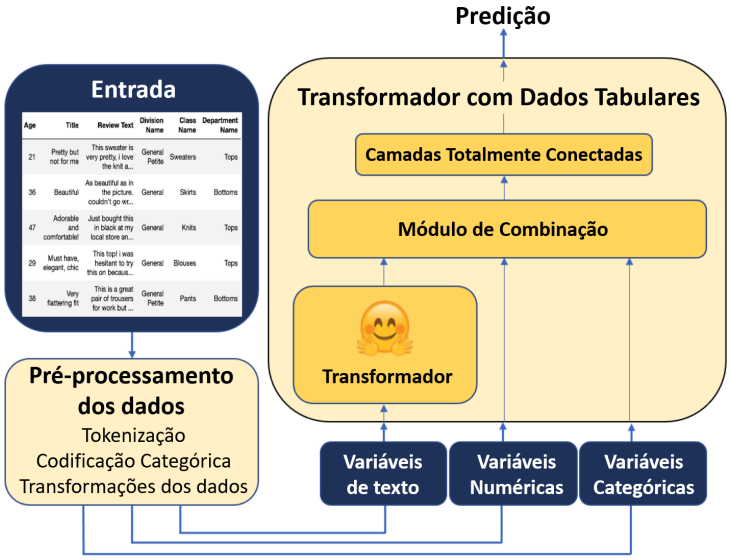


Figura 3.2: Estrutura do Multimodal-Toolkit (Fonte: [1]).

No módulo combinador, foram propostas múltiplas estratégias de combinação das variáveis em uma representação unificada. Considerando $\mathbf{x}$ a representação textual obtida do transformador e $\mathbf{n}$ e $\mathbf{c}$ os atributos numéricos e categóricos pré-processados, respectivamente, o módulo retorna uma representação multimodal combinada $\mathbf{m}$. A primeira e mais simples estratégia - chamada *Unimodal* pelos autores - envolve a conversão das variáveis tabulares em texto, de modo a passarem também pelo bloco do transformador, obtendo-se, ao final, $\mathbf{m} = \mathbf{x}$. Outra abordagem de menor complexidade diz respeito à concatenação direta dos *embeddings* de representação dos atributos, formulada por $\mathbf{m} = \mathbf{x}||\mathbf{n}||\mathbf{c}$, em que $||$ é o operador de concatenação. É possível ainda estabelecer perceptron multicamadas (MLP) para obter *embeddings* específicos das variáveis tabulares, tal que $\mathbf{m} = \mathbf{x}||\text{MLP}(\mathbf{n})||\text{MLP}(\mathbf{c})$, ou da concatenação delas, resultando em $\mathbf{m} = \mathbf{x}||\text{MLP}(\mathbf{n}||\mathbf{c})$. Estratégias mais complexas podem ainda ser adotadas, como a uti-

[1]https://huggingface.co/docs/transformers/index

lização de mecanismos de atenção e de *gating* nos atributos categóricos e numéricos, ou mesmo a soma ponderada de todas as variáveis.

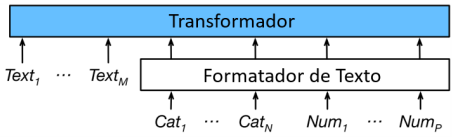
Para validar a ferramenta Multimodal-Toolkit, os autores, em [1], realizaram experimentos testando cada um dos métodos de combinação descritos para tarefas de regressão, classificação binária e classificação *multi-class*, com *datasets* de referência e métricas específicas para cada tarefa. Foi adotado, como linha-de-base, o desempenho de modelo construído apenas com as variáveis textuais. Os resultados obtidos apontam que todas as estratégias resultaram em ganho quando comparadas ao desempenho de referência, indicando vantagens na adição de dados tabulares ao modelo quando disponíveis. Por outro lado, o ganho obtido com a abordagem multimodal variou de acordo com o *dataset*, sugerindo que a melhor configuração depende do problema a ser resolvido.

Ainda sobre o tema, em [2] foram projetados sistemas automatizados de aprendizado supervisionado para tabelas de dados que contenham não apenas colunas numéricas/categóricas, mas também campos de texto. Um ponto forte desse estudo foi a utilização de *benchmarks* criados e publicados pelos autores em [3], constituídos de 15 tabelas de dados multimodais - cada uma contendo alguns campos de texto - derivadas de aplicações de negócios reais. A pesquisa apresenta diversas estratégias baseadas em adaptações multimodais de blocos de transformadores acoplados a modelos tabulares clássicos.

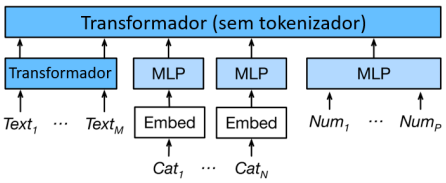
Os autores primeiro estabelecem um panorama geral dos métodos de categorização de dados textuais e tabulares. Em seguida, propõem quatro estratégias para adaptar transformadores para operar simultaneamente em entradas de ambas as modalidades, apresentadas na Figura 3.3. Por fim, estabelecem métodos de agregação dessas arquiteturas com modelos tabulares, sob o enfoque de ML automatizada (AutoML), não sendo abordados na presente dissertação, uma vez que fogem ao tema de interesse deste estudo.

Dentre as estratégias de agregação multimodal propostas em [2], a primeira, chamada pelos autores de *All-Text* - Figura 3.3 (a) -, envolve converter valores numéricos e categóricos em cadeia de caracteres e subsequentemente tratar suas colunas também como campos de texto, sendo, portanto, equivalente à estratégia Unimodal descrita em [1]. As outras duas abordagens são também similares as concatenações com MLPs propostas anteriormente. Porém, na *Fuse-Early* - Figura 3.3 (b) - um segundo transformador opera nos *embeddings* aprendidos para cada atributo, com arquiteturas de MLPs representando os dados tabulares. É ainda introduzida uma camada extra de *embedding* fatorado para mapear as variáveis categóricas nas mesmas representações textuais aprendidas pelo LM pré-treinado. As variáveis númericas, por sua vez, são codificadas conjuntamente no MLP de modo a se obter uma representação unificada.

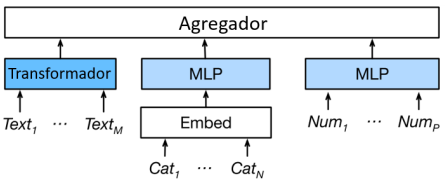
Na estratégia *Fuse-Late* - Figura 3.3 (c) -, em contrapartida, operações neurais distintas são efetuadas em cada tipo de dado, com a agregação ocorrendo próxima à camada de



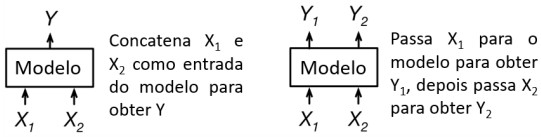
(a) *All-Text.* Converte valores numéricos e categóricos em *tokens* de texto adicionais.



(b) *Fuse-Early.* Transformador opera em *embeddings* aprendidos para cada variável.



(c) *Fuse-Late.* Ramos separados codificam cada modalidade, agregadas por concatenação.



(d) Notação usada nessas Figuras.

Figura 3.3: Estratégias de agregação multimodal (Fonte: [2]).

saída, o que permite a cada "ramo" extrair representações mais enriquecidas das modalidades antes de a rede neural decidir como estas devem ser agregadas, processo que ocorre via concatenação ou aglomeração (*pooling*) com média ou máximo. Os experimentos efetuados com o novo *benchmark* consideraram as mesmas tarefas de NLP utilizadas em [1], apontando desempenho melhores utilizando a estratégia de *Fuse-Late*.

3.5 Conclusão da revisão da literatura

Embora a literatura aborde fortemente a tarefa de classificação de texto, ainda podem ser observadas lacunas relacionadas a aplicações práticas com dados textuais longos na língua portuguesa, principalmente considerando as novas estratégias multimodais recentemente sugeridas para a combinação dos atributos textuais processados por transformadores com outras modalidades de variáveis, como as categóricas e numéricas, muitas vezes disponíveis em problemas do mundo real. Nesse cenário, a presente dissertação apresenta, a título de contribuição científica, os resultados obtidos com experimentos multimodais envolvendo o processamento de grandes entradas de texto pelo BERT no âmbito de aplicação prática de categoriação de texto no processo de tratamento das reclamações dos cidadãos e usuários do Sistema Financeiro Nacional (SFN).

Capítulo 4

Análise dos Dados

Este capítulo apresenta, primeiramente, o entendimento de negócio envolvido no processo de tratamento das reclamações abertas pelos clientes e usuários do SFN. Em seguida são descritos os procedimentos adotados para a coleta e a seleção dos dados do Sistema de Registro de Demandas do Cidadão (RDR) utilizados nos experimentos. Por fim, é feita análise dos dados selecionados, abarcando os rótulos de procedência das demandas, os atributos de texto - reclamação do cidadão e resposta da Instituição Financeira (IF) - e as variáveis tabulares selecionadas pelo Banco Central do Brasil (BCB).

4.1 Entendimento do negócio

O cidadão brasileiro pode fazer reclamações contra entidades supervisionadas pelo BCB, tais como bancos, instituições de pagamento e administradoras de consórcios. Para fins de simplificação, essas entidades foram denominadas apenas como IFs na presente dissertação. Embora o Banco Central não possa interferir na relação contratual com a IF, ao direcionar uma reclamação ao BCB, o cidadão ajuda a melhorar as normas e a fiscalização do SFN[1]. Essa comunicação pode ocorrer por diversos canais: "Fale conosco" (internet), telefone, presencial (suspenso pela pandemia do COVID-19), correspondência (postal), "Protocolo Digital" e plataforma "Fala.br". Todas as reclamações são posteriormente centralizadas em um único sistema, o RDR [18, 20].

Embora a ferramenta trate também de outros tipos de demandas, como pedidos de informação corriqueiros ou provenientes da Lei de Acesso à Informação (LAI), que podem ser realizados pelos mesmos canais retromencionados, esta pesquisa teve como foco apenas as reclamações (reguladas ou não), que representam cerca de 60% dos atendimentos registrados, como ilustrado na Figura 4.1 [21, 22].

[1]https://www.gov.br/pt-br/servicos/registrar-reclamacao-contra-instituicao-supervisionada-pelo-banco-central

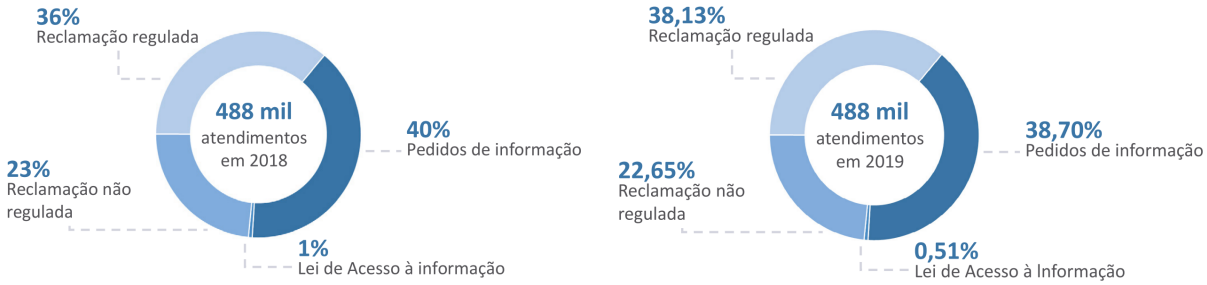


Figura 4.1: Atendimentos registrados em 2018 e 2019 (Fonte: adaptado de [21][22]).

Ao efetuar uma reclamação, o cidadão deve se identificar, descrever a sua demanda (quando via telefone, um servidor do BCB transcreve o texto do seu relato), indicar a entidade reclamada e selecionar a categoria da reclamação, podendo ainda anexar até 3 arquivos e informar quanto à existência de protocolo de atendimento aberto diretamente na IF. A categoria da reclamação permite que o RDR faça uma pré-classificação em demanda "regulada" ou "não-regulada", o que, contudo, não impede eventuais reclassificações por parte do Banco Central. Demandas "não-reguladas" não necessariamente indicam o descumprimento de obrigações regulatórias por parte da entidades, apenas que não é competência do BCB analisar a questão abordada. Uma demanda pode ainda ser cancelada, por exemplo, quando se trata de cópia de outra já existente, ou quando o reclamante não é parte legítima da reclamação [20].

Uma vez recebida a demanda, o BCB a repassa à entidade reclamada e acompanha o seu atendimento. Essa comunicação é sempre feita por meio do RDR e embora a grande maioria das reclamações siga direto para a IF, um percentual mínimo é retido pelo sistema em função de um filtro que analisa palavrões e certas palavras-chaves. A instituição financeira tem 10 dias úteis para responder, podendo pedir, de forma justificada, a prorrogação do prazo. A resposta é então encaminhada ao cidadão, com cópia para o Banco Central, que deve encerrá-la, classificando-a como "procedente" ou "improcedente". O fluxo de reclamações se encontra esquematizado na Figura 4.2, retirada do portal do BCB[2].

Para tomar uma decisão quanto à procedência da demanda, o servidor do Banco Central usualmente considera os textos do relato do cidadão e da resposta da IF, bem como os respectivos anexos. Outras informações que podem ser também consideradas envolvem o histórico de reclamações do cidadão, buscando, por exemplo, outras demandas similares, que podem indicar instatisfação quanto a respostas anteriores. Como material auxiliar, comumente é consultado sistema específico do Banco Central que concentra as "súmulas" e "jurisprudências" dos atendimentos realizados. No entanto, o servidor pode

[2] `www.bcb.gov.br/acessoinformacao/registrar_reclamacao`

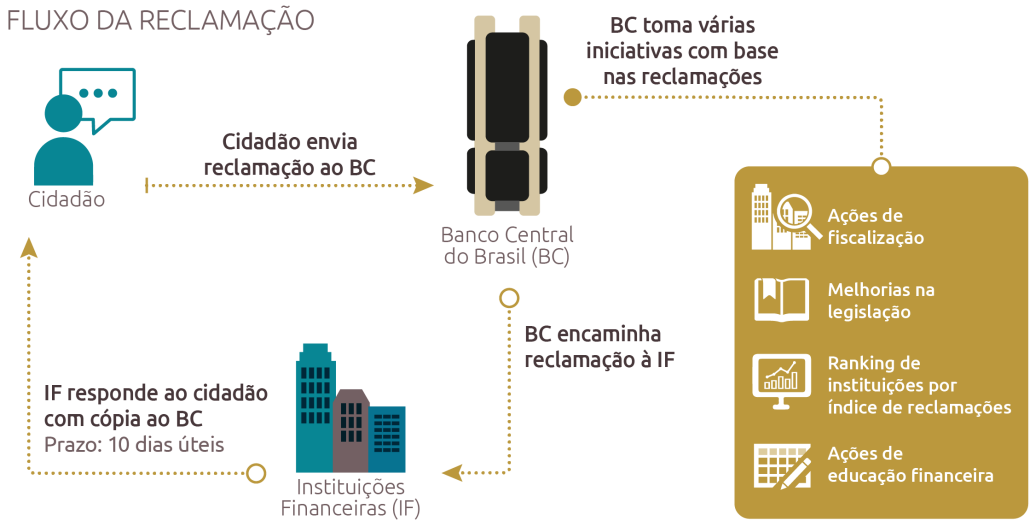


Figura 4.2: Fluxo das reclamações (Fonte: portal do BCB).

recorrer a diversas outras fontes de informação, desde pesquisas na internet até consultas à própria entidade reclamada. Eventualmente, em situações em que inexistam elementos que permitam a análise conclusiva da reclamação quanto a eventual descumprimento de normas pela IF, a demanda pode ser enquadrada como "não-conclusiva" [25].

No âmbito do BCB, essas atividades de tratamento das reclamações são realizadas manualmente por servidores do Departamento de Atendimento Institucional (Deati). Todavia, com o recente aumento do volume de demandas, atualmente só é possível analisar cerca de 60% da sua totalidade. Nesse contexto, foi desenvolvido, em janeiro de 2022, como resultado do projeto IPDR, modelo de ML para filtrar as reclamações com maior possibilidade de serem procedentes, buscando direcionar os esforços envidados pelo servidores do Deati para os casos em que de fato houve inobservância de obrigações regulatórias pelas entidades supervisionadas. Registra-se, contudo, que o modelo ainda não foi implementado em produção, inexistindo, portanto, registros da sua efetividade nos processos de trabalho do BCB [23, 25].

4.2 Configuração do ambiente de experimentos

Os experimentos desta pesquisa foram conduzidos utilizando os dados do RDR, armazenados em base corporativa do BCB. Porém, como os dados em questão podem apresentar aspectos pessoais sensíveis, nos termos do inciso II do artigo 5º da Lei Geral de Proteção

de Dados Pessoais (LGPD)[3], ou mesmo de sigilo bancário, estes tiveram de ser mantidos dentro do ambiente de segurança do BCB.

Consequentemente, a manipulação dos dados - extração, tratamento, processamento, modelagem, treinamento, validação e teste - teve de ser realizada utilizando equipamentos fornecidos pela autarquia, em respeito à sua Política de Segurança da Informação. Assim, na configuração do ambiente de experimentos da presente pesquisa, foi utilizado um computador com processador Intel(R) Xeon(R) W-2255 com 3,7 GHz de velocidade e com 160 GB de memória RAM; GPU NVidia Quadro RTX 5000 com 16 GB de memória RAM dedicada; e sistema operacional de 64 bits, Windows 10 Enterprise.

4.3 Coleta e seleção dos dados

Para a obtenção dos dados do RDR, foram reproduzidas consultas elaboradas quando da construção do modelo de classificação vigente, que retornam informações diversas relacionadas às reclamações encaminhadas pelos cidadãos e respectivas respostas das entidades reclamadas, como campos de identificação da demanda, do cidadão e da IF; a data e a hora do seu registro; a existência de protocolo aberto pelo cidadão junto à instituição financeira; dentre outras. Foram ainda levantadas, por meio dessas consultas, informações históricas adicionais, como o total de demandas feitas pelo reclamante no último ano e eventuais reclamações do mesmo cidadão contra a mesma IF no último mês. Maiores descrições sobre os dados coletados podem ser encontradas na Seção 4.4.

Como experimentações com modelos de aprendizagem profunda e arquitetura de transformadores costumam apresentar alto custo computacional, geralmente é utilizada a GPU (*Graphics Processing Unit*), no intuito de acelerar as etapas de treinamento, validação e teste de modelos de DL. No entanto, mesmo com poderosas GPUs, o treinamento de transformadores pode levar dias [68, 69]. Assim, foi feito estudo prévio buscando avaliar a viabilidade dos experimentos considerando os recursos disponíveis, o cronograma estabelecido e o volume de dados alocado.

Inicialmente, foram selecionados os mesmos dados utilizados no desenvolvimento do modelo do BCB, abrangendo, portanto, todas as demandas abertas e encerradas no período de outubro de 2019 a dezembro de 2021, que totalizam cerca de 765 mil registros. Em seguida, foram efetuadas simulações buscando estimar o tempo gasto para o treinamento de modelos - usando o computador disponibilizado - a partir do mesmo *dataset* adotado pelo Banco Central. Os modelos considerados foram um baseado no atualmente vigente (TF-IDF + LightGBM) - referenciado como **Modelo BCB** - e um modelo hierárquico

[3]dado pessoal sensível: dado pessoal sobre origem racial ou étnica, convicção religiosa, opinião política, filiação a sindicato ou a organização de caráter religioso, filosófico ou político, dado referente à saúde ou à vida sexual, dado genético ou biométrico, quando vinculado a uma pessoa natural.

(BERT + LSTM) proposto com base em experimentos com textos longos conduzidos por [15] e [56] - referenciado como **Modelo Proposto**. Maiores detalhes sobre esses modelos podem ser encontrados nos Capítulos 5 e 6, respectivamente.

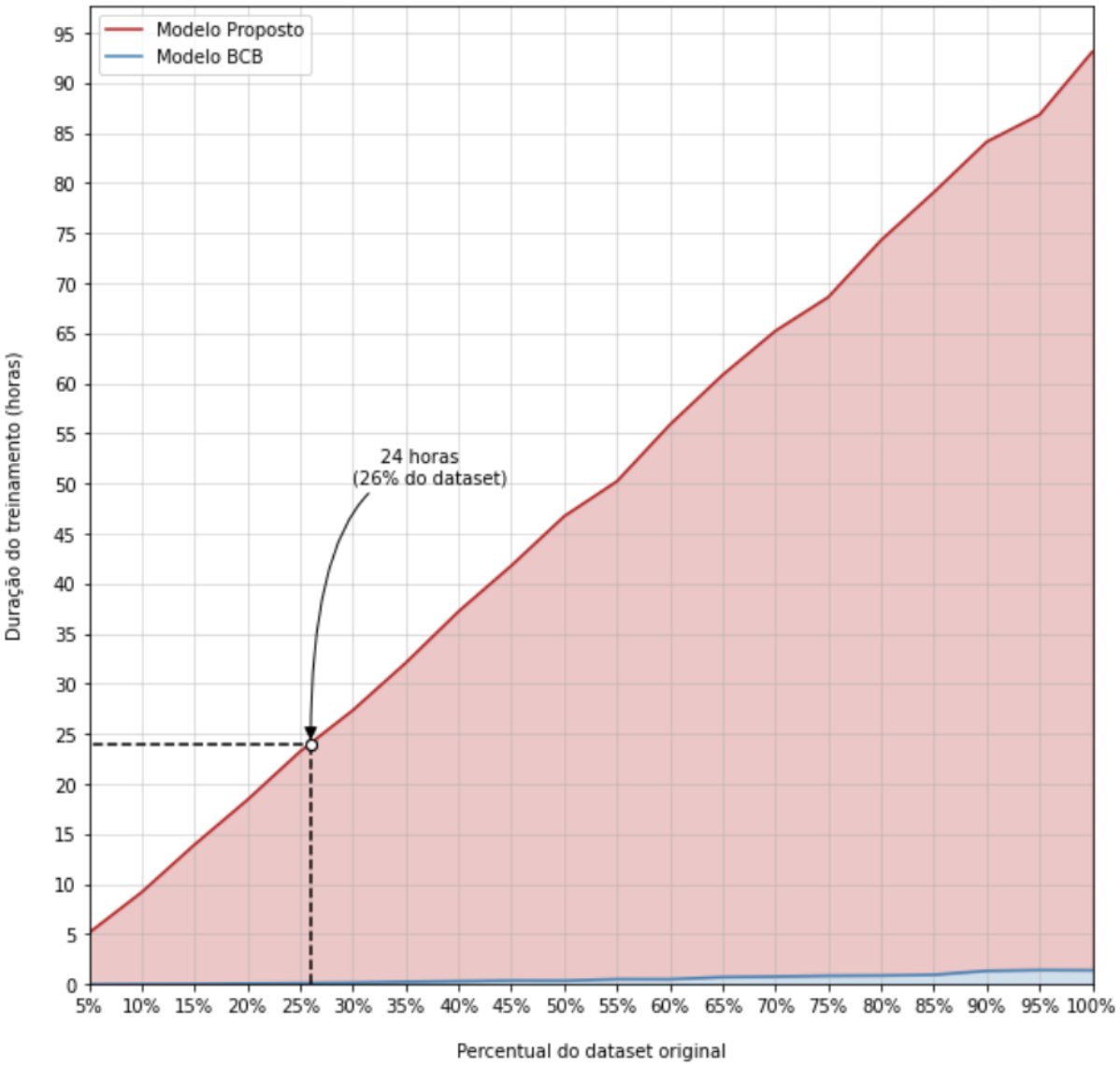


Figura 4.3: Simulação do tempo de treino dos modelos (Fonte: elaborado pelo autor).

O resultado das simulações pode ser observado na Figura 4.3. Os tempos de treinamento do Modelo BCB foram obtidos empiricamente, incrementando-se gradativamente o tamanho do *dataset* treinado. Para os tempos do Modelo Proposto, todavia, dada sua magnitude, foram registradas estimativas calculadas pelo algoritmo da Hugging Face[4] para cada proporção do *dataset* original. Observa-se que o treinamento do modelo de DL

[4] `https://huggingface.co/docs/transformers/main_classes/trainer`

com arquitetura de transformadores foi substancialmente maior que o do modelo tradicional. Dadas as limitações impostas pelo cronograma reservado para a pesquisa, seria inviável realizar experimentações com o modelo proposto utilizando a integralidade do *dataset*.

Destarte, foi selecionada uma parcela de 26% dos dados para a pesquisa. Estimou-se, com esse percentual, cerca de 24 horas (um dia) para o treinamento de cada modelo, tempo suficiente para finalizar os experimentos, mesmo considerando as variações do Modelo Proposto abordadas no Capítulo 6, e o processo de otimização e validação desses modelos. Adicionalmente, tendo em vista que a transferência de aprendizado a partir de um LM pré-treinado (como o BERT) permite a redução da quantidade de dados rotulados necessária para o aprendizado supervisionado em tarefas à jusante, conforme abordado na 2.4.1, assumiu-se que o volume de dados selecionado atenderia suficientemente à complexidade dos algoritmos de aprendizado em questão.

Do ponto de vista de negócio, seria interessante que os dados utilizados para o treinamento fossem o mais recentes o possível, visando aumentar a capacidade de generalização do modelo, em prol do seu desempenho em situações reais. A esse respeito, e alinhado com as justificativas anteriormente elencadas, ressalta-se que os dados do RDR de junho de 2021 - período em que houve, por decisões de gestão interna ao BCB, rotatividade dos servidores que efetuam a classificação da procedência das reclamações - até fevereiro de 2022 - dados mais atuais disponíveis - totalizaram cerca de 200 mil registros, aproximadamente 26% do *dataset* utilizado no desenvolvimento do classificador do BCB, o que reforça a escolha desse percentual para os experimentos da pesquisa.

Assim sendo, foram efetuadas adaptações nas consultas à base do RDR, visando a obtenção de dados referentes ao período de 01/06/2021 a 28/02/2022, e realizada a coleta no dia 24/03/2022, que retornou 199.495 registros. Para os experimentos, foram reservados 20% dos dados (39.899) para teste e 80% (159.596) para treinamento. Como o *dataset* se mostrou desbalanceado em relação à procedência das reclamações, questão melhor abordada na Subseção 4.4.1, sua partição ocorreu de forma estratificada, no intuito de preservar a mesma proporção de instâncias em cada classe que aquela observada originalmente no dados de experimento.

4.4 Análise dos dados

Para a apresentação dos dados nesta Seção, foram, primeiramente, avaliados os rótulos de procedência das reclamações envolvidos na tarefa de classificação desejada. Em seguida, foram analisadas, individualmente, as variáveis textuais e tabulares - que compreendem

as categóricas e numéricas - selecionadas na construção do modelo do BCB, que adota o método tradicional discorrido na Seção 2.3.

4.4.1 Rótulos de procedência das reclamações

Como mencionado na Seção 4.1, o processo de tratamento das reclamações consiste, de modo geral, no seu enquadramento como "procedente" ou "improcedente" a partir da análise do relato do cidadão, da resposta da IF, de eventuais documentos anexados à demanda e de julgamento sobre se houve ou não indício de descumprimento da regulamentação vigente. No RDR, contudo, o registro dessa classificação é feito considerando categorias mais granularizadas, definidas quando da criação do sistema. Consequentemente, foi necessário um mapeamento para se obter os rótulos desejados.

Tabela 4.1: Mapeamento dos rótulos de procedência das reclamações

Classificação RDR	**Rótulo de procedência**
Cancelada após resposta da IF/AC	Improcedente
Falta de dados (Reclamação)	Improcedente
Não conclusiva	Improcedente
Reclamação cancelada	Improcedente
Reclamação não regulada	Improcedente
Reclamação regulada improcedente	Improcedente
Repasse de dados não autorizado	Improcedente
Reclamação regulada procedente	Procedente
Unidade Supervisora Informada	Procedente

As consultas referidas na Seção 4.3 retornam apenas as demandas encerradas, uma vez que as categorias de reclamações em andamento não permitem o seu enquadramento quanto à procedência. A partir dessas, foi então feito um último mapeamento, ilustrado na Tabela 4.1. Na Figura 4.4 é ilustrada a proporção das categorias e rótulos no *dataset* de experimento, percebendo-se um nítido debalanceamento entre as classes existentes (aproximadamente 30% para procedentes e 70% para improcedentes), que se mostrou inalterado no período selecionado.

4.4.2 Variáveis textuais

Para a classificação das demandas quanto à sua procedência, as principais informações avaliadas são os textos do relato do cidadão e da resposta da IF, sendo estes, portanto, variáveis imprescindíveis para a aludida tarefa. Conforme descrito na Seção 4.1, a reclamação do cidadão pode ser encaminhada por diversos canais, sendo, posteriormente, centralizada em campos de texto no sistema RDR. O BCB então repassa a demanda para

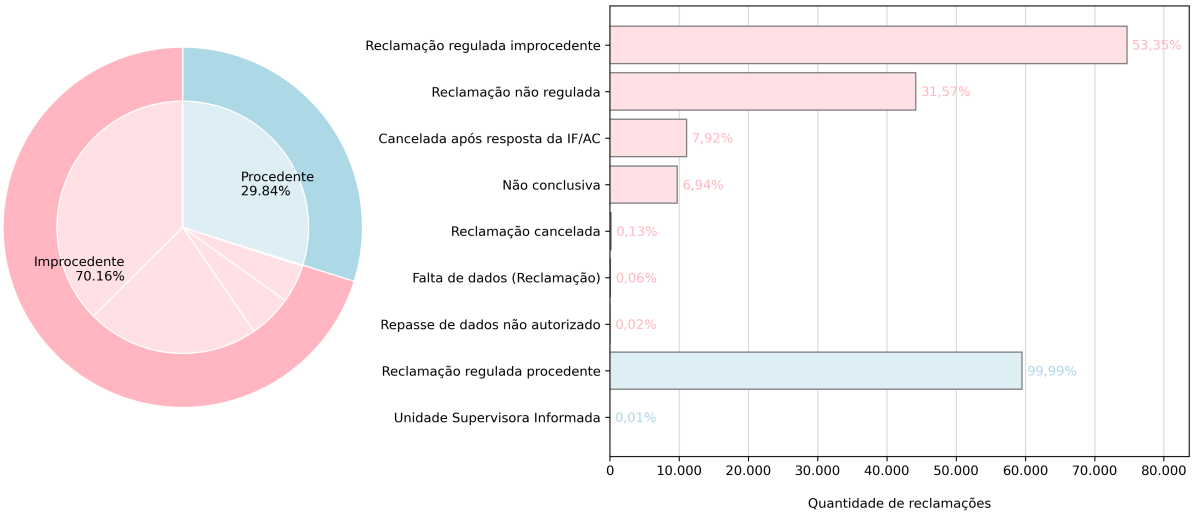


Figura 4.4: Proporção dos rótulos e das classes do RDR (Fonte: elaborado pelo autor).

a entidade reclamada, que deve apresentar uma resposta em até 10 dias. A IF pode ainda complementar suas informações com anexos de conteúdo livre.

Observou-se que os textos apresentam características bastante diversificadas, variando em aspectos como tamanho (curto, extenso), conteúdo (rico, pobre), linguagem (culta, coloquial) e ortografia (correta, incorreta). Como exemplo, as Figuras 4.5 e 4.6 apresentam a reclamação e a resposta[5] de demandas procedente e improcedente, respectivamente. É ainda ilustrada, a título informativo, a classificação realizada pelo servidor do Banco Central, devidamente justificada.

Na presente pesquisa, foram adotadas, para as variáveis textuais, duas representações comumente utilizadas na literatura: frequência dos termos do texto normalizada pela frequência inversa do documento (TF-IDF), e *embeddings* de contexto gerados a partir de um Modelo de Linguagem pré-treinado. Essas representações são melhor descritas nos Capítulos 5 e 6, respectivamente.

O conteúdo dos anexos também é de grande relevância para a categorização das demandas. No entanto, dada a sua grande diversidade (textos, imagens, áudios, arquivos comprimidos, dentre outros), não foi possível para os especialistas da área de negócio convertê-los em dados estruturados em tempo hábil para os experimentos.

Outra potencial fonte de conhecimento, também mencionada na Seção 4.1, são as "súmulas" e "jurisprudências" dos atendimentos realizados, que consistem em grandes volumes de textos oriundos de normativos e documentos internos e externos do BCB. Todavia, diante da sua característica não estruturada, massiva e de alta volatilidade -

[5]Nos exemplos, foram ocultadas informações que permitissem a identificação dos agentes envolvidos.

Reclamação:

O [redacted] não HONRA com os contratos disponiveis em seus canais digitais. Eu tive um imprevisto e ficou pendente a ultima parcela do parcelamento do [redacted] que venceu em [redacted]. O instituição me concedeu renegociação para que eu pudesse colocar em dia essa pendencia de forma parcelada. Fiz o procedimento e concordei com o novo contrato da seguinte forma: Entrada para o dia [redacted] de [redacted] de [redacted] e mais quatro parcelas com vencimento a primeira em [redacted] de [redacted]. Paguei a entrada ontem dia [redacted] de [redacted] de [redacted] e tinha entrado valor na conta para cobrir o limite do [redacted] que estava sendo utilizado, e o banco cobrou o valor da parcela integral mais o juros mesmo com o acordo firmado um dia antes. Entrei em contato no chat e o atendente informou que não poderia fazer nada pois era apenas um funcionário e que eu deveria aguardar um prazo de [redacted] dias uteis para ter o estorno do valor, seja ele a entrada ou o valor indevido. Entrei nos canais de atendimento via telefone e também sem sucesso. Numero do contrato realizado para parcelamento da pendencia [redacted]. Segue comprovante do pagamento da entrada do acordo e debito do valor integral de forma não proposta no acordo firmado.

Resposta:

[redacted], Conforme seu relato, identificamos que você possuía o contrato [redacted], efetivado em [redacted], que estava em atraso desde [redacted], referente à parcela [redacted]. Em [redacted] você renegociou as condições do seu contrato, realizando o primeiro pagamento em [redacted]. Neste sentido, a efetivação desta oferta e a interrupção das condições gerais do contrato negociado ocorre mediante a identificação do pagamento realizado em [redacted]. Com relação aos lançamentos mencionados, verificamos que ocorreram durante o período em que a renegociação estava sendo processada em nosso sistema. Identificamos suas interações com nossos canais de atendimento em [redacted], por meio dos quais orientamos a aguardar o estorno. Sendo assim, os valores debitados foram estornados na mesma data do respectivo débito, sendo creditado o valor de [redacted] em sua conta corrente [redacted] no dia [redacted], conforme demonstrado em seu extrato. Agradecemos a oportunidade de resposta e esperamos ter auxiliado com os esclarecimentos prestados. Temos o compromisso com a satisfação dos nossos clientes e trabalhamos de forma contínua na análise das demandas para identificar oportunidades de melhorias em nossos processos, produtos e serviços. Atenciosamente, [redacted].

Classificação:

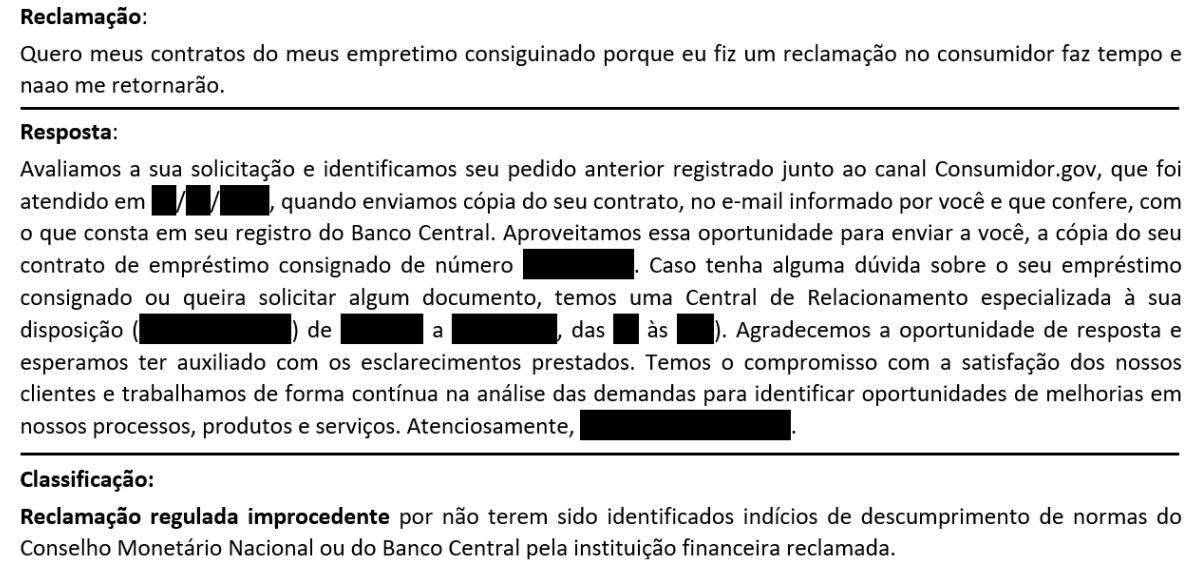
Se havia acordo firmado, a IF deveria suspender eventuais débitos existentes até a data acordada para o primeiro pagamento do acordo. Portanto, diante desta imprudência, impõe-se considerar a **reclamação regulada procedente**, tendo em vista a existência de indícios de descumprimento de disposições do artigo 3º da Resolução CMN 4.790/2020.

Figura 4.5: Exemplo de demanda procedente.

resultado de publicações periódicas e recorrentes -, esses dados se mostraram inviáveis para a construção do classificador almejado.

Reclamação do cidadão

De modo geral, as reclamações do cidadão costumam ser menores que as respostas da IF (questão melhor descrita na Subseção 4.4.3), bem como apresentar maior quantidade de termos coloquiais e erros ortográficos. No entanto, podem ser encontrados tanto relatos mais extensos e melhor detalhados, como o exemplificado na Figura 4.5, quanto textos curtos, sem maiores informações e com erros ortográficos, como o constante da Figura 4.6. A respeito, reclamações mais claras e ricas em informações relevantes costumam apresentar maior possibilidade de serem categorizadas como procedentes, uma vez que facilitam ao analista do BCB avaliar se houve ou não indício de descumprimento da regulamentação incidente.



Reclamação:

Quero meus contratos do meus empretimo consiguinado porque eu fiz um reclamação no consumidor faz tempo e naao me retornarão.

Resposta:

Avaliamos a sua solicitação e identificamos seu pedido anterior registrado junto ao canal Consumidor.gov, que foi atendido em [redacted], quando enviamos cópia do seu contrato, no e-mail informado por você e que confere, com o que consta em seu registro do Banco Central. Aproveitamos essa oportunidade para enviar a você, a cópia do seu contrato de empréstimo consignado de número [redacted]. Caso tenha alguma dúvida sobre o seu empréstimo consignado ou queira solicitar algum documento, temos uma Central de Relacionamento especializada à sua disposição ([redacted]) de [redacted] a [redacted], das [redacted] às [redacted]). Agradecemos a oportunidade de resposta e esperamos ter auxiliado com os esclarecimentos prestados. Temos o compromisso com a satisfação dos nossos clientes e trabalhamos de forma contínua na análise das demandas para identificar oportunidades de melhorias em nossos processos, produtos e serviços. Atenciosamente, [redacted].

Classificação:

Reclamação regulada improcedente por não terem sido identificados indícios de descumprimento de normas do Conselho Monetário Nacional ou do Banco Central pela instituição financeira reclamada.

Figura 4.6: Exemplo de demanda improcedente.

Resposta da Instituição Financeira

Os textos da resposta da IF geralmente são mais extensos, contendo informações sobre o seu vínculo com o cidadão (como número de contrato, valores de empréstimos, e datas de atendimento). No entanto, verificou-se que diversas respostas apenas referenciam o anexo encaminhado, sendo, portanto, enxutas e sem conteúdo de maior relevância, como se observa nos exemplos que seguem:

```
Prezados, boa tarde! Segue em anexo a resposta.
Boa Noite. Segue anexo, carta resposta.
Prezados Senhores, anexamos resposta encaminhada ao demandante.
```

Como já antecipado, não foi possível acessar o conteúdo dos anexos, em detrimento da qualidade das informações dessas respostas, o que resultou em limitação da pesquisa. Contudo, como o modelo do BCB foi implementado sem considerar esse conteúdo, a manutenção dos mesmos dados permitiu uma comparação mais justa com os modelos propostos na etapa de avaliação.

4.4.3 Variáveis categóricas

Além dos atributos textuais descritos anteriormente, o modelo implementado pelo Banco Central contou ainda com três variáveis categóricas, buscando, no âmbito da engenharia de variáveis característica de abordagens tradicionais, extrair outros conhecimentos

potencialmente relevantes para a classificação das reclamações quanto à sua prodecência, relacionados a aspectos contextuais envolvendo o cidadão, a entidade reclamada e a demanda envolvida.

Demandas anteriores

Foram levantados, para cada demanda encerrada, dados históricos relacionados a outras relamações por ventura abertas pelo mesmo cidadão contra a mesma IF nos últimos 30 dias. Assim, essa variável booleana informa se houve interações recentes entre o reclamante e o reclamado, buscando acrescentar conhecimentos contextuais para a classificação desejada. A Figura 4.7 apresenta, com base no *dataset* selecionado para experimentos, a proporção de demandas com interações anteriores no mês e sua respectiva distribuição quanto à procedência. Verifica-se que reclamações com histórico prévio, embora em menor quantidade (11,4% do total), apresentaram maior proporção de improcedência.

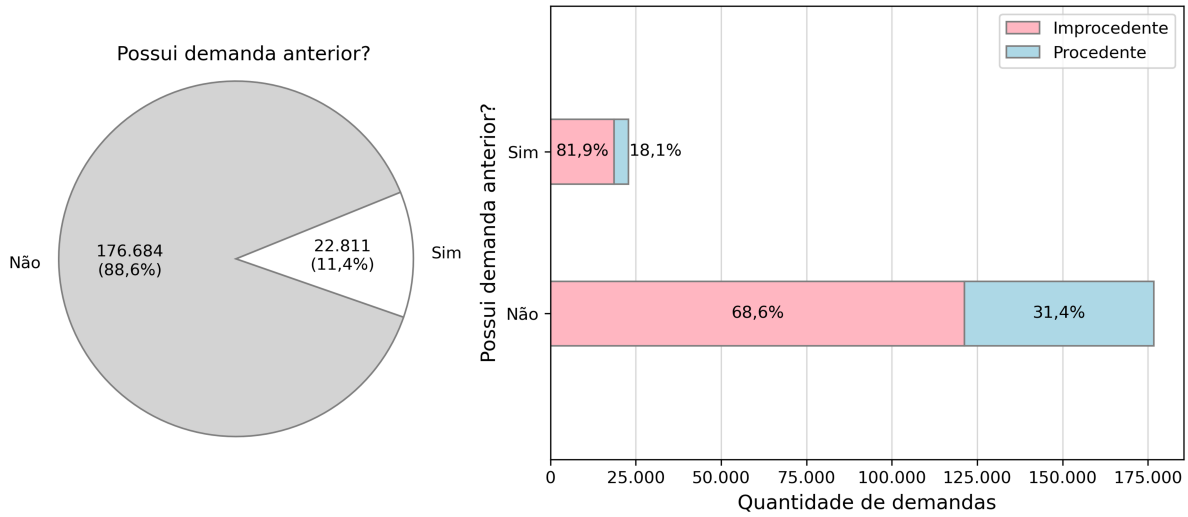


Figura 4.7: Demandas com reclamação anterior (Fonte: elaborado pelo autor).

Protocolos abertos

Ao encaminhar uma demanda ao BCB, o cliente ou usuário do SFN pode registrar o número de eventual protocolo aberto junto à IF, caso a reclamação já tenha sido endereçada a ela diretamente pelo cidadão. A presente variável categórica informa sobre a existência prévia de protocolo em cada uma das demandas encerradas, buscando acrescentar dados sobre o contexto da reclamação. É ilustrado, na Figura 4.8, a proporção de demandas com protocolos aberto e a distribuição quanto à sua procedência. Observa-se que as reclama-

ções sem protocolo prévio (cerca de metade das demandas) apresentaram um percentual maior de improcedência.

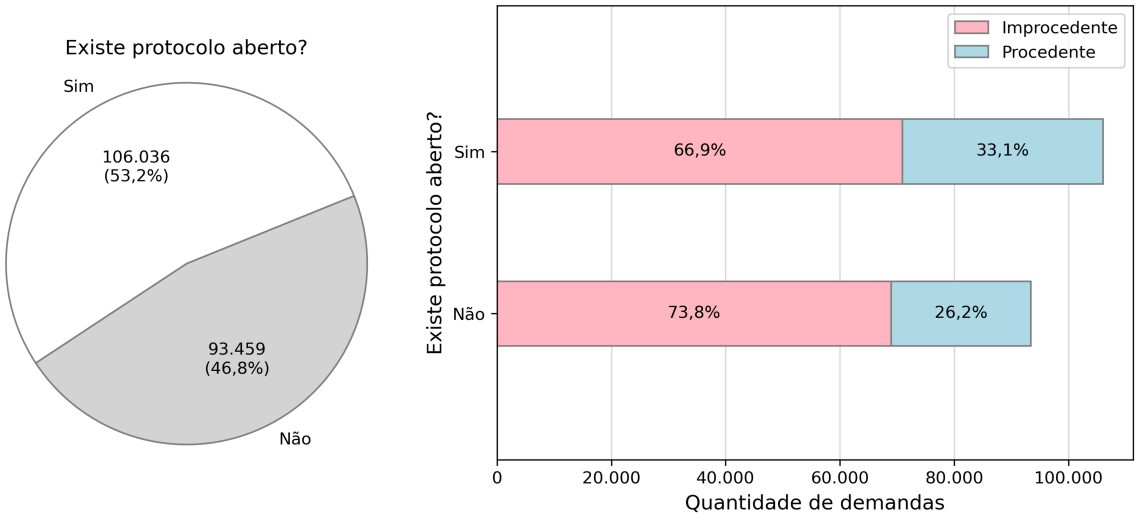


Figura 4.8: Demandas com protocolo aberto (Fonte: elaborado pelo autor).

Segmento de Instituição Financeira

As instituições autorizadas, reguladas ou supervisionadas pelo Banco Central podem ser classificadas em segmentos, discriminados no portal do BCB[6], de acordo com sua respectiva área de atuação. Assim, esta variável qualitativa busca auxiliar a tarefa de classificação das demandas fornecendo informações sobre a IF reclamada, especificamente o seu segmento no SFN. A Figura 4.9 apresenta a distribuição desses segmentos no *dataset* de experimentos, com as proporções de procedência de cada um deles (os valores tabelados podem ser observados no Apêndice A.1). Observa-se que o percentual de improcedência pode variar consideravelmente de acordo com o segmento da IF (a exemplo de 66,0% em Instituição de Pagamento contra 79,2% em Banco do Brasil - Banco Múltiplo).

4.4.4 Variáveis numéricas

Foram também consideradas variáveis númericas buscando trazer dados quantitativos (discretos e contínuos) a partir do conteúdo das variáveis textuais retromencionadas, ou relacionados ao contexto do reclamante e da IF reclamada.

[6] `https://www.bcb.gov.br/estabilidadefinanceira/encontreinstituicao`

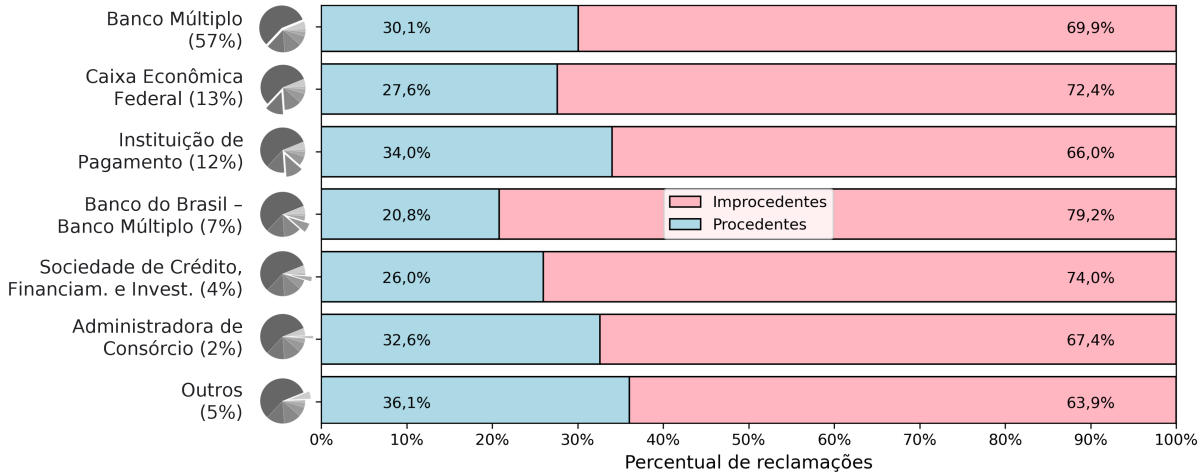


Figura 4.9: Demandas por segmento da IF (Fonte: elaborado pelo autor).

Reclamações por ano

A primeira variável númerica considerada é referente à quantidade de reclamações - encerradas ou não - abertas pelo cidadão nos últimos 365 dias (último ano), contados a partir da data de abertura da demanda. Analogamente, foram ainda levantados, para o mesmo período, dados históricos relacionados ao total de demandas encaminhadas pelo reclamante e que foram encerradas como improcedentes e procedentes, resultando, assim, em mais dois atributos. As Figuras 4.10 a 4.12 tratam da distribuição dessas variáveis no *dataset* de experimentos, com a respetiva proporção quanto à procedência das demandas.

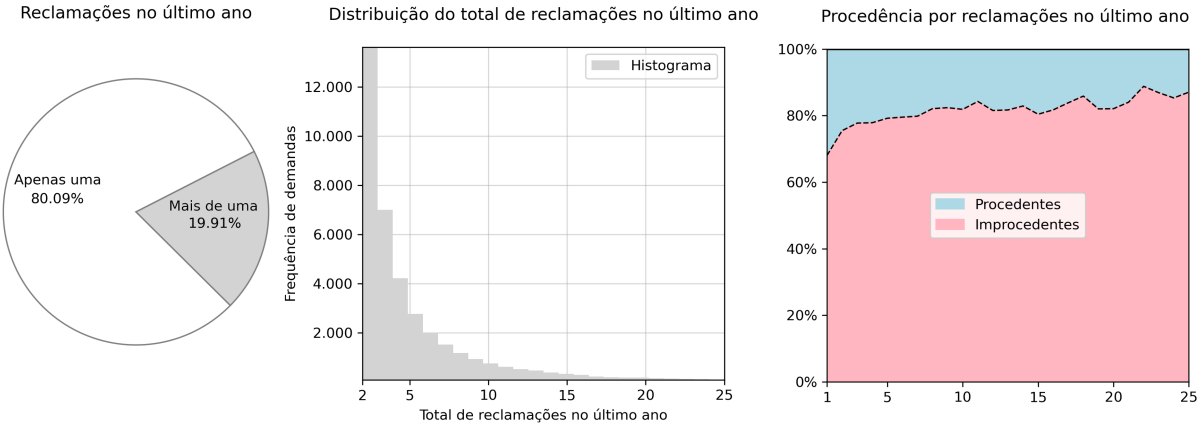


Figura 4.10: Demandas por total de reclamações no ano (Fonte: elaborado pelo autor).

Observa-se, na Figura 4.10, que aproximadamente 20% das demandas foram abertas por cidadãos com mais de uma reclamação no último ano. Para essa proporção, percebe-se que, à medida que o valor da variável numérica aumenta, a frequência de demandas obser-

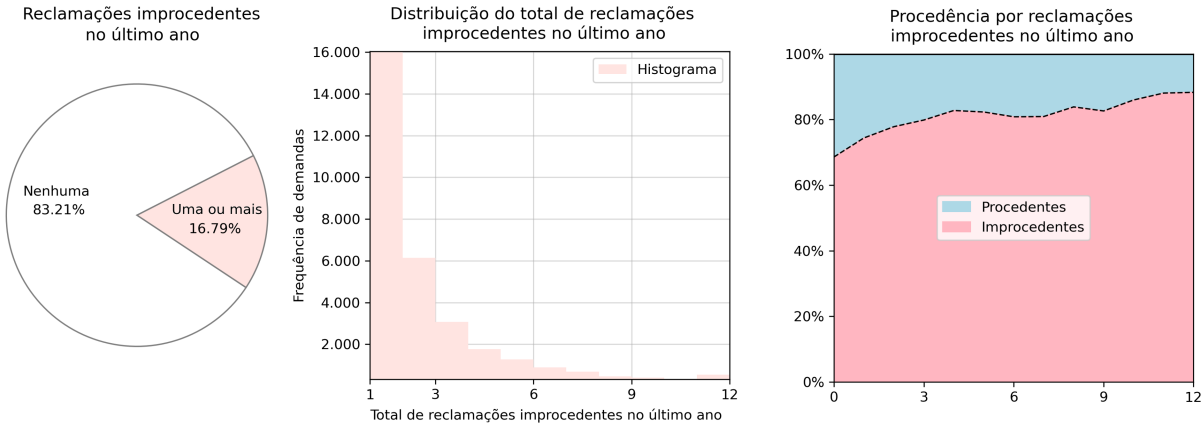


Figura 4.11: Demandas por reclamações improcedentes no ano (Fonte: elaborado pelo autor).

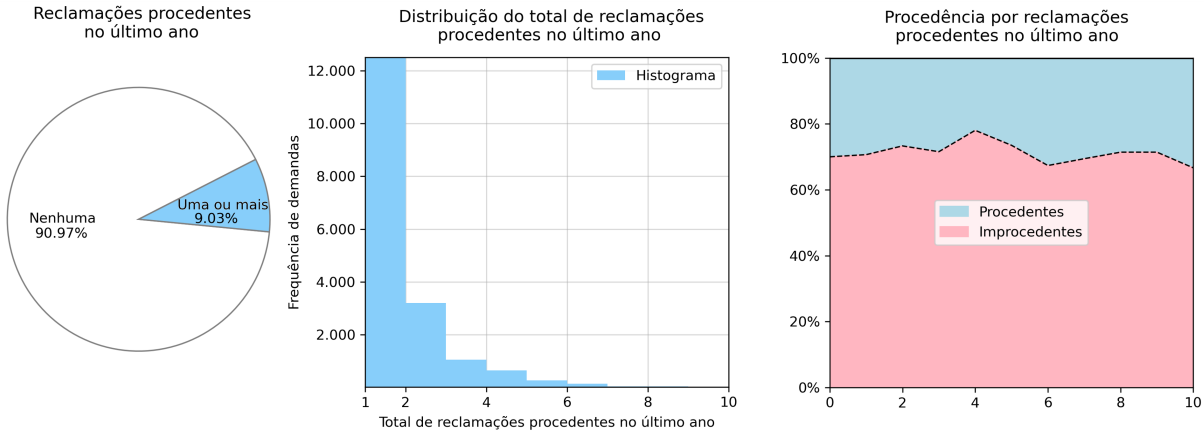


Figura 4.12: Demandas por reclamações procedentes no ano (Fonte: elaborado pelo autor).

vadas decresce exponencialmente, porém com um suave crescimento no seu percentual de improcedência. Um comportamento similar pode também ser observado na Figura 4.11, referente ao total de reclamações *improcedentes* por ano. No entanto, a proporção de procedência das demandas parece se manter relativamente estável para diferentes valores da terceira variável numérica (total de reclamações *procedentes* por ano), como ilustra a Figura 4.12.

Defasagem de data

A procedência da demanda pode ainda depender de aspectos temporais estabelecidos pela regulamentação vigente, a exemplo de prazos para pagamento de empréstimo consignado. No entanto, o único dado de tempo disponível de forma estruturada é a data de abertura

da reclamação. Não obstante, recorrentemente são encontradas passagens no relato do cidadão e na resposta da IF contendo datas de eventos relacionados à demanda. Assim sendo, buscando extrair e estruturar esses dados temporais, no intuito de agregar maiores informações para a tarefa de classificação, foram definidas variáveis de defasagem de data.

Sua construção ocorre na etapa de pré-processamento (descrita na Seção 5.1), na qual, primeiro, são extraídas as datas dos campos de texto (reclamação e resposta) e, em seguida, calculadas sua respectivas defasagens, em dias, da data de abertura da demanda (fixada como referência). No caso de uma data extraída ser anterior à da reclamação, são considerados números de dias negativos. Enfim, do resultado obtido, são então adotados, como atributos, os seguintes valores:

- **Defasagem mínima na reclamação**: *Menor* número de dias transcorridos no texto da *reclamação*.
- **Defasagem máxima na reclamação**: *Maior* número de dias transcorridos no texto da *reclamação*.
- **Defasagem mínima na resposta**: *Menor* número de dias transcorridos no texto da *resposta*.
- **Defasagem máxima na resposta**: *Maior* número de dias transcorridos no texto da *resposta*.
- **Defasagem mínima global**: *Maior* número de dias transcorridos nos textos da *reclamação e da resposta*.
- **Defasagem máxima global**: *Menor* número de dias transcorridos nos textos da *reclamação e da resposta*.

Para facilitar o entendimento dessas variáveis, a Figura 4.13 ilustra um exemplo de datas extraídas dos textos[7] de uma demanda procedente aberta em 26/08/2021. A partir da data de referência, calculam-se os seguinte valores:

```
Defasagem mínima na reclamação: 18/08/2021 - 26/08/2021 = - 8 dias
Defasagem máxima na reclamação: 26/08/2021 - 26/08/2021 =   0 dias
Defasagem mínima na resposta:   31/08/2021 - 26/08/2021 =   5 dias
Defasagem máxima na resposta:   08/09/2021 - 26/08/2021 =  13 dias
Defasagem mínima global:        18/08/2021 - 26/08/2021 = - 8 dias
Defasagem máxima global:        08/09/2021 - 26/08/2021 =  13 dias
```

[7]Foram ocultadas informações que permitissem a identificação dos agentes envolvidos e alterados os valores de data apresentados.

Reclamação:

No dia 18/08/2021 por volta das [redacted], fiz um deposito de [redacted] em uma caixa eletrônico de deposito imediato na agência localizada [redacted], cujo endereço é [redacted]. O caixa eletrônico informou que algumas notas não foram reconhecidas, foi pressionado botão para cancelar a operação, o caixa eletrônico devolveu [redacted], ficando preso na máquina [redacted]. O [redacted] foi contatado no mesmo momento e informou que seria aberta solicitação para abertura do caixa e que levaria [redacted] dias úteis, finalizando o prazo na data de 25/08/2021. Finalizado prazo, o [redacted] informou que a solicitação foi concluída e que após auditoria do equipamento não identificando falha no equipamento, não foi favorável o ressarcimento por não localizar erro na transação ou valor retido (palavras da atendente constada em gravação na data de 26/08/2021). Por esse motivo, perdi [redacted], além do nado financeiro, não consegui realizar pagamento de contas de meu consumo, gerando atraso e juros. Desejo ressarcimento do meu valor.

Resposta:

Em resposta à manifestação registrada no Banco Central do Brasil, esclarecemos que em conversa com a gerência [redacted], fomos informados que em 31/08/2021 houve a solicitação de conferência do equipamento. Neste sentido, após a análise de nosso equipamento, em 08/09/2021 efetuamos a devolução de [redacted] em sua Conta Corrente. Por fim, o protocolo [redacted] informado não é válido nesta [redacted]. Atenciosamente, [redacted].

Classificação:

Reclamação regulada procedente, tendo em vista a existência de indícios de descumprimento de disposições do art. 1º, inciso II, da Resolução 3.694/2009, com redação dada pela Resolução 4.283/2013. Indícios de falhas no depósito reclamado.

Figura 4.13: Exemplo de datas em demanda procedente.

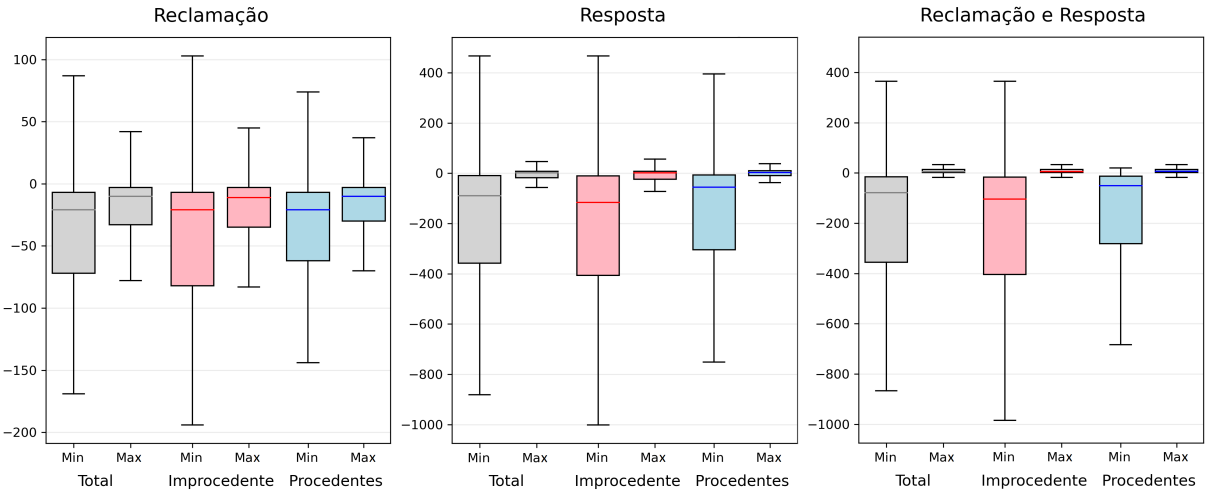


Figura 4.14: Demandas por defasagem de datas no texto (Fonte: elaborado pelo autor).

Constatou-se, no *dataset* de experimento, que aproximadamente 45% das reclamações e 60% das respostas tiveram ao menos uma data extraída. A distribuição das defasagens encontradas nesses percentuais, e a procedência das resepectivas demandas, são apresentadas na Figura 4.14. Observa-se que, em ambos os campos de texto, as medianas dos *boxplots* estão próximas de zero, indicando predominância de datas orbitando o momento

da abertura da reclamação. Adicionalmente, poucos valores positivos foram encontrados, sugerindo referências escassas à eventos futuros. Em contrapartida, a dispersão das defasagens mínimas é bem maior que a das defasagens máximas, principalmente nas respostas das IFs, indicando uma pluralidade nas datas de eventos passados mencionados nos textos. No entanto, não foram observadas mudanças nítidas quanto à procedência das reclamações no âmbito dessas variáveis.

Tamanho das variáveis textuais

Outro atributo que pode ser obtido a partir das variáveis textuais é o seu tamanho, em termos de quantidade de caracteres. Como já mencionado na Subseção 4.4.2, o tamanho dos textos pode variar consideralmente. Contudo, as respostas das IFs costumam ser mais robustas e, portanto, de maior extensão do que os relatos do cidadão. As Figuras 4.15 e 4.16 contêm a frequência de demandas por procedência e quantidade de caracteres das reclamações e respostas, respectivamente.

Quanto aos rótulos das demandas, constatou-se que reclamações de menor tamanho apresentam maior proporção de improcedência, possivelmente por não conterem informações relevantes suficientes para a classificação realizada pelo servidor do BCB. Por outro lado, não foram percebidas mudanças significativas na distribuição da procedência associadas à variações na extensão do texto da resposta da entidade reclamada.

Na Figura 4.15, observa-se um crescimento na frequência de demandas com reclamações contendo mais de 3.000 caracteres, seguido de uma queda abrupta em 3.200 caracteres. A esse respeito, foram realizadas consultas à área de negócio do BCB, buscando justificativas para o aludido comportamento. Contudo, em que pese existirem suspeitas de inconsistências no código de extração dos dados disponibilizados, não foi possível identificar a origem do problema. No entanto, verificou-se que os dados utilizados na construção do modelo do BCB também apresentaram o mesmo comportamento, de modo que a limitação técnica apresentada não prejudicaria a comparação dos modelos na etapa de avaliação.

Outra limitação já apresentada na Subseção 4.4.2 é referente a respostas cujo texto apenas faz referência ao conteúdo dos anexos, que, todavia, não foram disponibilizados para a pesquisa. Na Figura 4.16, percebe-se uma nítida concentração de demandas com respostas de menos de 500 caracteres, correspondendo a cerca de 20% do total, o que pode ser justificado pela limitação retromencionada.

Similaridade das variáveis textuais

Quando da classificação das demandas, o servidor do BCB pode recorrer a informações contextuais históricas, como a existência de outras reclamações ou respostas previamente

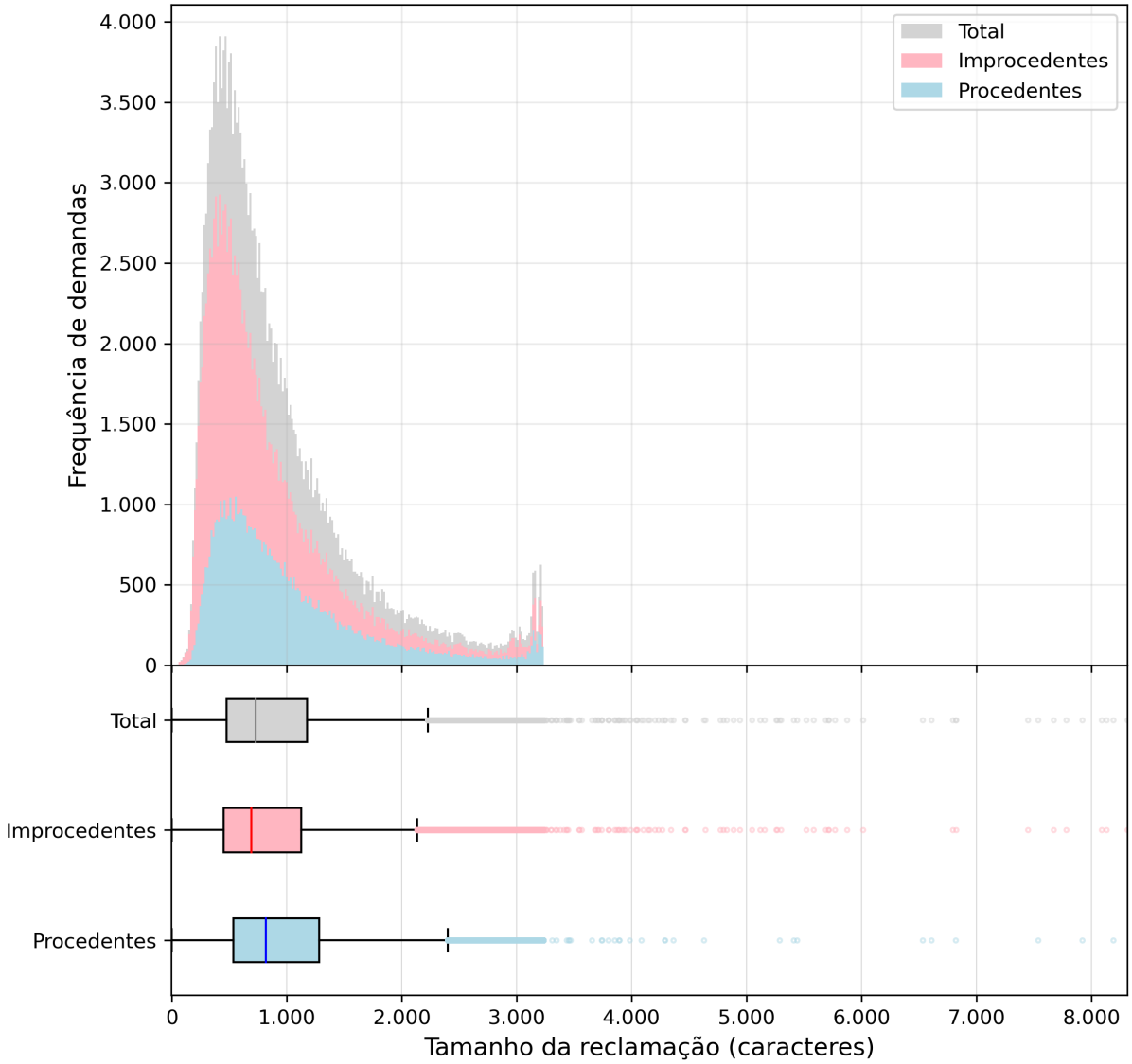


Figura 4.15: Demandas por tamanho da reclamação (Fonte: elaborado pelo autor).

encaminhadas pelo mesmos agentes (reclamante e reclamado). Ocasionalmente, uma demanda pode vir com relatos repetidos, tratando de mesmo evento, possivelmente já com julgamento quanto ao descumprimento da norma, não cabendo, portanto, nova marcação de procedência. De forma análoga, respostas idênticas por parte da IF podem indicar recorrência de fatos já abordados, influenciando na categorização em questão.

Como a variável categórica descrita no item *Demandas anteriores* da Subseção 4.4.3 trata apenas da existência de outras demandas abertas pela mesmo cidadão contra a mesma entidade reclamada nos últimos 30 dias, foram então acrescidos dois atributos quantitativos, buscando informar a similaridade entre os textos atuais e os anteriores.

Para isso, foi calculada a semelhança de cosseno, uma das medida mais utilizadas na

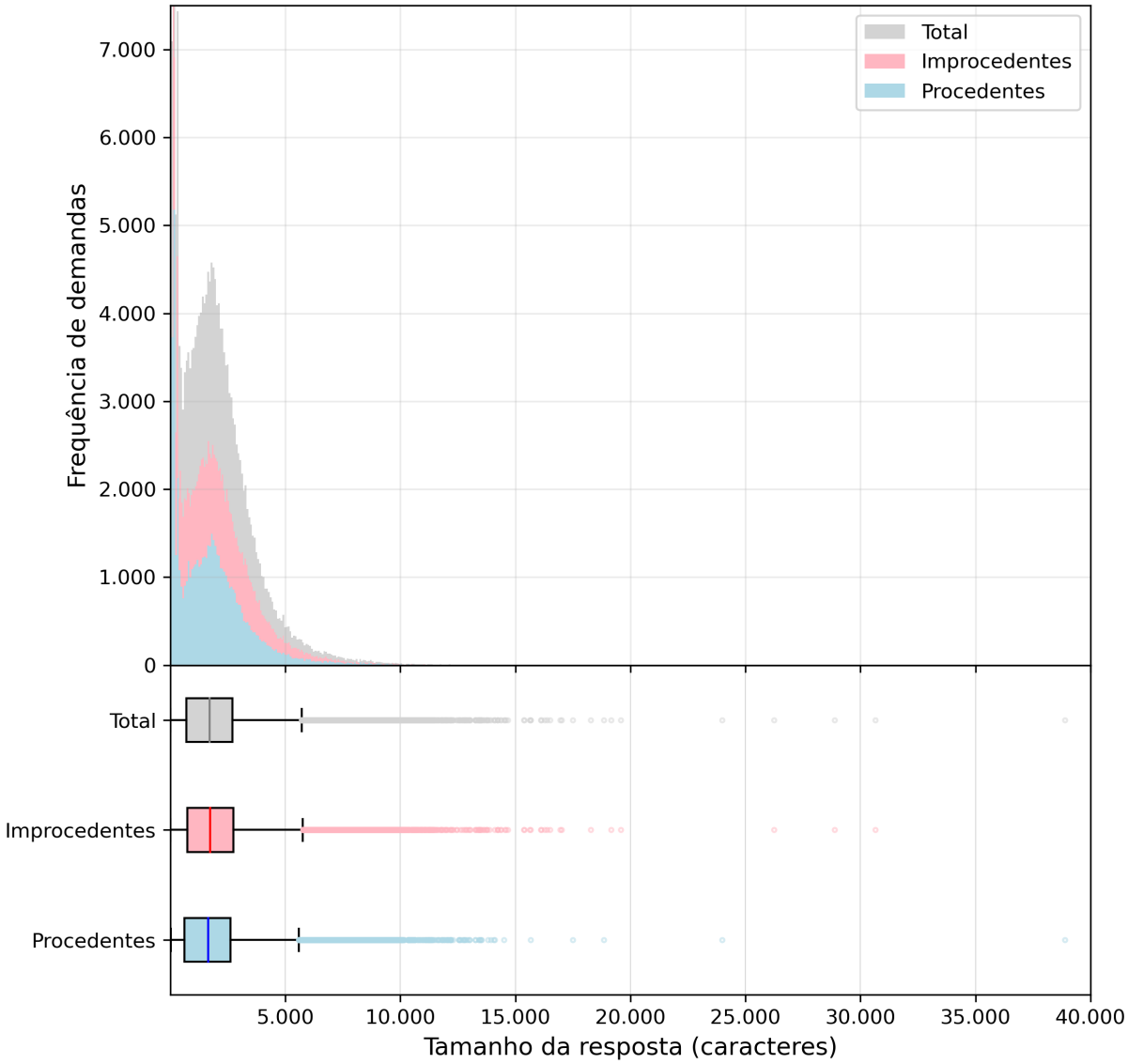


Figura 4.16: Demandas por tamanho da resposta (Fonte: elaborado pelo autor).

literatura para obter a similaridade de dois documentos (A e B) [35], obtida conforme descrito na Equação 4.1, sendo que similaridades próximas a 1 indicam documentos mais parecidos. A obtenção das variáveis de similaridade, assim como a das variáveis de defasagem de data tratadas no item *Defasagem de data* desta Subseção, ocorre na etapa de pré-processamento, uma vez que o seu cálculo demanda representações vetoriais dos dados de texto.

$$S(\mathbf{A},\mathbf{B}) = \cos(\theta) = \frac{\mathbf{A}.\mathbf{B}}{||\mathbf{A}||||\mathbf{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2}\sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{4.1}$$

A distribuição de similaridade e procedência de demandas com textos anteriores pode

ser observada nas Figuras 4.17 - para reclamações - e 4.18 - para respostas. Observa-se, em ambos os casos, que existe uma concentração de demandas com similaridade alta, indicando possível reincidência de relatos sobre eventos já abordados anteriormente. Adicionalmente, analisando as situações de similaridade máxima, observa-se uma proporção maior de improcedência, principalmente nos textos das reclamações.

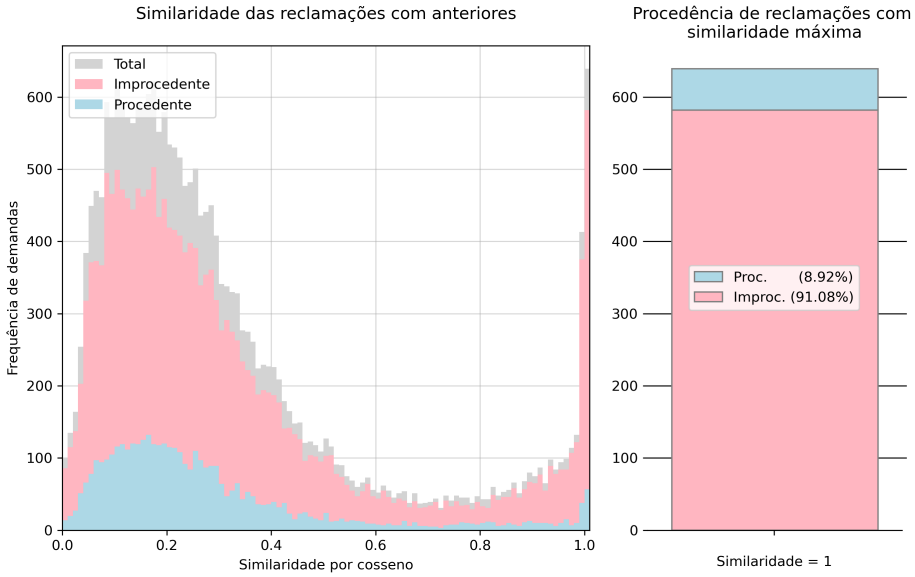


Figura 4.17: Demandas por similaridade da reclamação (Fonte: elaborado pelo autor).

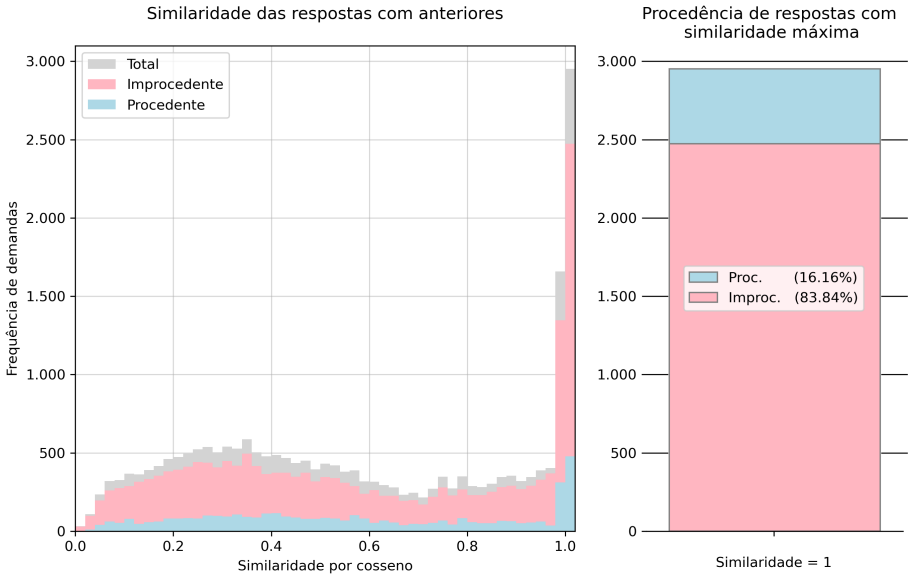


Figura 4.18: Demandas por similaridade da resposta (Fonte: elaborado pelo autor).

Capítulo 5

Reprodução do Modelo do BCB

O classificador do BCB foi construído como resultado de experimentos com diferentes representações de texto e modelos clássicos de ML, culminando na solução atual, composta pela representação TF-IDF e pelo modelo LightGBM, visto ter apresentado melhor desempenho para a tarefa de categorização das demandas dos cidadãos. Conforme descrito na Seção 2.3, o TF-IDF é uma das representações mais utilizadas no âmbito de métodos tradicionais, ao passo que o LightGBM é um algoritmo de ML muito conceituado na literatura por sua eficiência, performance e interpretabilidade, obtendo ótimos resultados para tarefas de categorização de textos. Assim sendo, o modelo vigente foi assumido como linha-de-base e referência para o desenvolvimento dos modelos propostos nesta pesquisa.

Contudo, pelos motivos expostos na Seção 4.3, não foi possível manter, para os experimentos conduzidos neste estudo, o mesmo conjunto de dados utilizados na criação do modelo atual. Consequentemente, no intuito de viabilizar comparações com os classificadores propostos, foi necessário, primeiro, reproduzir a metodologia adotada na construção do modelo do Banco Central, porém considerando o novo *dataset* selecionado para os experimentos, de modo a obter um classificador para representar aquele desenvolvido pela autarquia.

Diante do exposto, este capítulo descreve as etapas para a reprodução do Modelo do BCB, abordando desde o pré-processamento das entradas até o treinamento e otimização dos modelos. É ainda descrita a métrica de desempenho selecionada para avaliar os classificadores e estimar sua habilidade de generalização no mundo real.

5.1 Pré-processamento das entradas

O pré-processamento dos dados pode ser considerado um dos principais componentes em muitos algoritmos de NLP [35]. Nesta etapa, os dados são tratados e preparados, visando torná-los adequados à aplicação dos algoritmos selecionados para a modelagem [70, 31].

Na metodologia utilizada na criação do modelo do BCB, que adota uma abordagem clássica, essa etapa consistiu, basicamente, no tratamento dos dados textuais, na sua tokenização, no cálculo do TF-IDF e na criação de variáveis a partir dos textos. Embora o fluxo apresentado na Seção 2.2 aloque a extração dos atributos posteriormente ao pré-processamento, na reprodução do Modelo BCB, para fins práticos, essa atividade é descrita na presente etapa.

5.1.1 Variáveis de defasagem de datas e tamanho do texto

O pré-processamento efetuado pelo Banco Central conta, primeiramente, com a extração de datas dos textos da reclamação do cidadão e da resposta da IF, de modo a se obter as variáveis de defasagem descritas na Subseção 4.4.4. Para isso, são utilizadas expressões regulares buscando identificar segmentos textuais que possam corresponder a datas. Depois, é calculada a quantidade de caracteres desses campos, resultando nas variáveis de tamanho do texto, também descritas na aludida Subseção.

5.1.2 Tratamento dos textos

Uma vez compostos os atributos de data e de tamanho do texto, é realizado um tratamento de todos os campos textuais, inclusive das reclamações e respostas de eventuais demandas anteriores abertas pelo mesmo demandante contra a mesma entidade reclamada nos últimos 30 dias. Nessa fase, as seguintes transformações são realizadas:

1. Tokenização
2. Normalização Unicode
3. Conversão para o sistema de representação ASCII[1]
4. Transformação de letras maiúsculas em minúsculas
5. Substituição do texto `c/c` por `cc`
6. Substituição do texto `R$` por `reaisreais`
7. Substituição de endereços de rede por `urlurl`
8. Substituição de e-mails por `emailemail`
9. Substituição de datas por `datadata`
10. Substituição de números por `numnum`

[1] ASCII - Código Padrão Americano para o Intercâmbio de Informação

11. Substituição de nomes por `nomenome`
12. Substituição de termos abreviados ou coloquiais por termos formais
13. Remoção de pontuações
14. Remoção de *stopwords*

No passo 1, os campos de textos são tokenizados, isto é, quebrados em segmentos menores [70]. Os passos 2 e 3 buscam manter adequação a padrões internacionais, com o passo 4 estabelecendo uma representação de escrita unicameral (no caso, apenas minúsculas). Em seguida, nos passos 5 a 11, são utilizadas expressões regulares para identificar e substituir elementos do texto por categorias potencialmente relevantes para o aprendizado do modelo. Especificamente quanto à substituição tratada no passo 11, foi definida uma lista com os 760 nomes mais comuns observados nos dados do RDR. Analogamente, foi utilizado um dicionário com 15 abreviaturas e expressões coloquiais recorrentes para serem substituídas no passo 12 (ex.: *eh*, *vc*, *tá*). Enfim, os passos 13 e 14, respectivamente, removem pontuações e *stopwords* (termos considerados irrelevantes, como artigos, preposições, conjunções e outras palavras auxiliares que não agregam valor ao texto [31]). Para ilustrar esse tratamento, a Figura 5.1 apresenta as tranformações obtidas em um exemplo **fictício** de reclamação.

Reclamação original:
Bom dia,
Esta já eh a 4ª VEZ que eu faço uma reclamação deste banco desde 01/01/2021!!
Os pagamentos das competências 12/2018, 12/2019 e 12/2020 foram indevidos e ainda não foi depositado o estorno de R$1.000,00 na minha conta (c/c 1234-5)!
Sigo aguardando resposta.
Att.,
Fulano
Empresa ABCD
https://www.empresa.abcd.com.br/
fulano@mail.com

Reclamação tokenizada:
'bom', 'dia', 'numnum', 'vez', 'faco', 'reclamacao', 'deste', 'banco', 'desde', 'datadata', 'pagamentos', 'competencias', 'datadata', 'datadata', 'datadata', 'indevidos', 'ainda', 'depositado', 'estorno', 'reaisreais', 'numnum', 'conta', 'cc', 'numnum', 'sigo', 'aguardando', 'resposta', 'att', 'nomenome', 'empresa', 'abcd', 'urlurl', 'emailemail'

Figura 5.1: Exemplo de tratamento do texto (Fonte: elaborado pelo autor).

5.1.3 Vetorização dos textos e variáveis de similaridade

Após o tratamento do texto, é necessário transformá-lo em uma representação capaz de ser utilizada pelo algoritmo do modelo. Em abordagens tradicionais de classificação de documentos, essa conversão é geralmente feita por meio de vetorização, sendo o TF-IDF uma das técnicas mais utilizadas para esse fim [70, 71]. Como definido na Subseção 2.3.1, trata-se de uma matriz contendo a frequência de cada termo normalizada pelo inverso da sua ocorrência nos documentos [40], calculada por meio da Equação 2.1. Após experimentações com diferentes representações textuais, como o BOW e o N-gram, foi selecionado, para compôr a solução desenvolvida pelo BCB, o TF-IDF, uma vez que proporcionou melhores resultados para a categorização de interesse.

Assim, nesta fase, é calculado o TF-IDF para todos os campos de texto, considerando hiperparâmetros como `min_df` e `n_gram_range`, melhor abordados na Seção 5.2. Em seguida, com base nas representações vetoriais obtidas, é calculada a similaridade por cosseno das demandas atuais com as anteriores, conforme descrito na Subseção 4.4.4. Os vetores das reclamações e respostas passadas são então descartados, e os demais mantidos como variáveis do modelo. A título ilustrativo, a Figura 5.2 apresenta nuvens de palavras geradas a partir do TF-IDF - unigramas e bigramas - do texto das reclamações e das respostas. Por fim, uma matriz esparsa contendo todas as variáveis categóricas, numéricas e textuais é construída e passada para o algoritmo de ML.

(a) Texto das reclamações (b) Texto das respostas

Figura 5.2: Nuvens de palavras geradas com o TF-IDF (Fonte: elaborado pelo autor).

5.2 Modelagem

Uma vez pré-processadas, as entradas são passadas para um algoritmo efetuar a tarefa de classificação. Na solução desenvolvida pelo BCB, foi selecionado o LightGBM, visto ter apresentado desempenho superior aos demais modelos clássicos considerados, notadamente o NB, o SVM e o RF. Conforme descrito na Subseção 2.3.3, trata-se de algoritmo

de ML baseado em GBDT - técnica de *boosting* com conjuntos de árvores de decisão - com alto desempenho em termos de velocidade computacional e consumo de memória, e que tem obtido sucesso considerável em uma ampla gama de aplicações práticas [28], alcançando resultados de estado-da-arte em diversos *benchmarks* para tarefas de classificação [47, 29].

5.2.1 Treinamento e otimização do modelo

Assim como a maioria dos algoritmos de ML, o LightGBM contém hiperparâmetros que precisam ser ajustados com base no *dataset* selecionado para o treinamento do modelo. Esse processo de otimização geralmente conta com métodos como busca exaustiva, busca aleatória e otimização bayesiana [48]. Todavia, o processo realizado pelo BCB quando da construção do seu modelo foi baseado em experimentações, em detrimento da sua reproducibilidade nesta pesquisa. Destarte, no intuito de contornar essa limitação e proporcionar uma comparação mais justa entre os modelos, foi adotada uma estratégia de busca exaustiva com base em hiperparâmetros referenciados na literatura [29, 72, 73] e valores extraídos dos experimentos conduzidos pelo Banco Central.

De modo geral, os hiperparâmetros dos algoritmos baseados em GBDT podem ser agrupados em quatro categorias: (i) parâmetros relacionados à estrutura das árvores de decisão; (ii) parâmetros que afetam a velocidade do treinamento do modelo; (iii) parâmetros para melhorar o desempenho do modelo; e (iv) parâmetros para combater o sobreajuste do modelo (*overfitting*) [72, 73].

Considerando que no LightGBM o crescimento das árvores do *ensemble* é orientado pelas folhas (tornando-as tipicamente muito mais profundas do que as árvore orientadas à profundidade, para um número fixo de folhas), hiperparâmetros como o `num_leaves` - número máximo de folhas das árvores - e o `min_child_samples` - número mínimo de observações necessárias em uma folha - se tornam cruciais para controlar a complexidade dos estimadores base, podendo levar ao sobreajuste ou sub-ajuste do modelo. Por outro lado, o tempo total do treinamento pode aumentar consideravelmente à medida que mais nós são incluídos.

Uma estratégia comum para a obtenção de melhor desempenho envolve a definição de um grande conjunto de árvores com baixo aprendizado individual [28, 48]. Em outras palavras, valores elevados para `n_estimators` - número de estimadores para compor o *ensemble* - e reduzidos para `learning_rate` - taxa de aprendizagem no processo de *boosting*. Técnicas de GBDT envolvem a construção de conjuntos de árvores de forma iterativa, na qual cada estimador acrescido busca corrigir os erros dos demais. Essa abordagem, embora rápida e poderosa, está fortemente propensa ao sobreajuste do modelo.

Assim, ao definir valores baixos para o `learning_rate`, é possível controlar a velocidade de aprendizagem, porém aumentando o tempo gasto no treinamento.

O LightGBM possui ainda hiperparâmetros de regularização para controlar o efeito de sobreajuste, a exemplo dos termos de penalização L1 (Lasso) e L2 (Ridge), representados por `reg_alpha` e `reg_lambda`, e da rodada de parada precoce, referente ao `early_stopping_round`, na qual é estabelecido um limite de iterações sem melhoria no desempenho para interromper a aprendizagem, de modo a evitar o sobreajuste do modelo e, ao mesmo tempo, reduzir o tempo do treinamento.

Embora inúmeros outros hiperparâmetros do LightGBM possam ser ajustados durante o processo de otimização, a presente pesquisa se limitou àqueles aqui apresentados, com base em boas práticas sugeridas na literatura [29, 72, 73] e na documentação do LightGBM[2]. Foram também incluídos dois hiperparâmetros do TF-IDF utilizados na construção do modelo atual, mencionados na Seção 5.1, a saber: `min_df` (limite inferior para a frequência de documento dos termos) e `n_gram_range` (limites inferior e superior do intervalo de n valores para diferentes n-gramas de palavras a serem extraídos). Finalmente, diante do desbalanceamento dos dados, foi utilizada a configuração `is_unbalance` para definir pesos para as classes durante o treinamento.

Tabela 5.1: Hiperparâmetros Modelo BCB

Algoritmo	**Hiperparâmetro**	**Valores**
TF-IDF	`min_df`	100, 200
	`n_gram_range`	(1, 1), (1, 2)
LightGBM	`n_estimators`	500, 1500, 3000
	`num_leaves`	1500, 3000
	`min_child_samples`	200, 500
	`learning_rate`	0.001, 0.02
	`reg_alpha`	0.0, 0.01
	`reg_lambda`	0.0, 0.01
	`early_stopping_round`	15

Para cada hiperparâmetro selecionado, foram estabelecidos valores com base nas experimentações realizadas pelo BCB, conforme apresentado na Tabela 5.1, com os demais seguindo a configuração padrão do algoritmo de classificação do LightGBM empregado[3]. Foi então adotada a estratégia de busca exaustiva, que consiste na combinação dos valores pré-configurados - de modo a criar um conjunto de arranjos contendo todas combinações possíveis - e no treinamento de modelo para cada um dos arranjos [60].

O processo de otimização foi ainda conduzido utilizando técnica de validação cruzada com cinco dobras para o treinamento e avaliação dos modelos, no intuito de se obterem

[2]https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html
[3]https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html

resultados mais robustos para o estudo [4, 60]. Dessa forma, o *dataset* de treinamento foi particionado - de forma estratificada, visando preservar a proporção das classes - em cinco subconjuntos mutuamente exclusivos, com avaliações iterativas efetuadas considerando um subconjunto para validação e os demais para treinamento. As avaliações foram pautada na métrica de desempenho definida na Seção 5.3, sendo que, para cada arranjo de hiperparâmetros, considerou-se o desempenho médio obtido sobre as cinco dobras. Como resultado, o modelo com maior desempenho foi adotado como linha-de-base, passando a representar o classificador do Banco Central para fins de comparação na presente pesquisa, e sendo referenciado como **Modelo BCB**.

5.2.2 Variáveis tabulares

Adicionalmente, no intuito de melhor explorar a hipótese de pesquisa **H1** ("A utilização das variáveis tabulares levantadas pelo BCB em adição às variáveis de texto não contribui para a tarefa de classificação das demandas abertas pelos cidadãos"), foi reproduzido o mesmo modelo descrito na Subseção 5.2.1, porém removendo-se as variáveis categóricas e numéricas, isto é, passando para o classificador, como entrada, apenas o TF-IDF gerado a partir da reclamação do cidadão e da resposta da IF. Naturalmente, para fins de comparação, foi aplicado o mesmo processo de otimização, por meio de validação cruzada e com os mesmos hiperparâmetros e valores apresentados na Tabela 5.1, de modo a selecionar o modelo com melhor desempenho com a estratégia de busca exaustiva. O classificador obtido foi denominado **Modelo BCB$_{\textbf{notab}}$**.

5.3 Avaliação

Para avaliar os modelos apresentados na Seção 5.2, isto é, aferir sua capacidade de tomar decisões corretas de classificação, foi necessário, primeiro, definir uma métrica de desempenho com base no problema a ser solucionado. Neste estudo, conforme descrito na Seção 1.2, o interesse de negócio reside, primordialmente, em filtrar as reclamações abertas pelos clientes e usuários do SFN, no intuito de direcionar os esforços de análise e julgamento promovidos pelos servidores do Deati para os casos com maior possibilidade de serem procedentes, já que, atualmente, o BCB só consegue analisar cerca de 60% dessas demandas.

Do ponto de vista computacional, o problema em questão consiste em ranquear as instâncias de um *dataset* desbalanceado, com foco na classe minoritária (demandas procedentes), no intuito de selecionar as 60% com melhor *score* para serem avaliadas pelos servidores. Levando em consideração as características das métricas de desempenho apresentadas na Seção 2.5, aquela que melhor atendeu à finalidade descrita foi a Área sob a

curva Precisão-Revocação (**PRAUC**), visto se tratar de métrica de ranqueamento empregada precipuamente em tarefas de classificação binária envolvendo dados desbalanceados com interesse na classe positiva.

Pelos motivos expostos, o PRAUC foi a métrica de desempenho selecionada para a avaliação dos classificadores nos experimentos desta pesquisa, norteando, portanto, o processo de otimização dos modelos. Essa métrica foi também adotada na comparação entre o Modelo BCB (linha-de-base) e os classificadores propostos no Capítulo 6, no intuito de selecionar aquele com melhor desempenho. Por fim, o PRAUC foi calculado sobre as predições do classificador final - retreinado com todos os dados de treinamento, reservando-se 5% para o `early_stopping` - no *dataset* de teste, no intuito de estimar sua habilidade de generalização na tarefa de interesse.

Ademais, foi ainda proposta, pela área de negócio do BCB, outra medida, denominada "**revocação por faixas**" na presente dissertação, para estimar o impacto do classificador final nas atividades de tratamento das reclamações. O cálculo dessa medida envolve, primeiro, a ordenação decrescente das instâncias com base nos *scores* preditos pelo modelo. Em seguida, são definidos *thresholds* em pontos específicos, de modo a agregar proporções de instâncias equivalentes a percentuais pré-estabelecidos, dando origem às chamadas "faixas". Os percentuais se iniciam em 10%, e aumentam incrementalmente na mesma proporção, até que dez faixas tenham sido definidas. Para cada faixa, os *scores* do modelo são, então, convertidos em predições - sendo "procedentes" (1) as observações com maior pontuação, isto é, acima do *threshold* definido para a faixa, e "improcedente" (0) as demais - de modo a calcular a revocação (conforme Equação 2.14) frente às classes verdadeiras.

Tabela 5.2: Exemplo de revocação na faixa 60%

Passo 1		**Passo 2**			**Classe real**	**Situação**
i	**Score**	**i**	**Score**	**Predição**		
1	0,87	↑ 10	0,98	1	1	TP
2	0,02	↑ 9	0,92	1	0	FP
3	0,74	↓ 1	0,87	1	0	FP
4	0,33	↓ 3	0,74	1	1	TP
5	0,13	↑ 6	0,41	1	0	FP
6	0,41	↓ 4	0,33	1	1	TP
				threshold		
7	0,29	7	0,29	0	1	FN
8	0,01	↓ 5	0,13	0	0	TN
9	0,92	↓ 2	0,02	0	0	TN
10	0,98	↓ 8	0,01	0	0	TN

Para facilitar a compreensão dessa medida, a Tabela 5.2 e a Equação 5.1 apresentam um caso hipótetico de cálculo da revocação para a faixa de 60% em um conjunto i de 10

instâncias, com a Tabela 5.3 sumarizando os resultados para todas as 10 faixas. O objetivo dessa medida é estimar o ganho obtido com a utilização das predições do classificador para direcionar as atividades realizadas pelos servidores do BCB. Assim, no exemplo apresentado, caso fossem analisadas apenas 60% das demandas, com base no *score* gerado pelo modelo, seria possível abranger 3/4 do total daquelas de fato procedentes, isto é, nas quais houve indício de inobservância de obrigações regulatórias pelas entidades supervisionadas.

$$\text{Revocação na faixa } 60\% = \frac{TP}{TP + FN} = \frac{3}{3+1} = 75\% \tag{5.1}$$

Tabela 5.3: Exemplo de revocação por faixa

Faixa	100%:	100%
Faixa	90%:	100%
Faixa	80%:	100%
Faixa	70%:	100%
Faixa	60%:	75%
Faixa	50%:	50%
Faixa	40%:	50%
Faixa	30%:	25%
Faixa	20%:	25%
Faixa	10%:	25%

Capítulo 6

Desenvolvimento do Modelo Proposto

Neste Capítulo é apresentado o modelo proposto para a tarefa de categorização das reclamações abertas pelos clientes e usuários do SFN, desenvolvido com base nas hipóteses de pesquisa **H1** - "A utilização das variáveis tabulares levantadas pelo BCB em adição às variáveis de texto não contribui para a tarefa de classificação das demandas abertas pelos cidadãos" - e **H2** - "A utilização do *Bidirectional Encoder Representations from Transformers* (BERT) permite a construção de um classificador para as demandas abertas pelos cidadãos com melhor desempenho do que o desenvolvido pelo BCB"-, definidas na Seção 1.3, e tendo como referência o Modelo BCB reproduzido no Capítulo 5.

6.1 Abordagem escolhida

O classificador do Banco Central foi desenvolvido adotando-se método tradicional de categorização de texto. Nessa solução, foi utilizado o TF-IDF, representação que, embora propicie a vetorização textual, sendo amplamente empregada na literatura, apresenta duas grandes limitações. A primeira é referente à ordem sequencial das palavras no texto. Como apenas a frequência ponderada dos termos é armazenada, a posição em que esses aparecem no documento acaba sendo descartada. Ainda que a adoção de bigramas ou trigramas - que, no entanto, resulta no aumento da dimensionalidade da representação - possa recuperar parte dessa informação sequencial, o contexto, de modo geral, acaba sendo perdido após a vetorização [40].

A outra limitação do TF-IDF é que essa técnica não leva em consideração os valores semânticos das palavras, sendo cada termo assumido como independente dos demais. Consequentemente, ainda que existam relacionamentos entre palavras, como sinônimos e antônimos, essas informações são perdidas com a aludida representação textual [10]. A

limitação se estende ainda ao fenômeno de polissemia, em que duas palavras idênticas podem ter significados diferentes dependendo do contexto no qual se encontram [51].

No entanto, com os grandes avanços ocorridos na área de NLP, como resultado da migração dos métodos tradicionais para abordagens baseadas em DL, tanto as formas de representação dos textos quanto as maneiras de processá-los evoluiram consideravelmente. Assim, técnicas capazes não só de apresentar os textos em formatos númericos, mas também de capturar informações semânticas e sintáticas das palavras foram desenvolvidas [42, 7]. Dentre elas, a mais popular tem sido a representação por *embeddings* contextuais, uma vez que - diferentemente dos *word embeddings* tradicionais - conseguem desambiguar palavras polissêmicas com base no seu contexto [11].

A respeito, estratégias envolvendo o ajuste fino de robustos LMs pré-treinados em grandes conjuntos de documentos para gerar representações de *embeddings* contextuais têm obtido desempenhos que alcançaram o estado-da-arte em diversas tarefas de NLP. Nesse sentido, o BERT, baseado na arquitetura de transformadores, designado para pré-treinar representações bidirecionais profundas a partir de texto não rotulados, condicionando conjuntamente os contextos à esquerda e à direita, viabilizou a utilização dessas estratégias em diversos domínios e línguas [52, 12, 32].

Adicionalmente, enquanto abordagens tradicionais costumam requerer técnicas mais robustas de pré-processamento, buscando trabalhar os dados e refinar as variáveis do modelo, o que demanda conhecimento especializado do negócio, abordagens de DL, por sua vez, propõem um pré-processamento mínimo das entradas, de modo que o modelo seja treinado de ponta-a-ponta, aprendendo, portanto, a extrair as variáveis mais relevantes para a tarefa de interesse automaticamente a partir dos dados [74, 14].

Diante do exposto, foi escolhida, para o classificador proposto nesta pesquisa, a abordagem baseada em DL, especificamente solução com *embeddings* de contexto gerados a partir do BERTimbau - LM proposto em [13] obtido ao pré-treinar o BERT em um grande corpus em português, denominado brWaC - no intuito de se obter representações mais ricas dos textos da reclamação do cidadão e resposta da IF, em linha com a segunda hipótese de pesquisa definida (**H2**). Adicionalmente, no âmbito da hipótese **H1**, foram, em princípio, descartadas as variáveis tabulares estabelecidas pelo BCB, de modo que os atributos fossem definidos pelo próprio modelo a partir das variáveis de texto.

6.2 Pré-processamento das entradas

Embora modelos de DL usualmente dispensem a engenharia de variáveis característica de abordagens tradicionais, é ainda necessário certo tratamento e processamento dos dados textuais visando torná-los adequados à aplicação do algoritmo selecionado, no caso o

BERT. Assim sendo, foram descritos, nesta Seção, os procedimentos adotados para o pré-processamento das entradas do modelo.

6.2.1 Tokenização em pedaços de palavras

O BERT requer, como entrada, representações textuais tokenizadas a nível de pedaços de palavras (*wordpieces*), nas quais o texto é primeiro dividido em palavras e essas, em seguida, são segmentadas em unidades de sub-palavras com base em um vocabulário definido [12]. Para o BERTimbau [13], os autores propuseram um vocabulário em português bicameral com 30 mil sub-palavras, geradas a partir de 2 milhões de sentenças aleatórias obtidas de artigos em português do Wikipedia. Nesta pesquisa, contudo, os campos de textos foram convertidos integralmente em minúsculas, visto que diversas passagens em caixa-alta foram decorrentes dos processos de captação e transcrição dos dados, não necessariamente representando aspectos de interesse da língua. A Figura 6.1 ilustra o mesmo exemplo **fictício** de reclamação anteriormente apresentado na Seção 5.1.2, porém agora considerando a tokenização supracitada.

Reclamação original:
Bom dia,
Esta já eh a 4ª VEZ que eu faço uma reclamação deste banco desde 01/01/2021!!
Os pagamentos das competências 12/2018, 12/2019 e 12/2020 foram indevidos e ainda não foi depositado o estorno de R$1.000,00 na minha conta (c/c 1234-5)!
Sigo aguardando resposta.
Att.,
Fulano
Empresa ABCD
https://www.empresa.abcd.com.br/
fulano@mail.com

Reclamação tokenizada:
'bom', 'dia', ',', 'esta', 'ja', 'e', '##h', 'a', '4', '##a', 'vez', 'que', 'eu', 'fa', '##co', 'uma', 'reclama', '##ca', '##o', 'deste', 'banco', 'desde', '01', '/', '01', '/', '2021', '!', '!', 'os', 'pagamentos', 'das', 'compet', '##encia', '##s', '12', '/', '2018', ',', '12', '/', '2019', 'e', '12', '/', '2020', 'foram', 'inde', '##vidos', 'e', 'ainda', 'na', '##o', 'foi', 'depos', '##itado', 'o', 'esto', '##r', '##no', 'de', 'r', '$', '1', '.', '000', ',', '00', 'na', 'minha', 'conta', '(', 'c', '/', 'c', '12', '##34', '-', '5', ')', '!', 'sig', '##o', 'aguarda', '##ndo', 'resposta', '.', 'at', '##t', '.', ',', 'fu', '##lan', '##o', 'empresa', 'ab', '##c', '##d', 'http', '##s', ':', '/', '/', 'w', '##ww', '.', 'empresa', '.', 'ab', '##c', '##d', '.', 'com', '.', 'b', '##r', '/', 'fu', '##lan', '##o', '@', 'ma', '##il', '.', 'com'

Figura 6.1: Exemplo da tokenização para o BERT (Fonte: elaborado pelo autor).

Tabela 6.1: Exemplo de entrada do BERT.

Token	[CLS]	reclama	##ca	##o	[SEP]	resposta	i	##f	[SEP]
Token id	101	15158	304	22280	102	4299	254	22294	102
Segmento	A	A	A	A	A	B	B	B	B
Posição	0	1	2	3	4	5	6	7	8
Emb. Tok.	E_V^{101}	E_V^{15158}	E_V^{304}	E_V^{22280}	E_V^{102}	E_V^{4299}	E_V^{254}	E_V^{22294}	E_V^{102}
Emb. Seg.	E_{seg}^{A}	E_{seg}^{A}	E_{seg}^{A}	E_{seg}^{A}	E_{seg}^{A}	E_{seg}^{B}	E_{seg}^{B}	E_{seg}^{B}	E_{seg}^{B}
Emb. Pos.	E_{pos}^{0}	E_{pos}^{1}	E_{pos}^{2}	E_{pos}^{3}	E_{pos}^{4}	E_{pos}^{5}	E_{pos}^{6}	E_{pos}^{7}	E_{pos}^{8}

6.2.2 *Embeddings* de entrada

Após o processo de tokenização, são gerados os três *embeddings* de entrada do BERT, conforme discorrido na Subseção 2.4.4, quais sejam: os de token, os de segmento e os de posição, representados, respectivamente, pelas matrizes $\mathbf{E}_V^{x_i}$, $\mathbf{E}_{seg}^{A|B}$ e $\mathbf{E}_{pos}^{i}$. Como o modelo foi designado para desambiguar duas sentenças dentro de uma mesma sequência, optou-se por concatenar os tokens da reclamação do cidadão (A) e os da resposta da IF (B), com a devida dissociação ocorrendo na representação dos *embeddings* em $\mathbf{E}_{seg}^{A|B}$. A Figura 6.2 - adaptada de [12] - e a Tabela 6.1 - adaptada de [14] - ilustram um exemplo de representação da entrada do BERT para a solução proposta.

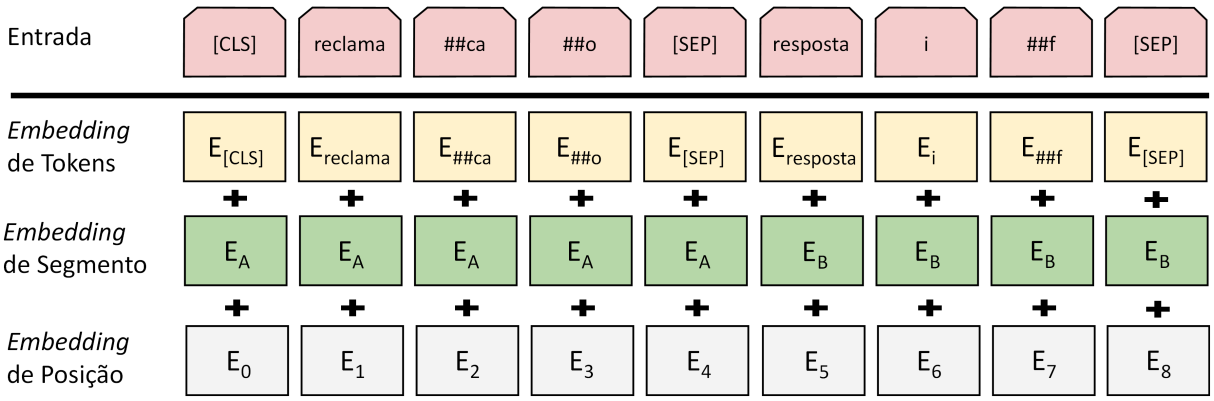


Figura 6.2: Exemplo de representação das entradas (Fonte: adaptado de [12]).

6.2.3 Tamanho máximo da sequência de entrada

Como mencionado na Subseção 2.4.4, a arquitetura do BERT consiste em um conjunto de transformadores bidirecionais empilhados uns sobre os outros. Desta forma, mecanismos de auto-atenção (*self-attention*) - onde cada token é processado para gerar um *embedding* de contexto usando os demais tokens - são utilizados, apresentando uma complexidade quadrática que acaba se tornando um limitador em termos de custo computacional à

medida em que o tamanho das entradas aumenta [56]. Consequentemente, o comprimento máximo de sequência que o modelo pode processar se resume a apenas 512 tokens [11, 74].

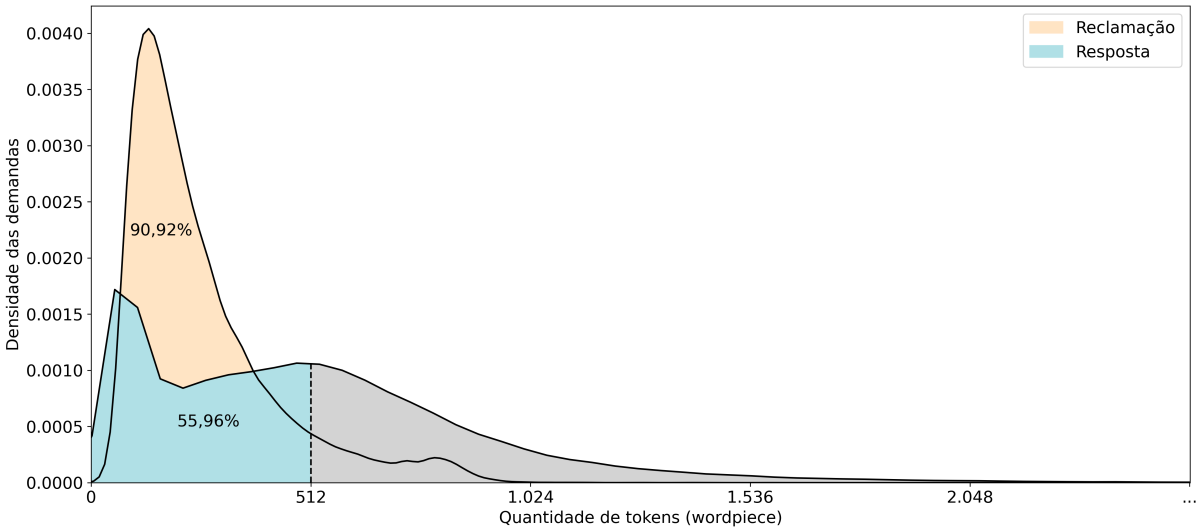


Figura 6.3: Quantidade de tokens: reclamação e resposta (Fonte: elaborado pelo autor).

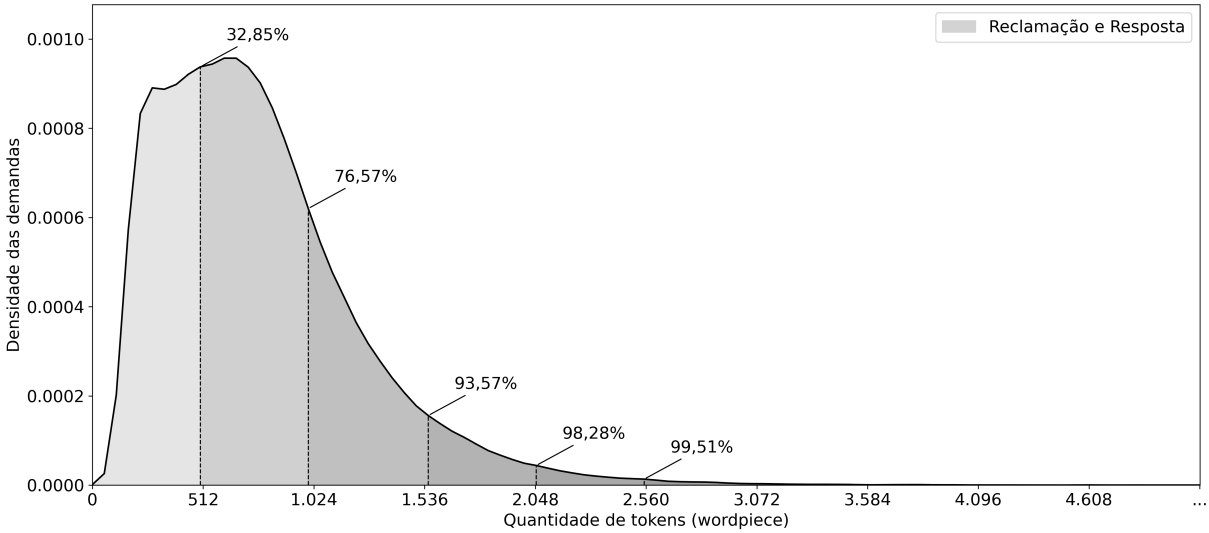


Figura 6.4: Quantidade de tokens: campos concatenados (Fonte: elaborado pelo autor).

Diante dessa limitação, foi levantada a densidade de demandas por quantidade de tokens (a nível de pedaços de palavras) no *dataset* de experimento, como consta da Figura 6.3. Verificou-se que aproximadamente 91% das reclamações e 56% das respostas apresentam comprimento dentro do limite de 512 tokens. Consequentemente, nem um 1/3 das demandas, em termos de volume total de texto, poderia ser processado pelo

BERT conjuntamente, como observado na Figura 6.4, o que prejudicaria a proposta de concatenação dos tokens mencionada na Subseção 6.2.2.

6.2.4 Representação hierárquica

Como documentos podem ser considerados cadeias sequenciais de sentenças, sendo estas, por sua vez, constituídas de sequências de palavras, diversos estudos na literatura têm utilizado, com sucesso, modelos hierárquicos para tarefas de processamento de linguagem natural com textos longos, buscando explorar a estrutura palavras/sentenças/documento. Assim, no primeiro nível da hierarquia, a sequência de palavras é processada para gerar representações de sentenças (*sentence embedding*). Já no segundo nível, a sequência de sentenças obtida é processada para gerar uma representação do documento (*document embedding*) [15, 56, 37].

Uma das saídas do BERT é o estado oculto final do primeiro token da sequência de entrada (`[CLS]`), cuja finalidade é produzir uma representação codificada agregada de toda a sequência [12]. Dessa forma, é possível quebrar entradas extensas em segmentos menores de tokens (referenciados como *chunks* neste trabalho), e passá-los ao BERT para que cada *chunk* seja representado por um *embedding* de contexto a nível de sentença, em outras palavras, um *sentence embedding*, reproduzindo, portanto, o primeiro processo do modelo hierárquico retromencionado. Em seguida, a representação de saída para cada *chunk* seria propagada através de uma rede neural superficial [15, 56] ou de outro transformador [15] de modo a gerar um *document embedding* para a classificação desejada. Na presente pesquisa, optou-se pela utilização de uma *Long short-term memory* (LSTM) no segundo nível do modelo hierárquico, conforme descrito na Seção 6.3.

6.2.5 Segmentação das entradas em *chunks*

Para quebrar as sequências de entrada em pedaços menores, foi, primeiramente, fixado o tamanho de cada *chunk* como sendo o limite de entrada do BERT, qual seja 512 tokens. Adicionalmente, foram estabelecidas janelas de sobreposição (*overlapping windows*) com recuos (*stride*) pré-definidos, buscando resgatar, em cada *chunk*, parte do contexto perdido [12, 75]. A Figura 6.5 ilustra esse processo de segmentação a partir das sequências tokenizadas e concatenadas dos textos da reclamação e da resposta. Observa-se que, para que cada *chunk* tenha o mesmo tamanho, além da segmentação pode ser necessária a aplicação de técnicas de preenchimento (*padding*), que envolvem a adição de tokens especiais (`[PAD]`).

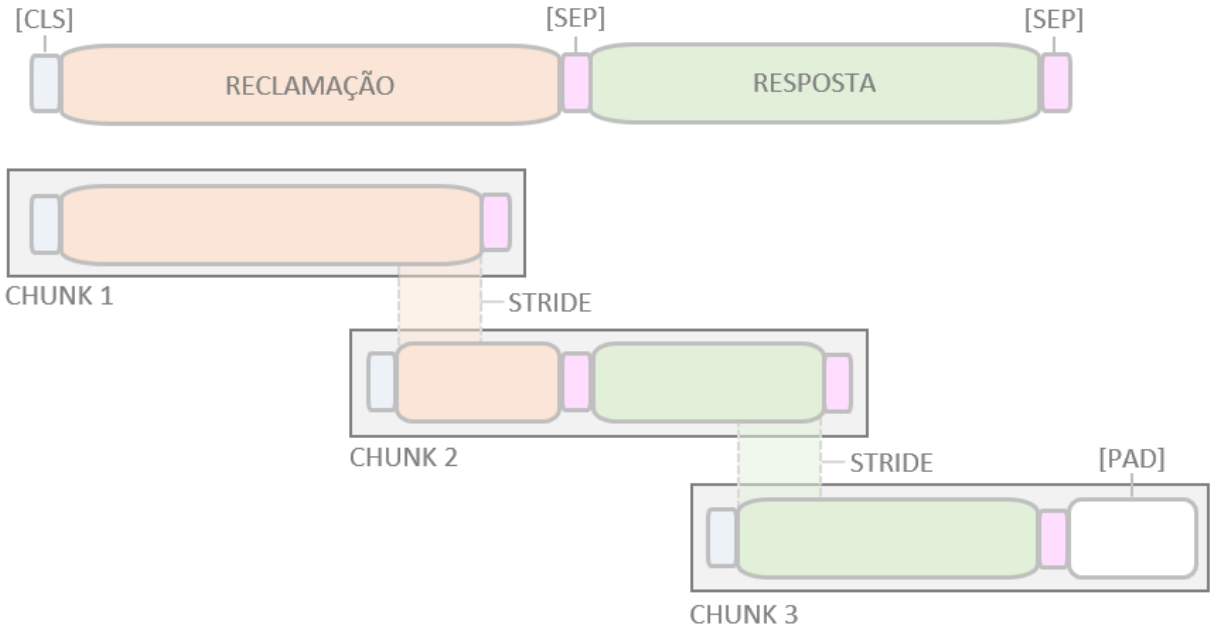


Figura 6.5: Segmentação dos *chunks* (Fonte: adaptado de [75]).

6.2.6 Representação para o treinamento do modelo

A inserção de janelas de sobreposição e a aplicação de técnicas de *padding* resultam no aumento da quantidade de tokens a serem processados quando da geração dos *embedding* de contexto. Embora, na estrutura hierárquica proposta, os *chunks* sejam passados individualmente para o BERT, o modelo final é treinado conjuntamente, de modo que cada *batch* deve receber, concomitantemente, todos os *chunks* segmentados dos tokens da reclamação e da resposta. No entanto, constatou-se, a título exemplificativo da limitação técnica exposta na Seção 4.2, que a memória da GPU da máquina disponibilizada pelo BCB não comporta, para o treinamento, entradas de grandes dimensões, sendo, portanto, necessário escolher entre *batches* de maior tamanho ou descarte de *chunks* nas entradas.

Assim sendo, foram levantadas as frequências de demanda por quantidade de *chunks* das sequências, considerando um *stride* de 128 tokens, definido com base na literatura [14, 15, 75]. A distribuição obtida, bem como sua frequência acumulada relativa, se encontra representada na Figura 6.6. Observou-se que aproximadamente 98% das demandas poderiam ser integralmente representadas por entradas com 5 *chunks*, e que o seu processamento, por ocasião do treinamento do modelo, seria viável adotando-se *batches* de tamanho 2. Logo, como a forma de representação textual é um dos principais objetos de estudo da presente pesquisa, optou-se por manter o tamanho dos *batches* reduzido em prol da manutenção de maiores volumes de dados textuais nas entradas.

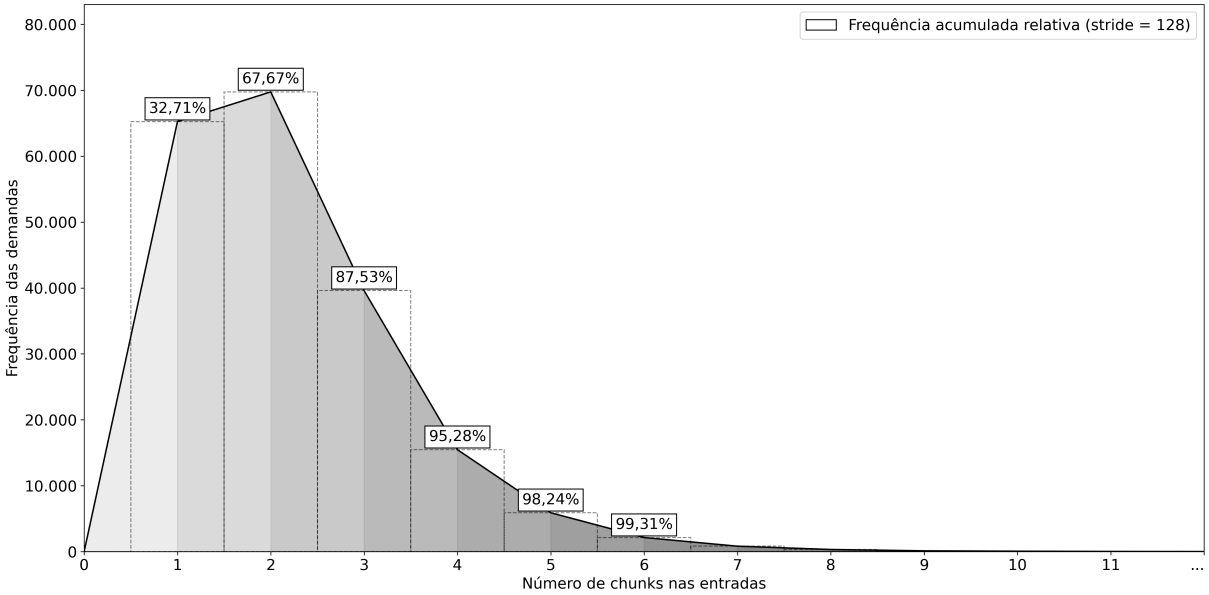


Figura 6.6: Quantidade de token: stride = 128 (Fonte: elaborado pelo autor).

6.2.7 Procedimentos finais para o pré-processamento

Como resultado dos experimentos, foi definido, para o modelo proposto, um pré-processamento constituído por 4 etapas. Na primeira, os campos de texto são, individualmente, minusculizados e tokenizados a nível de pedaços de palavras, considerando o vocabulário definido para o BERTimbau. Na segunda etapa, as sequências de tokens das reclamações e das respostas são concatenadas, formando um *embedding* único. Ao mesmo tempo, são definidos os *embeddings* de segmento e de posição para dissociar e sequenciar os tokens concatenados.

Na terceira etapa, as representações obtidas são segmentadas em *chunks* de 512 elementos, considerando janelas de sobreposição com *strides* de 128 tokens e aplicando-se técnicas de preenchimento. Na quarta e última etapa, é verificado o tamanho das entradas. Caso uma sequência tenha sido segmentada em menos de cinco *chunks*, outro *padding* é efetuado para completar as lacunas. Em contrapartida, entradas com mais de cinco pedaços têm seus excedentes descartados. Ao final do pré-processamento, cada demanda foi representada por três tensores únicos compreendendo os cinco *chunks* obtidos a partir dos *embeddings* de token, segmento e posição.

Esses tensores foram passados para um modelo hierárquico no qual, primeiramente, os *chunks* são transformados em *embeddings* contextuais a nível de sentença, gerados pelo BERT, e, em seguida, um *embedding* de documento é obtido a partir do processamento dessas saídas por uma LSTM. Essa representação final é então utilizada para a tarefa de classificação das demandas.

6.3 Modelagem

A hipótese de pesquisa **H2** foi levantada buscando representações mais enriquecidas para as variáveis de texto, notadamente os *embeddings* contextualizados. Conforme descrito na Subseção 6.1, de modo a viabilizar o *fine-tuning* em diversas tarefas distintas, o BERT foi designado para desambiguar diferentes sentenças passadas cojuntamente na sequência de entrada. Assim sendo, optou-se por concatenar os textos - tokenizados - da resposta e da reclamação em um único *embedding*, buscando melhores resultados para as representações contextuais. Todavia, dada a limitação de tamanho máximo das sequências imposta pelo mecanismo de auto-atenção dos transformadores, não foi possível compreender todo o conteúdo da demanda em uma mesma entrada. Consequentemente, tendo como referência estudos recentes observados na literatura [15, 56, 37], foi proposta uma solução baseada na abordagem de modelagem hierárquica, conforme estrutura apresentada na Figura 6.7.

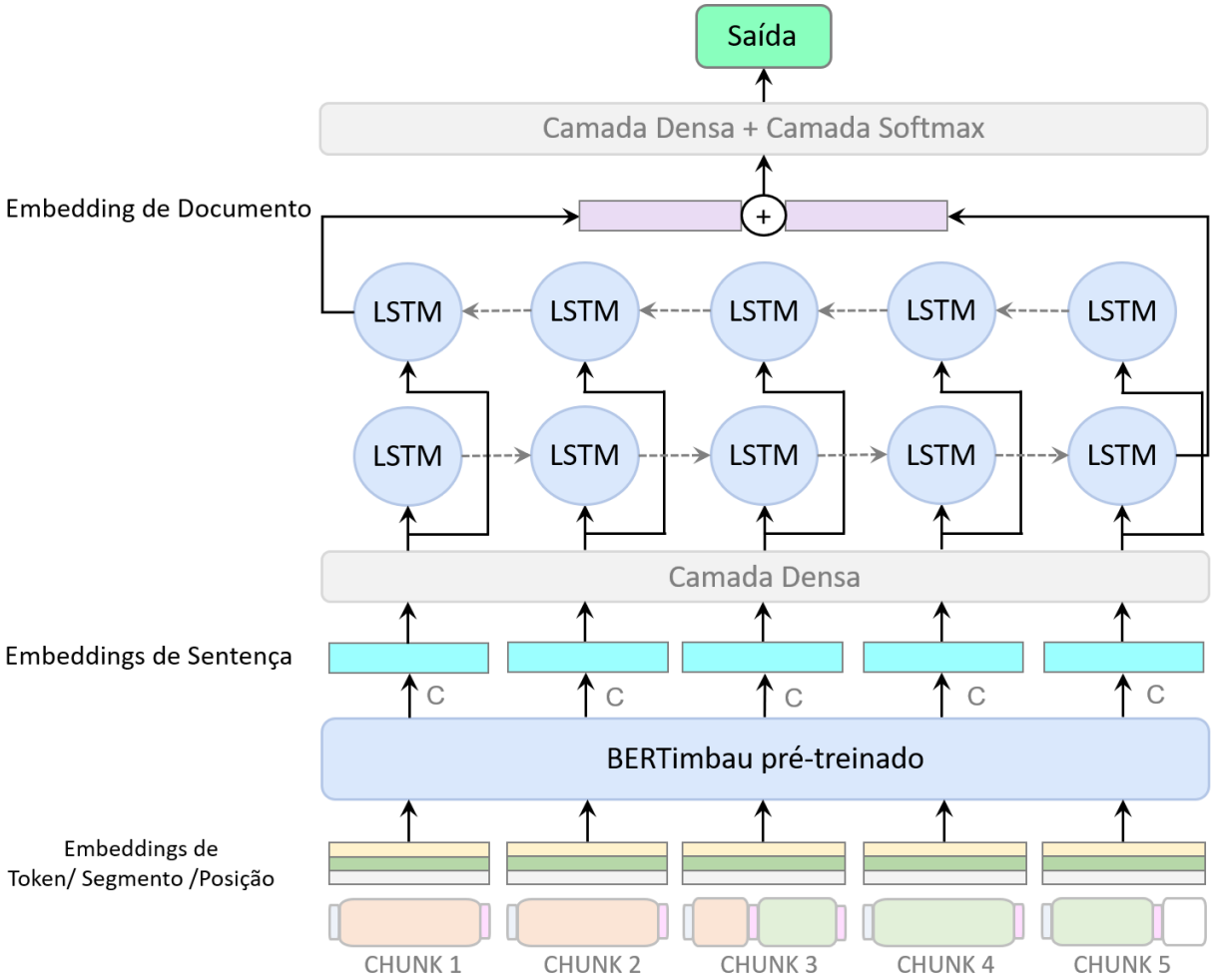


Figura 6.7: Estrutura do Modelo Proposto (Fonte: elaborado pelo autor).

6.3.1 Estrutura do modelo hierárquico

Para a estrutura proposta, cada demanda deve primeiro ser pré-processada, conforme descrito na Seção 6.2, de modo a se obter três *embeddings* (tokens, segmento e posição) com cinco *chunks* de 512 tokens. Em seguida, no primeiro nível da hierarquia, o BERT recebe as entradas e retorna, para cada uma, o estado oculto final do primeiro token (`C`), que contém uma representação agregada de todo o *chunk*, a título de *embedding* de contexto em nível de sentença (referida como "Embedding de Sentença" na Figura 6.7). Essas saídas passam por uma camada densa antes de prosseguirem para a etapa seguinte.

No segundo nível hierárquico, foi definida uma LSTM bidirecional (BiLSTM), comumente empregada em tarefas de classificação de texto, e que tem se mostrado bem-sucedida na representação de linguagem comum, dada a sua capacidade de capturar ordens sequenciais [57, 58]. Nessa etapa, cada *embedding* de sentença é processado, sendo retornados os estados ocultos finais nas camadas de sentido direto (*forward*) e inverso (*backward*), concatenados, constituindo uma representação da demanda a nível de documento (referida como "Embedding de Documento" na Figura 6.7). Por fim, duas camadas totalmente conectadas são estabelecidas para se obter as previsões do classificador.

6.3.2 Arquitetura do modelo

O modelo BERT utilizado foi o BERTimbau, proposto em [13], tratando-se de LM pré-treinado no corpus brWaC, a maior coleção aberta em português até o momento [14], conforme apresentado na Seção 3.1. No trabalho original, os autores efetuaram o treinamento com dois tamanhos diferentes: *Base* (12 camadas, dimensão oculta de 768, 12 cabeças de atenção e 110 milhões de parâmetros) e *Large* (24 camadas, dimensão oculta de 1024, 16 cabeças de atenção e 330 milhões de parâmetros). Contudo, dada a limitação de memória da GPU disponível para esta pesquisa, foi utilizado, nos experimentos, apenas o BERTimbau *Base*.

Na saída do BERT, previamente ao processamento dos *embeddings* de sentença pela LSTM, foi inserida uma camada densa intermediária com tangente hiperbólica (*tanh*) como função de ativação e inicializador normal Xavier para os pesos, visando auxiliar na regulação dos valores que fluem pela rede e permitir maior controle sobre o aprendizado do modelo. Nessa camada totalmente conectada, foram mantidas, na saída, as mesmas dimensões dos *embeddings* retornados, sendo 768 unidades no BERTimbau Base.

A arquitetura adotada para o segundo nível hierárquico do modelo foi baseada nos experimentos realizados em [57] e [58]. Consequentemente, foi construída uma BiLSTM com duas camadas de 256 unidades para retornar os *embeddings* de documento, seguida

de duas camadas totalmente conectadas com ativações ReLU (64 unidades) e *softmax* (2 unidades) para obter as predições finais do modelo.

6.3.3 Treinamento e otimização do modelo

O treinamento do modelo foi realizado de forma conjunta, adotando uma abordagem de ponta-a-ponta (E2E), na qual o aprendizado baseado em gradiente descendente é aplicado no sistema como um todo [76], abrangendo, concomitantemente, tanto os parâmetros do BERTimbau (inicializados com a configuração do pré-treino) quanto os da LSTM e das demais camadas ocultas. Em [12], os autores recomendam, após ajustes-fino do BERT para diferentes tarefas de NLP, a utilização dos mesmos hiperparâmetros do pré-treino, com exceções pontuais, para as quais foram sugeridos valores específicos, a saber: tamanho do *batch* (`batch_size`): 16, 32; taxa de aprendizagem (`learning_rate`): 5e^{-5}, 3e^{-5} e 2e^{-5}; e número de épocas (`num_train_epochs`): 2, 3 e 4. Assinalam, ainda, que *datasets* com mais de 100 mil observações rotuladas costumam ser menos sensitivos à escolha dos hiperparâmetros.

Outrossim, para o ajuste-fino do modelo hierárquico, foi adotada a mesma configuração de pré-treino do BERTimbau - com um otimizador AdamW com $\beta_1 = 0{,}9$ e $\beta_2 = 0,999$ e decaimento de peso L2 de 0,01 [14] - variando-se apenas o `learning_rate` e o `num_train_epochs`, de acordo com os valores sugeridos em [12], e o `batch_size`, sendo fixado em 2 dadas as limitações técnicas expostas na Subseção 6.2.6. Adicionalmente, os experimentos foram realizados considerando representações obtidas com *stride* de 128 tokens, com referência em [14, 15, 75]. A Tabela 6.2 sumariza os hiperparâmetros usados na otimização do modelo. Ademais, considerando que os dados são desmbalanceados, a função de perda de entropia cruzada foi customizada, visando estabelecer pesos para as classes durante o treinamento.

Tabela 6.2: Hiperparâmetros Modelo Proposto

Etapa	**Hiperparâmetro**	**Valores**
Pré-processamento	`stride`	128
Treinamento	`learning_rate`	$5e^{-5}, 3e^{-5}, 2e^{-5}$
	`num_train_epochs`	2, 3, 4
	`batch_size`	2

Enfim, analogamente ao procedimento descrito na Subseção 5.2.1, o processo de otimização do modelo contou com validação cruzada com cinco dobras particionadas estratificadamente, adotando-se a estratégia de busca exaustiva para selecionar o melhor arranjo de hiperparâmetros com base na métrica de desempenho definida. Dessa forma,

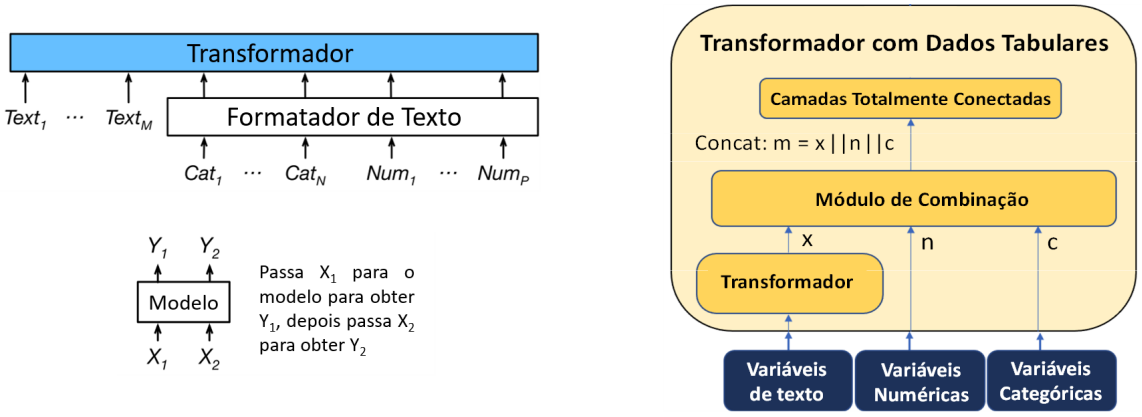
foi selecionado, ao final, o modelo com maior desempenho médio, denominado **Modelo Proposto** nesta dissertação.

6.3.4 Variáveis tabulares

Diante do contexto abordado na hipótese de pesquisa **H1**, referente à importância das variáveis tabulares para o desempenho do modelo, buscou-se estender os experimentos realizados para uma abordagem multimodal, de modo a combinar os *embeddings* de documento gerados pelo modelo hierárquico proposto com os atributos categóricos e numéricos definidos pelo BCB, descritos nas Subseções 4.4.3 e 4.4.4, respectivamente.

Embora o uso de modelos tabulares junto de arquiteturas de texto baseadas em transformadores tenha recebido pouca atenção pela comunidade científica [3], nos últimos anos, estudos pontuais, como os apresentados em [1] e [2], foram conduzidos buscando investigar como o paradigma de modelos de DL de ponta-a-ponta com arquitetura de transformadores poderia ser alavancado para abarcar entradas simultâneas de texto e dados tabulares, com diversas estratégias sendo propostas para adaptar as redes de transformadores de modo a operar concomitantemente em entradas de ambas as modalidades.

Para esta pesquisa, foram selecionadas duas estratégias de menor complexidade, quais sejam a *Unimodal/All-Text* e a *Concat*, representadas na Figura 6.8, uma vez que poderiam ser implementadas sem demandar grandes alterações na estrutura do modelo hierárquico definida anteriormente. Assim sendo, foram construídas duas novas soluções para compor os modelos propostos nos experimentos conduzidos neste estudo, descritas a seguir.



(a) *Unimodal / All-Text.* Converte valores numéricos e categóricos em *tokens* de texto adicionais.

(b) *Concat.* Concatena as variáveis tabulares com as variáveis de texto processadas pelo transformados.

Figura 6.8: Arquiteturas multimodais sugeridas (Fonte: adaptado de [1][2]).

Estratégia *Unimodal/All-Text*

A estratégia possivelmente mais simples para a combinação multimodal abordada envolve a conversão das variáves numéricas e categóricas em texto, de modo a comporem a entrada do transformador, como mostra a Figura 6.8 (a), sendo chamada de *Unimodal* em [1] e *All-Text* em [2] e [3]. Embora referida como "grosseira" em [2], essa estratégia foi a que obteve melhor desempenho em tarefa de classificação binária nos experimentos realizados por [1], todavia com ressalva dos autores de que, no *dataset* selecionado, as variáveis textuais abordavam informações de grande relevância para a tarefa em questão, com pouco ganho promovido pela inclusão de variáveis tabulares adicionais.

No presente estudo, a aplicação dessa estratégia envolveu, basicamente, transformar as 16 variáveis tabulares definidas pelo BCB em dados textuais. No entanto, como o tamanho das entradas tem impacto no treinamento do modelo, pelos motivos expostos na Seção 6.2, buscou-se manter sequências curtas de texto sem perder conteúdos relevantes. Para isso, os atributos foram, primeiro, discretizados e combinados com base em julgamento de negócio e na sua distribuição no *dataset* de experimentos. Em seguida, essas "novas variáveis" foram agrupadas e convertidas em sentenças curtas, e inseridas no início do texto da reclamação do cidadão e da resposta da IF - como ilustra o exemplo da Tabela 6.3 -, seguindo conjuntamente para a etapa de pré-processamento descrita na Seção 6.2, com subseguinte entrada no modelo hierárquico a partir do BERT. Os domínios definidos para as transformações efetuadas estão apresentados no Apêndice A.2. Como resultado, foi possível minimizar a perda de conteúdo das entradas, com a frequência relativa de demandas sem descarte de texto passando de 98,24% (Figura 6.6) para 97,85%.

Estratégia *Concat*

A outra estratégia - também de baixa complexidade - se refere à concatenação dos *embeddings* retornados pelo transformador diretamente com as variáveis numéricas e categóricas em uma camada de combinação multimodal, conforme ilustrado na Figura 6.8 (b). Essa abordagem, denominada *Concat* em [1], permite representações segregadas das variáveis tabulares, sem se sujeitarem aos mecanismos de atenção existentes na arquitetura de transformadores, tendo apresentado resultados satisfatórios nos experimentos realizados pelos autores em [1].

Para esta pesquisa, as variáveis categóricas e numéricas retromencionadas foram concatenadas diretamente nos *embeddings* de documento retornados pela BiLSTM, passando pelas camadas densas finais para gerar as predições do classificador. Contudo, para que os modelos convergissem mais rapidamente durante o treinamento, os atributos tabulares foram primeiro normalizados. As variáveis numéricas passaram por um estimador

Tabela 6.3: Exemplo de transformação unimodal

Variável Tabular				
Tipo	**Descrição**			**Valor**
Categórica	Demandas Anteriores			Sim
	Protocolos Abertos			Sim
	Segmento Instituição Financeira			Banco Múltiplo
Numérica	Total de reclamações (ano)			3
	Reclamações procedentes (ano)			1
	Reclamações improcedentes (ano)			2
	Defasagem de data	Reclamação	Mín	0
			Máx	0
		Resposta	Mín	-135
			Máx	9
		Global	Mín	-135
			Máx	9
	Tamanho das variáveis textuais	Reclamação		1021
		Resposta		2694
	Similaridade demanda anterior	Reclamação		0,1947
		Resposta		0,7109

Conversão das variáveis tabulares em texto	
Entrada	**Sentença gerada**
Reclamação	"Reclamação longa. Reclamação aborda um evento atual. Protocolo aberto na entidade reclamada. Outra reclamação diferente no último mês. Cidadão fez algumas reclamações no último ano, a maioria improcedente".
Resposta	"Resposta longa. Resposta aborda um evento em passado distante e outro futuro. Outra resposta parecida no último mês. O segmento da entidade é Banco Múltiplo".

"MinMaxScaler"[1] para dimensionar e traduzir cada atributo individualmente de modo a comporem um intervalo determinado, especificamente entre 0 e 1. Referente às variáveis categóricas, as "Demandas Anteriores" e os "Protocolos Abertos" já apresentavam um domínio binário, de modo que seus valores foram apenas convertidos para uma codificação ordinal (Não = 0 e Sim = 1). Todavia, o "Segmento das IFs" é uma variável categórica nominal, compreendendo um conjunto finito de valores discretos sem relação entre si, logo não seguindo uma ordem natural. Dessa forma, foi necessário converter esse atributo para uma codificação a quente (*one-hot-encoding*), na qual cada rótulo foi transformado em uma variável binária (0 e 1) exclusiva [77]. A Figura 6.9 ilustra a estratégia *Concat* adotada.

Para melhor comparabilidade, o mesmo processo de otimização, por meio de validação cruzada com os mesmos hiperparâmetros e valores relacionados na Tabela 6.2, foi

[1]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

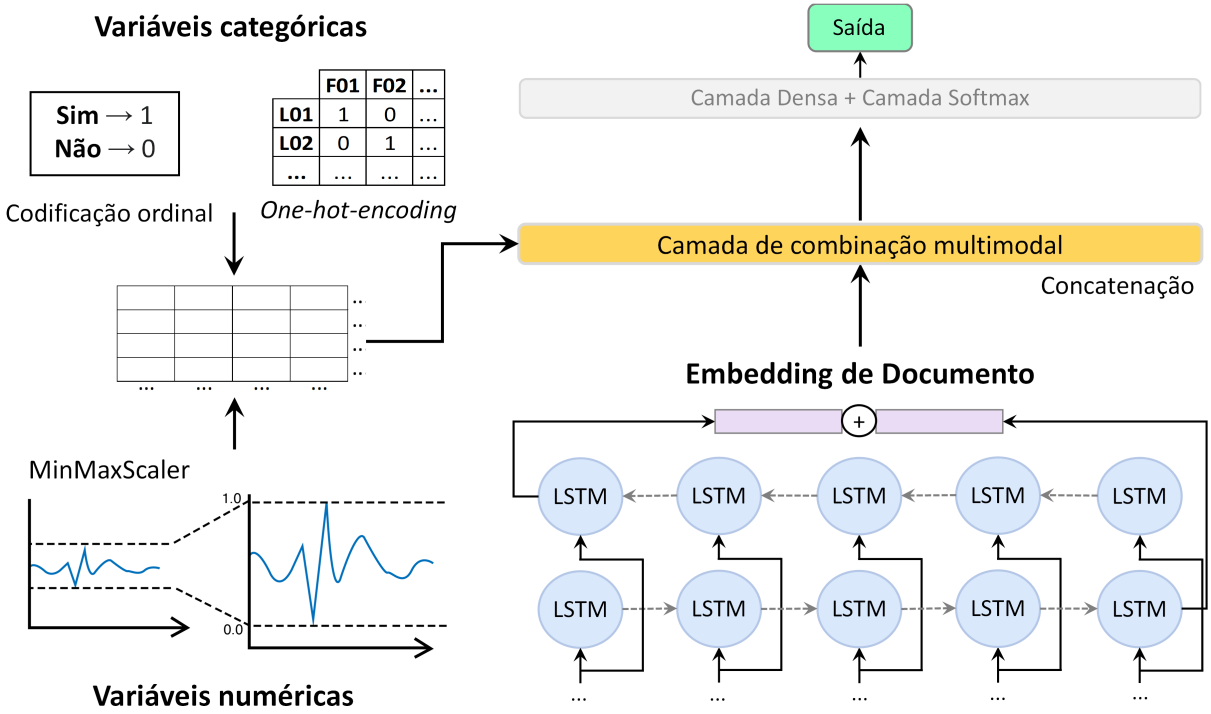


Figura 6.9: Estratégia *Concat* no modelo hierárquico (Fonte: elaborado pelo autor).

conduzido para selecionar, pela técnica de busca exaustiva, o modelo com maior desempenho médio. Na estratégia *Unimodal/All-Text*, a solução final foi denominada **Modelo Proposto$_{\text{alltext}}$**, ao passo que, na estratégia *Concat*, o classificador obtido foi referido como **Modelo Proposto$_{\text{concat}}$**.

6.4 Avaliação

Para a validação dos classificadores no processo de otimização, foi também considerada a métrica de desempenho definida na Seção 5.3, qual seja o PRAUC. Uma vez escolhidos os modelos com melhor desempenho, a mesma métrica foi utilizada para a comparação das soluções, entre si e com o Modelo BCB reproduzido no Capítulo 5, no intuito de estabelecer um classificador final para a tarefa de categorização das demandas abertas pelos clientes e usuários do SFN.

O modelo selecionado foi então treinado em todo o *dataset* de treinamento (reservando-se 5% para o `early_stopping`), sendo calculada a aludida métrica de desempenho para suas predições sobre o *dataset* de teste, visando estimar sua habilidade de generalização no mundo real. Complementarmente, foi calculada a medida "revocação por faixas", descrita na Seção 5.3, para avaliar o impacto do classificador final nas atividades de tratamento das reclamações.

Capítulo 7

Resultados e Análises

Os resultados obtidos com os experimentos descritos nos Capítulos 5 e 6 se encontram sumarizados nas tabelas a seguir, que elencam o arranjo de hiperparâmetros escolhidos e o desempenho médio alcançado - considerando a métrica PRAUC - para o Modelo $\text{BCB}_{\text{notab}}$ e o Modelo BCB (Tabela 7.1), bem como para o Modelo Proposto, o Modelo $\text{Proposto}_{\text{concat}}$ e o Modelo $\text{Proposto}_{\text{alltext}}$ (Tabela 7.2). Os valores do PRAUC calculados em cada dobra da validação cruzada podem ainda ser encontrados no Apêndice B.1.

Tabela 7.1: Resultados dos experimentos - Modelos BCB

Modelo	**Hiperparâmetros**	**PRAUC médio**
Modelo $\text{BCB}_{\text{notab}}$	`min_df`: 200 `n_gram_range`: (1, 2) `n_estimators`: 1500 `num_leaves`: 1500 `min_child_samples`: 200 `learning_rate`: 0.02 `reg_alpha`: 0 `reg_lambda`: 0	69,87%
Modelo BCB	`min_df`: 100 `n_gram_range`: (1, 2) `n_estimators`: 1500 `num_leaves`: 1500 `min_child_samples`: 200 `learning_rate`: 0.02 `reg_alpha`: 0 `reg_lambda`: 0	70,82%

É também apresentado, na Figura 7.1, gráfico comparativo do desempenho dos classificadores construídos, sendo considerado um Intervalo de Confiança (IC) - para a distribuição t de Student - de 83%, valor recomendado pelos autores em [78] para averiguar a igualdade de dois parâmetros a partir da sobreposição de intervalos.

Tabela 7.2: Resultados dos experimentos - Modelos Propostos

Modelo	**Hiperparâmetros**	**PRAUC médio**
Modelo Proposto	`stride`: 128 `learning_rate`: $3e^{-5}$ `num_train_epochs`: 2 `batch_size`: 2	71,41%
Modelo $\text{Proposto}_{\text{concat}}$	`stride`: 128 `learning_rate`: $3e^{-5}$ `num_train_epochs`: 3 `batch_size`: 2	71,58%
Modelo $\text{Proposto}_{\text{alltext}}$	`stride`: 128 `learning_rate`: $5e^{-5}$ `num_train_epochs`: 2 `batch_size`: 2	71,78%

Complementarmente, foi esboçada a curva PR, calculada sobre as predições dos modelos em cada dobra da validação cruzada, como ilustra a Figura 7.2. Como as partições para validação são mutuamente exclusivas, a curva foi construída com as estimativas dos modelos para todas as observações do treinamento. Nesse gráfico, a reta tracejada representa a linha-de-base, correspondendo à proporção de casos positivos (demandas procedentes) no *dataset*, qual seja 29,84%.

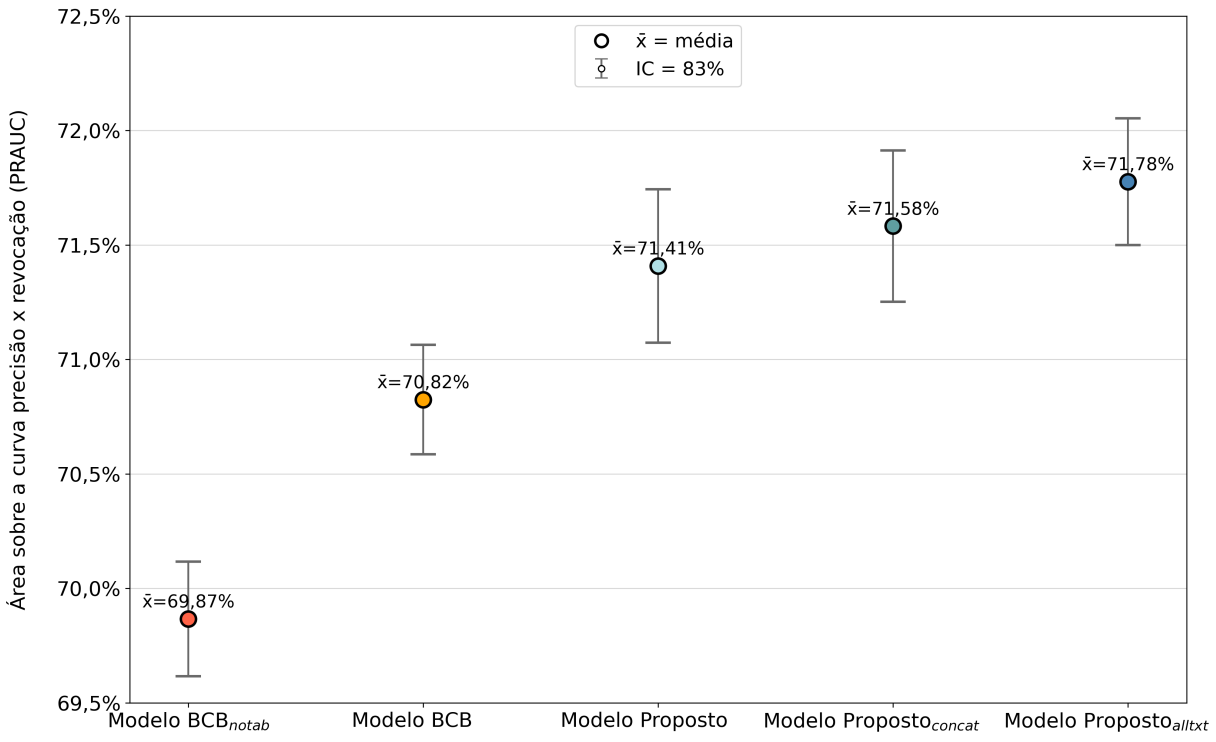


Figura 7.1: Comparativo do desempenho dos classificadores (Fonte: elaborado pelo autor).

Observa-se - tanto nas tabelas 7.1 e 7.2 quanto nas figuras 7.1 e 7.2 - que os valores calculados para o desempenho dos modelos se mostraram consideravelmente próximos, o que motivou a condução de teste estatístico para comparar os resultados obtidos.

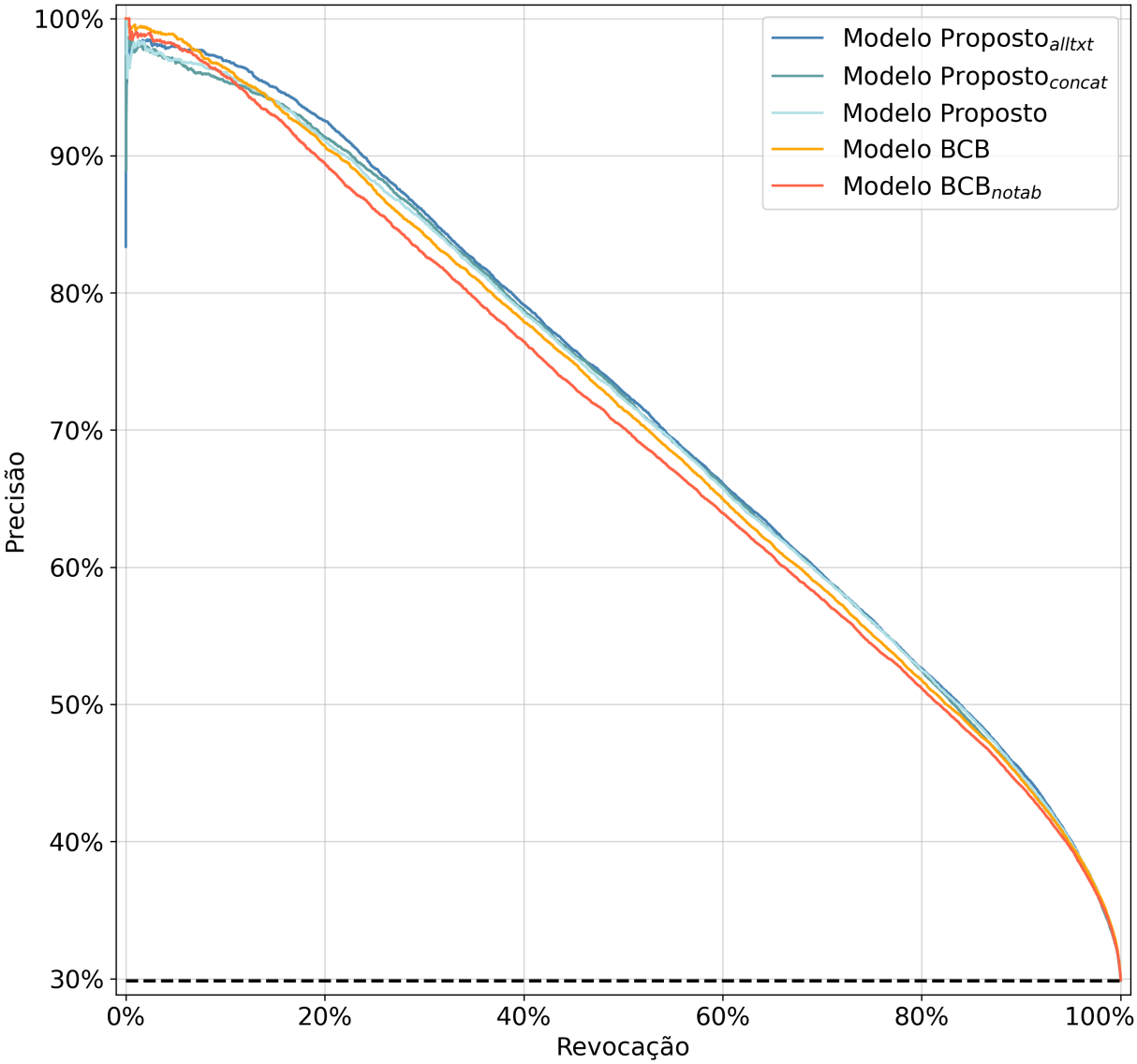


Figura 7.2: Curva precisão-revocação dos classificadores (Fonte: elaborado pelo autor).

Diferentes métodos são recomendados na literatura para determinar se um algoritmo de aprendizado supera outro em uma determinada tarefa [79, 80]. Especificamente para avaliações por meio de validação cruzada, um procedimento estatístico comumente empregado envolve a aplicação do teste t pareado para avaliar se a diferença no desempenho médio entre os dois modelos é estatisticamente significante [81, 82, 83]. Todavia, em estudo comparativo realizado em [84], os autores recomendaram a utilização do teste t para

variâncias desiguais, mais conhecido como *Welch's t-test*[1], se mostrando uma abordagem mais rigorosa e conservadora, motivo pelo qual foi selecionada para as comparações realizadas na presente pesquisa, apresentadas nas seções seguintes deste Capítulo. Os cálculos estatísticos efetuados podem ser encontrados no Apêndice B.2.

Contudo, cabe ressalvar que, como os *datasets* reservados para treinamento foram particionados com sobreposição quando da validação cruzada, é possível que a premissa de independência das amostras - fortemente assumida em procedimentos estatísticos comparativos [81, 82], inclusive no *Welch's t-test* [84] - tenha sido violada. Não obstante, diante da inexistência de outras alternativas para os experimentos conduzidos nesta pesquisa, optou-se pela manutenção do aludido teste estatístico para nortear a comparação entre os classificadores, reiterando ser um método amplamente utilizado na literatura.

7.1 Hipótese de pesquisa H1

Para avaliar a primeira hipótese de pesquisa (**H1**), que estabelece que "a utilização das variáveis tabulares levantadas pelo BCB em adição às variáveis de texto não contribui para a tarefa de classificação das demandas abertas pelos cidadãos", foram comparados os desempenhos do **Modelo BCB**, resultado da reprodução da solução desenvolvida pelo Banco Central sobre o *dataset* de experimentos, e do **Modelo BCB$_{\textbf{notab}}$**, construído da mesma forma que o anterior, porém removendo-se as variáveis tabulares definidas pela autarquia.

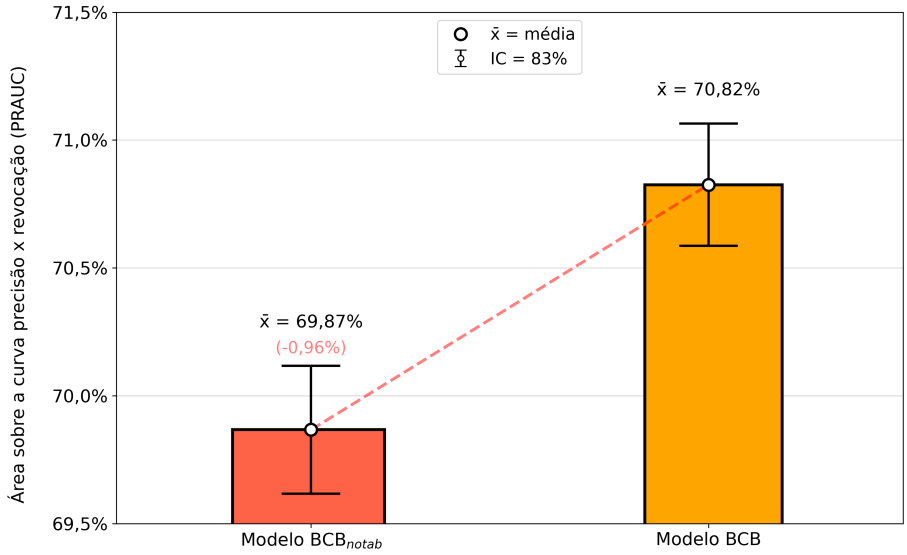


Figura 7.3: Modelo BCB$_{\text{notab}}$ x Modelo BCB (Fonte: elaborado pelo autor).

[1]https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html

A Figura 7.3, assim como a Figura 7.1, contém o PRAUC calculado para os classificadores considerando o mesmo intervalo de confiança de 83%, porém com uma representação em gráficos de barras e com filtro para os modelos destacados nesta Seção. Observa-se nítida diferença nos desempenhos, inexistindo sobreposição entre seus intervalos. Corroboram com essa afirmação os resultados obtidos com o *Welch's t-test*, como consta no Apêndice B.2, sendo calculada uma estatística $t = 4,6307$, com um respectivo p-valor $= 0,0017$, portanto rejeitando-se, para um nível de significância $\alpha = 5\%$, a hipótese nula de que as médias são idênticas. Logo, considerando que houve uma queda de 0,96 pontos percentuais no PRAUC médio ao remover as variáveis tabulares do Modelo BCB, **não é possível sustentar a hipótese de pesquisa H1**, restando inquestionável a relevância desses atributos para o classificador do Banco Central.

Acessoriamente, como o Modelo BCB adota um algoritmo de GBDT, qual seja o LightGBM, foi também explorada a importância relativa das variáveis tabulares para a classificação das demandas, calculada com base no ganho de informação otido com as partições dos nós em cada árvore construída[2] [29]. Constatou-se que, do total de 55.452 atributos selecionados no *ensemble* final do modelo, sendo 16 tabulares e 55.436 correspondentes a unigramas e bigramas gerados a partir dos textos da reclamação e resposta, apenas 14.819 (26,72% do total) foram utilizadas em *splits* durante a construção da solução. Dentre esses, verificou-se notória importância das variáveis númericas e categóricas, como ilustrado na Figura 7.4.

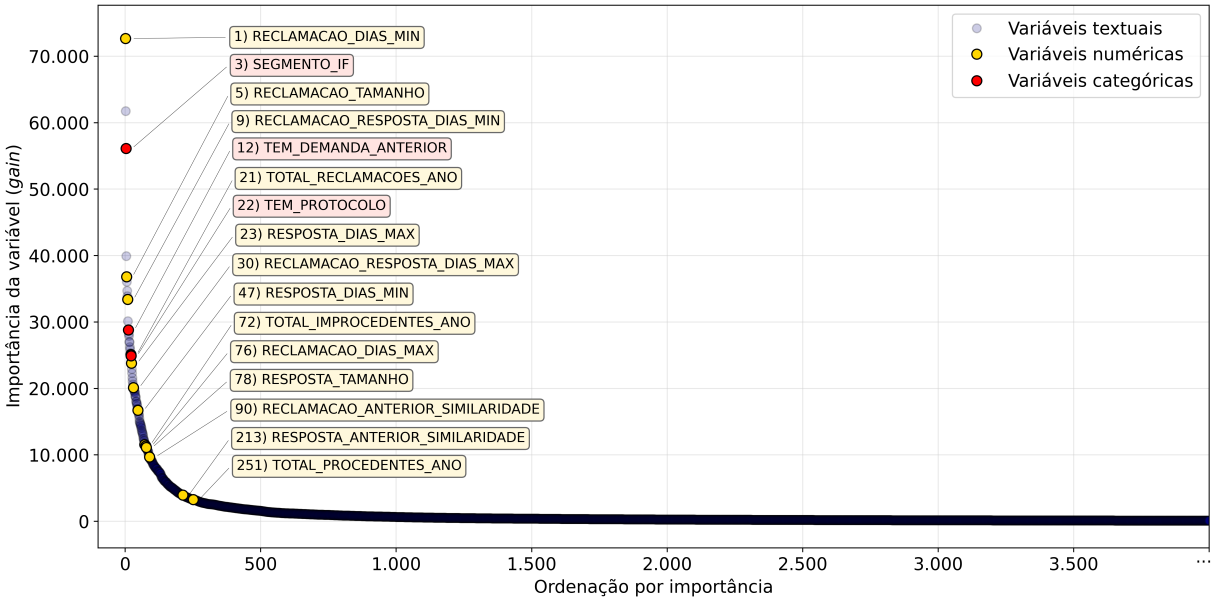


Figura 7.4: Curva de importância das variáveis (Fonte: elaborado pelo autor).

[2]https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.plot_importance.html

Pode-se observar que as variáveis tabulares estão entre as 300 com maior ganho informacional para o Modelo BCB, com destaque para a defasagem mínima de datas na reclamação (`RECLAMACAO_DIAS_MIN`), e para o segmento da IF (`SEGMENTO_IF`), que apresentaram ganhos consideravelmente maiores que as demais. Foi ainda construída nuvem de importância desses atributos, buscando ressaltar a relevância das variáveis tabulares frente às textuais, como consta da Figura 7.5. A título informativo, constatou-se que o atributo de texto com a maior importância relativa foi o bigrama "`novembro numnum`" na reclamação do cidadão, se tratando de data potencialmente relevante para a tarefa em questão, abarcando conhecimento possivelmente já extraído nas variáveis tabulares de defasagem de data.

Figura 7.5: Nuvem de importância das variáveis (Fonte: elaborado pelo autor).

7.2 Hipótese de pesquisa H2

No âmbito da hipótese de pesquisa **H2**, que sugere que "a utilização do BERT permite a construção de um classificador para as demandas abertas pelos cidadãos com melhor desempenho do que o desenvolvido pelo BCB", foi feita comparação entre o **Modelo BCB**, representando a solução do Banco Central no contexto desta pesquisa, e o **Modelo Proposto**, desenvolvido com base nos *embeddings* contextuais gerados pelo BERT. O desempenho desses classificadores foi ilustrado na Figura 7.6, com a mesma representação adotada na Seção 7.1.

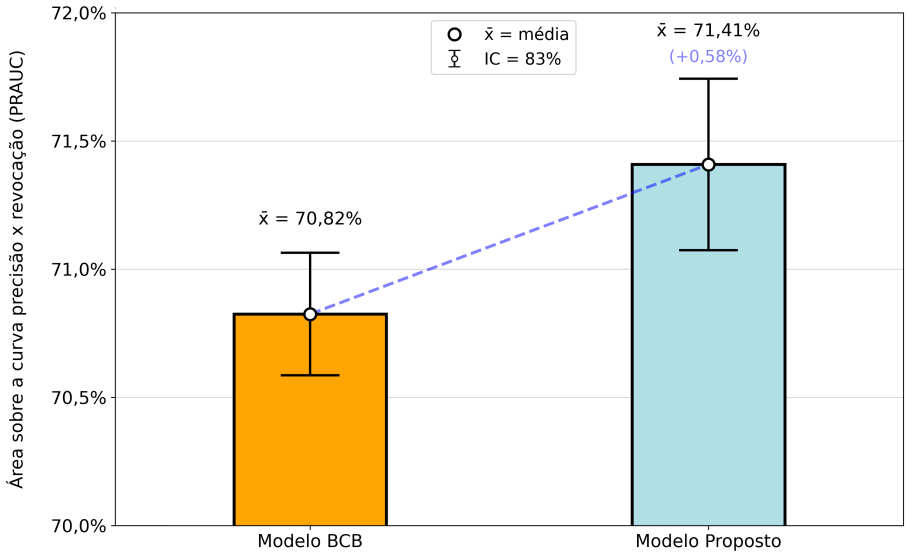


Figura 7.6: Modelo BCB x Modelo Proposto (Fonte: elaborado pelo autor).

Embora os valores calculados para o PRAUC tenham sido próximos, com os respectivos intervalos de confiança beirando a sobreposição, os resultados do *Welch's t-test* (t-valor $= 2,3723$, p-valor $= 0,0483$ e $\alpha = 5\%$) apontaram que a diferença entre o desempenho médio dos classificadores foi estatisticamente significante. Assim sendo, pode-se considerar que o Modelo Proposto - inobstante receber, como variável, apenas os atributos de texto, configurando a solução de menor complexidade proposta nos experimentos deste estudo - apresentou desempenho superior ao Modelo BCB, com um acréscimo de 0,58 pontos percentuais no PRAUC médio, **confirmando, portanto, a hipótese de pesquisa H1**.

7.3 Propostas multimodais

Foram também propostos, no Capítulo 6, modelos multimodais buscando combinar representações textuais obtidas a partir do BERT - o que se mostrou vantajoso para o problema definido nesta pesquisa, conforme destacado na Seção 7.2 - com as variáveis tabulares selecionadas pelo BCB - que apresentaram nítida relevância para a classificação das demandas, como aferido na Seção 7.1. No intuito de selecionar um classificador final para a tarefa em questão, foram feitas comparações com os modelos multimodais **Modelo Proposto$_{\mathbf{concat}}$** e **Modelo Proposto$_{\mathbf{alltext}}$**, assumindo como linha-de-base o **Modelo Proposto**, visto ter apresentado desempenho superior ao do Modelo BCB. Os valores calculados para o PRAUC constam da Figura 7.7.

Constatou-se pequeno ganho nos resultados obtidos com as soluções multimodais quando comparadas ao Modelo Proposto, com acréscimo de 0,17 pontos percentuais no

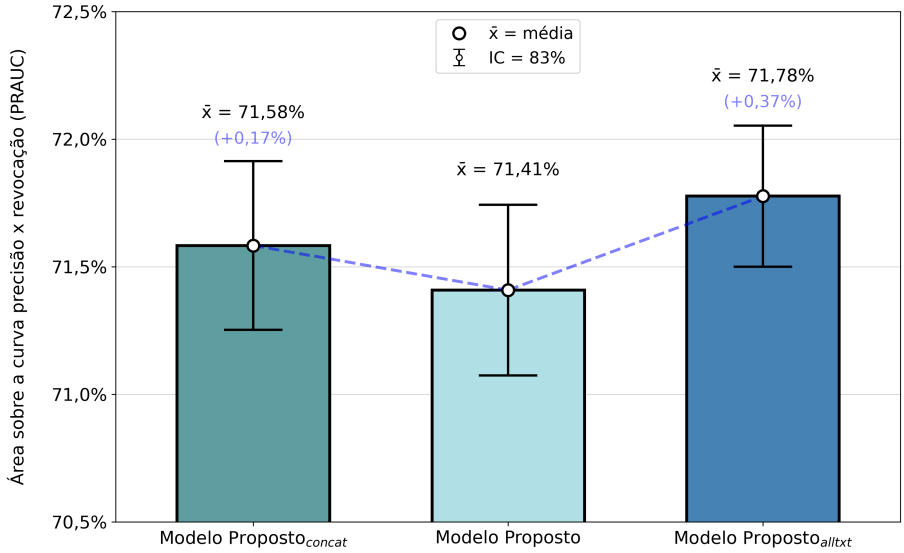


Figura 7.7: Modelo Proposto x Modelos Multimodais (Fonte: elaborado pelo autor).

PRAUC médio do Modelo Proposto$_{\text{concat}}$ e de 0,37 pontos percentuais no do Modelo Proposto$_{\text{alltext}}$. Contudo, observa-se considerável sobreposição entre os intervalos de confiança, não sendo possível, com os valores apurados com o *Welch's t-test* ($\alpha = 5\%$, com t-valor $= 0,6201$ e p-valor $= 0,5525$ para a estratégia *Concat* e t-valor $= 1,4165$ e p-valor $= 0,1957$ para a estratégia *All-Text*), rejeitar a hipótese nula de que as médias são idênticas. Em outras palavras, **os ganhos proporcionados pelos modelos multimodais não foram estatisticamente significantes no contexto desta pesquisa**.

Outrossim, em que pese as variáveis tabulares terem mostrado grande importância relativa para o modelo do BCB, a sua combinação com os atributos de texto no âmbito das estratégias multimodais propostas não foi suficiente para promover avanços significativos em termos de desempenho. Consequentemente, em linha com a navalha de Occam [82, 83], **o modelo final selecionado para a tarefa de classificação das reclamações abertas por clientes e usuários do SFN foi o Modelo Proposto**, sendo a solução mais simples dentre as propostas neste estudo, com desempenho superior ao do Modelo BCB e estatisticamente equivalente ao dos modelos multimodais.

7.4 Habilidade de generalização do classificador

O classificador final - Modelo Proposto - foi treinado sobre 95% dos dados de treinamento (151.616 observações), selecionados por meio de partição estratificada, com 5% (7.980 observações) sendo reservado para o `early_stopping`. Posteriormente, foram feitas predições com o modelo sobre o *dataset* de teste, de modo a calcular a métrica de desempenho

PRAUC, para estimar a habilidade de generalização do classificador, e a medida "revocação por faixas", visando avaliar o impacto da solução na atividade de tratamento das reclamações, conforme descrito nas Seções 5.3 e 6.4.

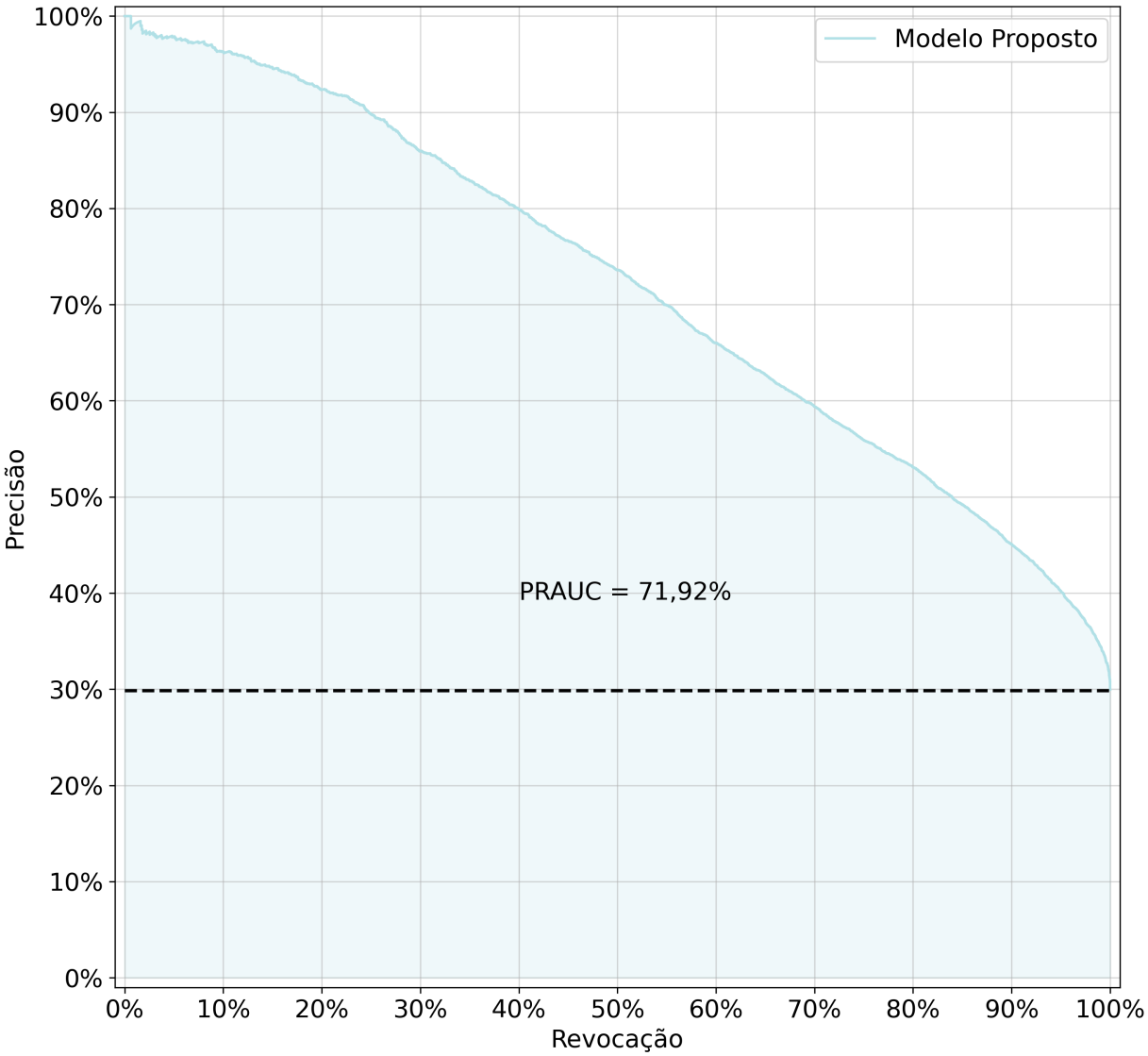


Figura 7.8: Estimativa final: curva precisão-revocação (Fonte: elaborado pelo autor).

Como resultado, apurou-se um PRAUC de 71,92%, representado na Figura 7.8 como a área sobre a curva PR, também calculada sobre as predições do modelo e esboçada no gráfico para fins ilustrativos. Observa-se que o desempenho estimado para o modelo foi superior aos obtidos com as soluções propostas por ocasião da validação cruzada (Figura 7.1), indicando resultados satisfatórios para suas habilidades de generalização na tarefa de interesse.

Tabela 7.3: Estimativa final: revocação por faixa

Faixa	100%	100,00%
Faixa	90%	99,55%
Faixa	80%	97,88%
Faixa	70%	94,79%
Faixa	60%	90,23%
Faixa	50%	83,89%
Faixa	40%	74,97%
Faixa	30%	63,76%
Faixa	20%	49,54%
Faixa	10%	29,06%

Por fim, a Tabela 7.3 elenca os valores calculados para a "revocação por faixas", com destaque para a faixa de 60%. Reitera-se que o objetivo dessa medida é estimar o ganho obtido com a utilização das predições do classificador para direcionar as atividades realizadas pelos servidores do BCB. Assim sendo, caso sejam analisadas apenas 60% das demandas, porém selecionadas com base no *score* gerado pelo modelo, estima-se uma abrangência de 90,23% do total daquelas de fato procedentes, ou seja, nas quais houve indício de descumprimento de obrigações regulatórias por parte das instituições financeiras.

Capítulo 8

Conclusão e Trabalhos Futuros

O trabalho apresentado realizou estudo e aplicação de técnicas de ML e NLP com o objetivo de desenvolver um modelo para ranquear as reclamações abertas pelos clientes e usuários do SFN de acordo com a sua chance de procedência. Assumindo como referência o classificador construído pelo BCB, foram estabelecidas duas hipóteses para nortear a pesquisa, relacionadas à importância das variáveis tabulares (categóricas e numéricas) e à representação dos atributos de texto. Foram explorados diferentes modelos, arquiteturas e estratégias no âmbito da abordagem de aprendizagem profunda, em contraponto ao método tradicional empregado na solução atualmente vigente, de modo a se obter melhores desempenhos para a tarefa de interesse.

Primeiramente, o trabalho contou com o entendimento do negócio, buscando compreender a atividade de tratamento das reclamações. Em seguida, foram coletados e selecionados os dados a serem utilizados nos experimentos, levando em consideração fatores limitantes da pesquisa, como a sensibilidade e o sigilo dos dados, o volume a ser processado, o cronograma definido, a capacidade de processamento da máquina disponível, e o tempo de treinamento estimado para os modelos propostos. Posteriormente, foi feita análise dos dados, com ênfase nos rótulos de procedência das demandas, nas variáveis textuais, quais sejam os textos da reclamação do cidadão e da resposta da entidade reclamada, e dos atributos tabulares considerados na solução vigente.

Neste estudo, o modelo do BCB foi adotado como linha-de-base, uma vez que foi resultado de diversos experimentos com abordagens tradicionais, abrangendo diferentes representações de texto e modelos clássicos, culminando em solução com o TF-IDF, representação amplamente empregada na literatura, e o LightGBM, algoritmo de ML bastante conceituado por sua eficiência, performance e interpretabilidade. Contudo, como foi selecionado um novo conjunto de dados para os experimentos conduzidos neste trabalho, fez-se necessária a reprodução da metodologia de construção do classificador do BCB para obter uma solução representativa no contexto desta pesquisa. Ademais, como a

proposta original foi desenvolvida por meio de experimentações, em detrimento da sua reproducibilidade, os hiperparâmetros de treinamento tiveram de ser escolhidos com base na literatura. Como extensão, o mesmo modelo foi também treinado removendo-se as variáveis tabulares, de modo a avaliar a sua relevância para a tarefa em questão.

Tendo como referência a solução desenvolvida pelo BCB e à luz das hipóteses de pesquisa definidas, foi proposto novo modelo para a tarefa de classificação das demandas abertas pelos cidadãos. A arquitetura utilizada contou com o BERTimbau [13], um BERT pré-treinado em um grande corpus em português (brWaC), seguido de uma BiLSTM, em uma estrutura hierárquica recomendada na literatura [15, 56] para contornar a limitação de tamanho das entradas do BERT. Nessa configuração, os textos pré-processados foram primeiro quebrados em segmentos (*chunks*) de 512 tokens, adotando-se uma janela de sobreposição com recuo de 128 tokens, para então serem transformados, pelo aludido LM, em *embeddings* contextuais a nível de sentença. Estes, em seguida, foram processados pela BiLSTM no intuito de se obter um único *embedding* para representar o documento.

Adicionalmente, com o objetivo de explorar as variáveis tabulares levantadas pelo BCB, dois outros classificadores foram construídos, adotando-se estratégias multimodais, sugeridas em trabalhos publicados recentemente [1, 2], para combinar esses atributos com os *embeddings* de contexto gerados pelo BERT. A primeira estratégia, denominada *All-Text*, envolveu a conversão das variáveis categóricas e numéricas em texto, recorrendo-se a transformações unimodais para discretizar, combinar e agrupar os dados em sentenças curtas passadas para o BERT junto das demais entradas de texto. Na segunda estratégia, chamada *Concat*, os atributos tabulares foram, primeiro, normalizados, e depois concatenados com o *embedding* de documento na saída da BiLSTM.

Para o treinamento e otimização dos modelos reproduzidos e construídos nesta pesquisa, foi reservado um conjunto composto por 80% dos dados de experimento (sendo os outros 20% para teste), sobre os quais foi aplicada técnica de validação cruzada com cinco dobras, considerando partições estratificadas, dada a natureza desbalanceada das classes (30% procedente e 70% improcedente). Foram definidas, com base no problema abordado no estudo, a métrica de desempenho PRAUC, para fins de comparação dos modelos e estimação da habilidade de generalização do classificador final, e a medida "revocação por faixas", para avaliação do impacto da solução na atividade de tratamento das reclamações.

No resultado dos experimentos, calculou-se um PRAUC médio de 70,82% para o classificador reproduzido do BCB, com uma queda de 0,96 pontos percentuais ao serem removidas as variáveis tabulares do modelo, indicando nítida relevância desses atributos para o problema em estudo, não sendo, portanto, possível sustentar a primeira hipótese de pesquisa ("a utilização das variáveis tabulares levantadas pelo BCB em adição às variáveis de texto não contribui para a tarefa de classificação das demandas abertas pelos cidadãos").

Por outro lado, o desempenho do modelo hierárquico proposto alcançou um PRAUC médio de 71,41%, promovendo um ganho de 0,58 pontos percentuais quando comparado com a linha-de-base, em consonância com a segunda hipótese levantada ("a utilização do BERT permite a construção de um classificador para as demandas abertas pelos cidadãos com melhor desempenho do que o desenvolvido pelo BCB"). Ademais, embora a adoção das estratégias multimodais tenha proporcionado ganhos frente ao desempenho obtido com a abordagem hierárquica, sendo 0,37 pontos percentuais com a *All-Text* e 0,17 pontos percentuais com a *Concat*, estes não se mostraram estatisticamente significantes.

Consequentemente, foi selecionado, como classificador final desta pesquisa, o modelo construído com a abordagem hierárquica BERT + BiLSTM, sendo a solução proposta de menor complexidade e apresentando desempenho superior ao do BCB e estatisticamente equivalente ao dos modelos multimodais. Foi calculado, para o modelo vencedor, um PRAUC de 71,92% sobre os dados de teste, indicando habilidades satisfatórias de generalização do classificador para a tarefa de interesse. Finalmente, com a medida "revocação por faixas", estimou-se que, caso a solução seja utilizada para direcionar as atividades realizadas pelo BCB de análise e julgamento das reclamações, seria possível, avaliando apenas 60% das demandas, abarcar 90,23% do total de procedentes, isto é, das situações nas quais, de fato, houve indício de inobservância de obrigações regulatórias pelas entidades supervisionadas.

8.1 Considerações finais e trabalhos futuros

Ainda que fontes de informação potencialmente importantes - como os anexos do cidadão e da instituição reclamada, e os documentos contendo as "súmulas" e "jurisprudências" dos atendimentos realizados - tenham sido descartadas na construção dos modelos, tendo em vista limitações técnicas da pesquisa, observou-se, com os resultados obtidos, que os classificadores propostos parecem ter aprendido conhecimentos relevantes para a tarefa de classificação das reclamações. Outra questão a ser observada é que a utilização de apenas cinco dobras na validação cruzada, limitada em decorrência do tempo de treinamento dos modelos, possivelmente impactou o poder estatístico dos testes empregados na comparação dos desempenhos calculados.

Quanto à eventual implementação do classificador proposto nesta pesquisa nos processos de trabalho do BCB, faz-se necessário, previamente, avaliar seu custo-benefício do ponto de vista de negócio. Embora a solução final dispense outras variáveis que não os textos da reclamação e da resposta, e tenha obtido desempenho superior, trata-se de modelo "caixa-preta", característico de métodos de DL, com visível desvantagem em termos de interpretabilidade frente ao classificador tradicional atualmente vigente. Dessa forma, as

predições do modelo não poderiam ser utilizadas para embasar ou respaldar julgamentos quanto à procedência das demandas. Cabe ainda ressalva quanto à quantidade de dados selecionadas para os experimentos deste trabalho, reduzida em decorrência do tempo de treinamento dos modelos propostos, limitação esta que, contudo, não se aplica ao classificador do BCB, que poderia ter sido rapidamente treinado em conjuntos maiores de dados. Enfim, registra-se que as contribuições científicas e tecnológicas realizadas neste estudo são de cunho estritamente acadêmico, de modo que as visões expressas são de responsabilidade do autor, não necessariamente representando o posicionamento institucional do Banco Central.

Adicionalmente, a título de trabalhos futuros, propõe-se, primeiramente, a aplicação de técnicas de balanceamento, como a reamostragem (*undersampling* e *oversampling*) [82, 10], para reequilibrar a proporção de classes do *dataset*, em prol da classificação desejada. Em seguida, objetiva-se a extração e o enriquecimento de informações a partir dos anexos encaminhados pelo cidadão e pelas entidades, bem como dos normativos externos e internos mantidos pelo BCB, por meio da metodologia de estruturação proposta em [42]. Pretende-se, ainda, estudar o uso de modelos robustos designados para o processamento de textos longos - como o Longformer [16] e o Reformer [53], que dispensam manobras hierárquicas - e a adoção de estratégias mais elaboradas para a combinação de variáveis tabulares e textuais, a exemplo do *Fuse-Early* e do *Fuse-Late* [2, 3]. Por fim, propõe-se adotar técnicas de interpretabilidade no intuito de explicar os resultados obtidos com modelos "caixa-preta", como o *Local Interpretable Model-agnostic Explanations* (LIME) [35], de modo a auxiliar nas análises feitas pelos servidores do BCB.

Referências

[1] Gu, Ken e Akshay Budhkar: *A package for learning on tabular and text data with transformers.* Em *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, páginas 69–73. Association for Computational Linguistics, 2021, ISBN 978-1-95-408525-1. doi: 10.18653/v1/2021.maiworkshop-1.10. 1, 2, 32, 33, 34, 35, 76, 77, 91

[2] Shi, Xingjian, Jonas Mueller, Nick Erickson, Mu Li e Alex Smola: *Multimodal autoML on structured tables with text fields.* Em *8th ICML Workshop on Automated Machine Learning (AutoML)*. ICML, 2021. url: https://openreview.net/forum?id=OHAIVOOl7Vl. 1, 34, 35, 76, 77, 91, 93

[3] Shi, Xingjian, Jonas Mueller, Nick Erickson, Nick Erickson, Mu Li, Alexander Smola e Alexander Smola: *Benchmarking multimodal automl for tabular data with text fields.* Em *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. J. Vanschoren and S. Yeung, 2021. url: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/9bf31c7ff062936a96d3c8bd1f8f2ff3-Paper-round2.pdf. 1, 2, 32, 34, 76, 77, 93

[4] Horttanainen, Esa Pekka Tapani: *Classificação automatizada de reclamações de usuários de serviços públicos: um estudo do caso aéreo.* Dissertação de mestrado, Instituto de Pesquisa Econômica Aplicada, 2019. 1, 18, 30, 62

[5] Li, Qian, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu e Lifang He: *A survey on text classification: From traditional to deep learning.* ACM Trans. Intell. Syst. Technol., 13, 2022. doi: 10.1145/3495162. 1, 6, 8, 9, 14, 15, 27, 28

[6] Yang, JinXiong, Liang Bai e Yanming Guo: *A survey of text classification models.* Em *Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence*, página 327–334. Association for Computing Machinery, 2020, ISBN 978-1-45-038830-6. doi: 10.1145/3438872.3439101. 2, 6, 8, 9, 14, 15

[7] Araújo, Pedro Henrique Luz de: *Domain-specific datasets for document classification and named entity recognition.* Dissertação de mestrado, Universidade de Brasília, 2021. 2, 7, 14, 17, 18, 20, 29, 66

[8] Howard, Jeremy e Sebastian Ruder: *Universal language model fine-tuning for text classification.* Em *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, páginas 328–339. Association for

Computational Linguistics, 2018, ISBN 978-1-94-808732-2. doi: 10.18653/v1/P18-1031. 2, 30

[9] Araújo, Pedro Henrique Luz de, Teófilo Emídio de Campos, Fabricio Ataides Braz e Nilton Correia da Silva: *VICTOR: a dataset for Brazilian legal documents classification*. Em *Proceedings of the 12th Language Resources and Evaluation Conference*, páginas 1449–1458. European Language Resources Association, 2020, ISBN 979-1-09-554634-4. 2, 8, 9

[10] Tang, Xiaobo, Hao Mou, Jiangnan Liu e Xin Du: *Research on automatic labeling of imbalanced texts of customer complaints based on text enhancement and layer-by-layer semantic matching*. Scientific Reports, 11, 2021. doi: 10.1038/s41598-021-91189-0. 2, 6, 14, 29, 65, 93

[11] Da Silva, Eric Hans Messias, João Laterza, Marcos Paulo Pereira Da Silva e Marcelo Ladeira: *A proposal to identify stakeholders from news for the institutional relationship management activities of an institution based on named entity recognition using bert*. Em *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, páginas 1569–1575. IEEE, 2021, ISBN 978-1-6654-4338-8. doi: 10.1109/ICMLA52953.2021.00251. 2, 14, 19, 20, 66, 69

[12] Devlin, Jacob, Ming Wei Chang, Kenton Lee e Kristina Toutanova: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Em *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, páginas 4171–4186. Association for Computational Linguistics, 2019, ISBN 978-1-95-073713-0. doi: 10.18653/v1/N19-1423. 2, 6, 19, 20, 21, 22, 66, 67, 68, 70, 75

[13] Souza, Fábio, Rodrigo Nogueira e Roberto Lotufo: *Bertimbau: Pretrained bert models for brazilian portuguese*. Em *Intelligent Systems*, páginas 403–417. Springer International Publishing, 2020, ISBN 978-3-03-061377-8. doi: https://doi.org/10.1007/978-3-030-61377-8_28. 2, 6, 15, 16, 20, 22, 28, 66, 67, 74, 91

[14] Souza, Fábio: *Bertimbau: modelos bert pré-treinados para português brasileiro*. Dissertação de mestrado, Universidade Estadual de Campinas, 2020. 2, 10, 14, 20, 21, 28, 66, 68, 71, 74, 75

[15] Pappagari, R., Piotr Żelasko, Jesús Villalba, Yishay Carmiel e Najim Dehak: *Hierarchical transformers for long document classification*. 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2019. doi: 10.1109/ASRU46091.2019.9003958. 2, 16, 31, 40, 70, 71, 73, 75, 91

[16] Beltagy, Iz, Matthew E. Peters e Arman Cohan: *Longformer: The long-document transformer*. ArXiv, abs/2004.05150, 2020. doi: 10.48550/ARXIV.2004.05150. 2, 16, 93

[17] Finardi, Paulo, José Dié Viegas, Gustavo T. Ferreira, Alex F. Mansano e Vinicius Fernandes Caridá: *Bertaú: Itaú BERT for digital customer service*. CoRR, abs/2101.12015, 2021. doi: 10.48550/ARXIV.2101.12015. 2

[18] Brasil: *Lei nº 4.595, de 31 de dezembro de 1964*. Diário Oficial da União, 1964. http://www.planalto.gov.br/ccivil_03/leis/l4595.htm, acesso em 2022-03-16, Dispõe sobre a Política e as Instituições Monetárias, Bancárias e Creditícias, Cria o Conselho Monetário Nacional e dá outras providências. 2, 36

[19] BCB: *Regimento interno do banco central do brasil*. Diário Oficial da União, 2020. https://www.in.gov.br/web/dou/-/portaria-n-108.150-de-27-de-agosto-de-2020-274962835, acesso em 2022-03-16. 3

[20] BCB: *Circular nº 3.729, de 17 de novembro de 2014*. Diário Oficial da União, 2014. https://www.bcb.gov.br/estabilidadefinanceira/exibenormativo?tipo=Circular&numero=3729, acesso em 2022-03-16, Altera a denominação do Sistema de Registro de Denúncias, Reclamações e Pedidos de Informação (RDR) e o tratamento de registros nesse sistema. 3, 36, 37

[21] BCB: *Relatório de gestão*. 2018. https://www.bcb.gov.br/publicacoes/relatoriogestao, acesso em 2022-03-16. 3, 36, 37

[22] BCB: *Relatório integrado do banco central*. 2019. https://www.bcb.gov.br/publicacoes/rig-nossosresultados, acesso em 2022-03-16. 3, 36, 37

[23] BCB: *Relatório de auditoria interna nº 2019/010*. 2019. https://www.bcb.gov.br/publicacoes/raint, acesso em 2022-03-16, Relata o resultado do trabalho de auditoria interna realizado de 18 de setembro de 2019 a 6 de dezembro de 2019 no objeto auditado Atendimento ao cidadão. 3, 38

[24] Brasil: *Decreto nº 9.203, de 22 de novembro de 2017*. Diário Oficial da União, 2017. http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2017/decreto/d9203.htm, acesso em 2022-03-16, Dispõe sobre a política de governança da administração pública federal direta, autárquica e fundacional. 3

[25] BCB: *Plano do projeto corporativo cidadania digital*. 2020. https://www.bcb.gov.br/publicacoes/relatoriogestao, acesso em 2022-03-16, Estabelece plano para implementação do Projeto Cidadania Digital no Banco Central do Brasil. 3, 38

[26] Brasil: *Decreto nº 9.319, de 21 de março de 2018*. Diário Oficial da União, 2018. http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/decreto/D9319.htm, acesso em 2022-03-16, Institui a Estratégia Brasileira para a Transformação Digital. 3

[27] Brasil: *Decreto nº 10.332, de 28 de abril de 2020*. Diário Oficial da União, 2020. https://www.in.gov.br/web/dou/-/decreto-n-10.332-de-28-de-abril-de-2020-254430358, acesso em 2022-03-16, Institui a Estratégia de Governo Digital para o período de 2020 a 2022, no âmbito dos órgãos e das entidades da administração pública federal direta, autárquica e fundacional e dá outras providências. 3

[28] Natekin, Alexey e Alois Knoll: *Gradient boosting machines, a tutorial*. Frontiers in Neurorobotics, 7, 2013. doi: 10.3389/fnbot.2013.00021. 4, 11, 60

[29] Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye e Tie Yan Liu: *Lightgbm: A highly efficient gradient boosting decision tree.* Em *Proceedings of the 31st International Conference on Neural Information Processing Systems*, página 3149–3157. Curran Associates Inc., 2017, ISBN 978-1-51-086096-4. doi: 10.5555/3294996.3295074. 4, 11, 13, 60, 61, 84

[30] Kadhim, Ammar Ismael: *Survey on supervised machine learning techniques for automatic text classification.* Artif. Intell. Rev., 52, 2019. doi: 10.1007/s10462-018-09677-1. 7, 8

[31] Andrade, Patrícia Helena Maia Alves de: *Aplicação de técnicas de mineração de textos para classificação de documentos: um estudo da automatização da triagem de denúncias na cgu.* Dissertação de mestrado, Universidade de Brasília, 2015. 7, 10, 22, 56, 58

[32] Minaee, Shervin, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu e Jianfeng Gao: *Deep learning–based text classification: A comprehensive review.* ACM Comput. Surv., 54, 2021. doi: 10.1145/3439726. 7, 8, 15, 17, 28, 66

[33] Hassan, Sayar Ul, Jameel Ahamed e Khaleel Ahmad: *Analytics of machine learning-based algorithms for text classification.* Sustainable Operations and Computers, 3, 2022. doi: https://doi.org/10.1016/j.susoc.2022.03.001. 7, 10

[34] Ibrahim, Yusuf, Emmanuel Okafor, Basira Yahaya, Shehu Mohammed Yusuf, Zainab Mukhtar Abubakar e Umar Yusuf Bagaye: *Comparative study of ensemble learning techniques for text classification.* Em *2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS)*, páginas 1–5. IEEE, 2021, ISBN 978-1-66-543494-2. doi: 10.1109/ICMEAS52683.2021.9692306. 8, 11

[35] Santos, Keila Barbosa Costa dos: *Categorização de textos por aprendizagem de máquina.* Dissertação de mestrado, Universidade Federal de Alagoas, 2019. 8, 9, 10, 14, 22, 54, 56, 93

[36] Costa Silva, Thalys Melicio da: *Um estudo comparativo entre algoritmos de aprendizagem de máquina supervisionados para predição de solução de reclamações no procon.* Monografia, Centro Universitário Christus, 2021. 8

[37] Zheng, Jianming, Yupu Guo, Chong Feng e Honghui Chen: *A hierarchical neural-network-based document representation approach for text classification.* Mathematical Problems in Engineering, 2018, 2018. doi: 10.1155/2018/7987691. 8, 16, 70, 73

[38] Kamath, Cannannore Nidhi, Syed Saqib Bukhari e Andreas Dengel: *Comparative study between traditional machine learning and deep learning approaches for text classification.* Em *Proceedings of the ACM Symposium on Document Engineering 2018*, páginas 1–11. Association for Computing Machinery, 2018, ISBN 978-1-45-035769-2. doi: 10.1145/3209280.3209526. 8

[39] Alzamzami, Fatimah, Mohamad Hoda e Abdulmotaleb El Saddik: *Light gradient boosting machine for general sentiment classification on short texts: A comparative evaluation*. IEEE Access, 8, 2020. doi: 10.1109/ACCESS.2020.2997330. 8, 11

[40] Subakti, Alvin, Hendri Murfi e Nora Hariadi: *The performance of bert as data representation of text clustering*. Journal of Big Data, 9, 2022. doi: 10.1186/s40537-022-00564-9. 9, 59, 65

[41] Souza Pita, Marcelo Rodrigo de: *Word embedding-based representations for short text*. Tese de doutorado, Universidade Federal de Minas Gerais, 2019. 10

[42] Paiva, Eduardo e Fernando Pereira: *Extraction and enrichment of features to improve complaint text classification performance*. Em *Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional*, páginas 338–349. SBC, 2021. doi: 10.5753/eniac.2021.18265. 10, 66, 93

[43] Wang, Congcong, Paul Nulty e David Lillis: *A comparative study on word embeddings in deep learning for text classification*. Em *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, página 37–46. Association for Computing Machinery, 2020, ISBN 978-1-45-037760-7. doi: 10.1145/3443279.3443304. 10, 14

[44] Tebbe, Eva e Benjamin Wegener: *Is natural language processing the cheap charlie of analyzing cheap talk? A horse race between classifiers on experimental communication data*. Journal of Behavioral and Experimental Economics, 96, 2022. doi: https://doi.org/10.1016/j.socec.2021.101808. 10, 11

[45] Chagas, Beatriz Nery Rodrigues: *Aplicações de algoritmos de aprendizado de máquina em crm: Revisão sistemática da literatura*. Dissertação de mestrado, Universidade Estadual do Maranhão, 2019. 10, 29

[46] Qi, Zhang: *The text classification of theft crime based on tf-idf and xgboost model*. Em *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, páginas 1241–1246. IEEE, 2020, ISBN 978-1-72-817006-0. doi: 10.1109/ICAICA50127.2020.9182555. 11

[47] Chen, Tianqi e Carlos Guestrin: *Xgboost: A scalable tree boosting system*. Em *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, página 785–794. Association for Computing Machinery, 2016, ISBN 978-1-45-034232-2. doi: 10.1145/2939672.2939785. 11, 12, 60

[48] Liang, Weizhang, Suizhi Luo, Guoyan Zhao e Hao Wu: *Predicting hard rock pillar stability using gbdt, xgboost, and lightgbm algorithms*. Mathematics, 8, 2020. doi: 10.3390/math8050765. 11, 12, 60

[49] Tang, Mingzhu, Qi Zhao, Steven X. Ding, Huawei Wu, Linlin Li, Wen Long e Bin Huang: *An improved lightgbm algorithm for online fault detection of wind turbine gearboxes*. Energies, 13, 2020. doi: 10.3390/en13040807. 12

[50] Daoud, Essam Al: *Comparison between xgboost, lightgbm and catboost using a home credit dataset*. International Journal of Computer and Information Engineering, 13, 2019. doi: https://doi.org/10.5281/zenodo.3607805. 13

[51] Wang, Jun, Xiaofang Zhang e Lin Chen: *How well do pre-trained contextual language representations recommend labels for GitHub issues?* Knowledge-Based Systems, 232, 2021. doi: https://doi.org/10.1016/j.knosys.2021.107476. 14, 17, 18, 66

[52] Mayeesha, Tasmiah Tahsin, Abdullah Md Sarwar e Rashedur M. Rahman: *Deep learning based question answering system in bengali*. Journal of Information and Telecommunication, 5, 2021. doi: 10.1080/24751839.2020.1833136. 16, 28, 66

[53] Kitaev, Nikita, Lukasz Kaiser e Anselm Levskaya: *Reformer: The efficient transformer*. Em *International Conference on Learning Representations*. ICLR, 2020. url: https://openreview.net/forum?id=rkgNKkHtvB. 16, 93

[54] Gao, Shang, Mohammed Alawad, M. Todd Young, John Gounley, Noah Schaefferkoetter, Hong Jun Yoon, Xiao Cheng Wu, Eric B. Durbin, Jennifer Doherty, Antoinette Stroup, Linda Coyle e Georgia Tourassi: *Limitations of transformers on clinical text classification*. IEEE Journal of Biomedical and Health Informatics, 25, 2021. doi: 10.1109/JBHI.2021.3062322. 16, 31

[55] Dai, Zihang, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le e Ruslan Salakhutdinov: *Transformer-XL: Attentive language models beyond a fixed-length context*. Em *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, páginas 2978–2988. Association for Computational Linguistics, 2019, ISBN 978-1-95-073749-9. doi: 10.18653/v1/P19-1285. 16

[56] Khandve, Snehal, Vedangi Wagh, Apurva Wani, Isha Joshi e Raviraj Joshi: *Hierarchical neural network approaches for long document classification*. CoRR, abs/2201.06774, 2022. doi: 10.1145/3529836.3529935. 16, 31, 32, 40, 69, 70, 73, 91

[57] Zhao, Wenjie, Gaoyu Zhang, George Yuan, Jun Liu, Hongtao Shan e Shuyi Zhang: *The study on the text classification for financial news based on partial information*. IEEE Access, 8, 2020. doi: 10.1109/ACCESS.2020.2997969. 18, 19, 74

[58] Ebrahimi, Mohammadreza, Mihai Surdeanu, Sagar Samtani e Hsinchun Chen: *Detecting cyber threats in non-english dark net markets: A cross-lingual transfer learning approach*. Em *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, páginas 85–90. IEEE, 2018, ISBN 978-1-53-867849-7. doi: 10.1109/ISI.2018.8587404. 18, 74

[59] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser e Illia Polosukhin: *Attention is all you need*. Em *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017, ISBN 978-1-51-086096-4. url: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf. 19

[60] Cunha, Urias Cruz da: *Utilização de técnicas de aprendizagem de máquina nos pagamentos de cobertura do proagro*. Monografia, Universidade de Brasília, 2019. 22, 23, 24, 25, 61, 62

[61] Ferri, C., J. Hernández-Orallo e R. Modroiu: *An experimental comparison of performance measures for classification*. Pattern Recognition Letters, 30, 2009. doi: https://doi.org/10.1016/j.patrec.2008.08.010. 22, 23, 25, 26

[62] Japkowicz, Nathalie: *Assessment Metrics for Imbalanced Learning*, páginas 187–206. John Wiley & Sons, Ltd, 2013, ISBN 978-1-11-864610-6. doi: https://doi.org/10.1002/9781118646106.ch8. 22, 23, 25, 26

[63] Branco, Paula, Luis Torgo e Rita Ribeiro: *A survey of predictive modelling under imbalanced distributions*. CoRR, abs/1505.01658, 2015. doi: 10.48550/ARXIV.1505.01658. 22, 24

[64] Saito, Takaya e Marc Rehmsmeier: *The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets*. PLOS ONE, 10, 2015. doi: 0.1371/journal.pone.0118432. 22, 25

[65] Davis, Jesse e Mark Goadrich: *The relationship between precision-recall and roc curves*. Em *Proceedings of the 23rd International Conference on Machine Learning*, página 233–240. Association for Computing Machinery, 2006, ISBN 978-1-59-593383-6. doi: 10.1145/1143844.1143874. 24, 25

[66] Boyd, Kendrick, Kevin H. Eng e C. David Page: *Area under the precision-recall curve: Point estimates and confidence intervals*. Em *Machine Learning and Knowledge Discovery in Databases*, páginas 451–466. Springer Berlin Heidelberg, 2013, ISBN 978-3-64-240994-3. doi: https://doi.org/10.1007/978-3-642-40994-3_29. 25

[67] Cambazoglu, B. Barla, Mark Sanderson, Falk Scholer e Bruce Croft: *A review of public datasets in question answering research*. SIGIR Forum, 54, 2021. doi: 10.1145/3483382.3483389. 28

[68] Gao, Yanjie, Yu Liu, Hongyu Zhang, Zhengxian Li, Yonghao Zhu, Haoxiang Lin e Mao Yang: *Estimating GPU Memory Consumption of Deep Learning Models*, página 1342–1352. Association for Computing Machinery, 2020, ISBN 978-1-45-037043-1. doi: https://doi.org/10.1145/3368089.3417050. 39

[69] Ivanov, Andrei, Nikoli Dryden, Tal Ben-Nun, Shigang Li e Torsten Hoefler: *Data movement is all you need: A case study on optimizing transformers*. Em *Proceedings of Machine Learning and Systems*, páginas 711–732. mlsys.org, 2021. doi: https://proceedings.mlsys.org/book/2020. 39

[70] Gusmão, Camila, Karla Figueiredo e Walkir Brito: *Técnicas de processamento de linguagem natural em denúncias criminais: Automatização e classificação de texto em português coloquial*. Em *Anais do XLVIII Seminário Integrado de Software e Hardware*, páginas 172–182. SBC, 2021. doi: 10.5753/semish.2021.15820. 56, 58, 59

[71] Bhattacharjee, Uddipta, P.K. Srijith e Maunendra Sankar Desarkar: *Term specific tf-idf boosting for detection of rumours in social networks*. Em *2019 11th International Conference on Communication Systems e Networks (COMSNETS)*, páginas 726–731. COMSNETS, 2019, ISBN 978-1-5386-7903-6. doi: 10.1109/COMSNETS.2019.8711427. 59

[72] Brownlee, Jason: *XGBoost With python: Gradient boosted trees with XGBoost and scikit-learn*. Machine Learning Mastery, 2016, ISBN 979-8-54-044627-3. 60, 61

[73] Wade, Corey: *Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with Python*. Packt Publishing Ltd, 2020, ISBN 978-1-83-921835-4. 60, 61

[74] Lopes-Cardoso, Henrique, Tomás Freitas Osório, Luís Vilar Barbosa, Gil Rocha, Luís Paulo Reis, João Pedro Machado e Ana Maria Oliveira: *Robust complaint processing in portuguese*. Information, 12, 2021. doi: 10.3390/info12120525. 66, 69

[75] Da Silva, Eric Hans Messias, João Laterza e Thiago Faleiros: *New state-of-the-art for question answering on portuguese squad v1.1*. Em *Anais do X Symposium on Knowledge Discovery, Mining and Learning*. SBC, 2022. 70, 71, 75

[76] Glasmachers, Tobias: *Limits of end-to-end learning*. Em *Proceedings of The 9th Asian Conference on Machine Learning, ACML 2017, Seoul, Korea, November 15-17, 2017*, páginas 17–32. PMLR, 2017. doi: http://proceedings.mlr.press/v77/glasmachers17a.html. 75

[77] Zheng, Alice e Amanda Casari: *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media, Inc., 2018, ISBN 978-1-49-195324-2. 78

[78] Payton, Mark E., Matthew H. Greenstone e Nathaniel Schenker: *Overlapping confidence intervals or standard error intervals: What do they mean in terms of statistical significance?* Journal of Insect Science, 3, 2003. doi: 10.1093/jis/3.1.34. 80

[79] Demšar, Janez: *Statistical comparisons of classifiers over multiple data sets*. The Journal of Machine Learning Research, 7, 2006. doi: 10.5555/1248547.1248548. 82

[80] Gardner, Josh e Christopher Brooks: *Statistical approaches to the model comparison task in learning analytics*. Em *Workshop on Methodology in Learning Analytics (MLA)*, páginas 1–14. CEUR-WS.org, 2017. url: http://ceur-ws.org/Vol-1915/paper2.pdf. 82

[81] Dietterich, Thomas G.: *Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms*. Neural Computation, 10, 1998. doi: 10.1162/089976698300017197. 82, 83

[82] Witten, Ian H., Eibe Frank e Mark A. Hall: *Chapter 5 - Credibility: Evaluating What's Been Learned*, páginas 147–187. Morgan Kaufmann, 2011, ISBN 978-0-12-374856-0. doi: https://doi.org/10.1016/B978-0-12-374856-0.00005-5. 82, 83, 87, 93

[83] Mitchell, Tom M.: *Machine learning, International Edition.* McGraw-Hill, 1997, ISBN 978-0-07-042807-2. 82, 87

[84] Ruxton, Graeme D.: *The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test.* Behavioral Ecology, 17, 2006. doi: 10.1093/beheco/ark016. 82, 83, 111

Apêndice A

Informações complementares dos dados

A.1 Segmento das Instituições Financeiras

A.2 Transformação unimodal das variáveis tabulares

Nesta Seção são apresentadas as transformações efetuadas nas variáveis tabulares - por ocasião da construção do **Modelo Proposto$_{\mathbf{alltext}}$** - visando padronizar as entradas para a modalidade de texto. Para isso, os atributos categóricos e numéricos foram discretizados, combinados e agrupados de acordo com julgamento de negócio e com sua distribuição no *dataset* de experimentos, resultando em sentenças curtas a serem incluídas no início dos textos da reclamação do cidadão e da resposta da IF.

A.2.1 Transformações para o texto da reclamação

As transformações unimodais efetuadas nas variáveis tabulares a serem incorporadas no texto da reclamação do cidadão estão descritas nas tabelas apresentadas a seguir.

Protocolos abertos

A variável categórica "Protocolos abertos" deu origem ao domínio apresentado na Tabela A.2, contendo as possíveis sentenças a serem incorporadas no texto da reclamação.

Tabela A.1: Distribuição das demandas por segmento da IF no *dataset* de experimento

Segmento da IF	**Demandas**
Banco Múltiplo	113.486 (56,89%)
Caixa Econômica Federal	26.170 (13,12%)
Instituição de Pagamento	24.805 (12,43%)
Banco do Brasil - Banco Múltiplo	13.590 (6,81%)
Sociedade de Crédito, Financiamento e Investimento	6.731 (3,37%)
Administradora de Consórcio	3.775 (1,89%)
Banco Comercial	2.746 (1,38%)
Cooperativa de Crédito	2.308 (1,16%)
Instituição de Pagamento não sujeita a autorização pelo BCB	1.993 (1,00%)
Sociedade Corretora de TVM	1.315 (0,66%)
Sociedade de Crédito Direto	1.146 (0,57%)
Sociedade de Crédito ao Microempreendedor	536 (0,27%)
Sociedade Distribuidora de TVM	262 (0,13%)
Banco Múltiplo Cooperativo	249 (0,12%)
Sociedade de Empréstimo entre Pessoas	84 (0,04%)
Sociedade Corretora de Câmbio	79 (0,04%)
Sociedade de Arrendamento Mercantil	50 (0,03%)
Associação de Poupança e Empréstimo	36 (0,02%)
Banco de Investimento	25 (0,01%)
Banco de Câmbio	23 (0,01%)
Companhia Hipotecária	22 (0,01%)
BNDES	18 (0,01%)
Banco de Desenvolvimento	17 (0,01%)
Entidade Operadora Infraestrutura Mercado Financeiro - IMF	13 (0,01%)
Agência de Fomento	9 (0,00%)
Banco Comercial Estrangeiro - Filial no país	7 (0,00%)

Tabela A.2: Transformação para protocolos abertos

Valor	**Sentença resultante**	**Proporção**
Sim	Protocolo aberto na entidade reclamada	53,15%
Não	Nenhum protocolo aberto na entidade reclamada	46,85%

Tamanho do texto da reclamação

A variável categórica "Tamanho do texto da reclamação" deu origem ao domínio apresentado na Tabela A.3, contendo as possíveis sentenças a serem incorporadas no texto da reclamação, geradas com base na distribuição dos dados no *dataset* de experimentos.

Defasagens de data na reclamação: Mínima e Máxima

Para as variáveis de defasagem de data na reclamação, as transformações unimodais foram efetuadas em duas etapas. A primeira consistiu em classificar as variáveis numéricas

Tabela A.3: Transformação para o tamanho do texto da reclamação

Valor	Sentença resultante	Proporção
< 475	Reclamação muito curta	25%
475 a 727	Reclamação curta	25%
728 a 1175	Reclamação longa	25%
> 1175	Reclamação muito longa	25%

"Defasagem mínima na reclamação" e "Defasagem máxima na reclamação" de acordo com julgamento de negócio, resultando no domínio apresentado na Tabela A.4.

Tabela A.4: Classificação das defasagens de data na reclamação

Valor	Descrição (eventos ocorridos...)	Classificação	Proporção	
			Defas. Mín.	Defas. Máx.
$<$ - 30	a mais de 30 dias	passado distante	16,61%	9,47%
-1 a -30	entre 1 e 30 dias	passado próximo	23,09%	22,62%
0	na data da demanda	atual	59,49%	64,44%
> 0	após a data da demanda	futuro	0,81%	3,47%

Na segunda etapa, foi feita comparação entre as variáveis "Defasagem mínima na reclamação" e "Defasagem máxima na reclamação", no intuito de definir se tratam-se de eventos concomitantes (considerados um único evento para fins de registro de data), ocorridos em períodos iguais (com a mesma classificação na Tabela A.4), ou em períodos distintos (classificação diferente na Tabela A.4), conforme domínio descrito na Tabela A.5. Por fim, as sentenças finais obtidas para esses atributos se encontram elencadas na Tabela A.6. Como as mesmas transformações são realizadas para as defasagens de data na resposta (Subseção A.2.2), tornou-se redundante converter as variáveis de defasagem de data global em texto, não sendo estas, portanto, consideradas na transformação unimodal.

Tabela A.5: Comparação entre defasagens de data mínima e máxima

Ocorrência	Enquadramento
Defas. mín. = Defas. máx.	Evento único
Defas. mín. $\neq$ Defas. máx., com classificações iguais	Dois eventos iguais
Defas. mín. $\neq$ Defas. máx., com classificações diferentes	Dois eventos distintos

Demandas Anteriores e Similaridade entre reclamações

Como as variáveis numéricas de similaridade - "Similaridade entre reclamações" e "Similaridade entre respostas" - estão diretamente associadas à variável categórica "Demandas Anteriores", já que é necessário existir uma demanda anterior para avaliar a similaridade entre os textos da reclamação/resposta, a transformação unimodal com esses atributos

Tabela A.6: Sentenças para as defasagens de data da reclamação

Sentença resultante	Proporção
Reclamação aborda um evento em passado distante	6,62%
Reclamação aborda um evento em passado próximo	13,76%
Reclamação aborda um evento atual	59,33%
Reclamação aborda um evento futuro	0,74%
Reclamação aborda dois eventos em passado distante	2,85%
Reclamação aborda dois eventos em passado próximo	4,86%
Reclamação aborda dois eventos futuros	0,07%
Reclamação aborda um evento em passado distante e outro em passado próximo	4,00%
Reclamação aborda um evento em passado distante e outro atual	1,70%
Reclamação aborda um evento em passado distante e outro futuro	1,44%
Reclamação aborda um evento em passado próximo e outro atual	3,41%
Reclamação aborda um evento em passado próximo e outro futuro	1,06%
Reclamação aborda um evento atual e outro futuro	0,16%

foi realizada de forma conjunta. Assim, no âmbito das reclamações, foi considerado o domínio apresentado na Tabela A.7, contendo as possíveis sentenças a serem incorporadas no texto da reclamação, geradas com base na distribuição dos dados no *dataset* de experimentos.

Tabela A.7: Transformação para a similaridade das reclamações

Valor das variáveis tabulares		Sentença resultante	Prop.
Demanda Anterior	Similaridade reclamações		
Não	-	Única reclamação no último mês	88,57%
Sim	0,00 a 0,13	Outra reclamação muito diferente no último mês	2,86%
Sim	0,14 a 0,23	Outra reclamação diferente no último mês	2,86%
Sim	0,24 a 0,40	Outra reclamação parecida no último mês	2,86%
Sim	0,41 a 1,00	Outra reclamação muito parecida no último mês	2,86%

Reclamações por ano: Total, Procedentes e Improcedentes

Para a transformação unimodal, as variáveis "Total de reclamações no ano", "Total de procedentes no ano" e "Total de improcedentes no ano" foram combinadas em única sentença, construída por meio de procedimento de duas etapas. Na primeira, foi feita classificação a partir da variável "Total de reclamações no ano", com base em julgamento de negócio, considerando o domínio estabelecido na Tabela A.8. Ressalta-se que esse atributo abrange todas as demandas nos últimos 365 dias, tanto as encerradas quanto as abertas.

Na segunda etapa, as demandas abertas por cidadãos que tenham feito mais de uma reclamação no último ano, isto é, outras além daquela sendo avaliada pelos classificadores, foram, primeiro, categorizadas como encerradas ou não. As encerradas, por sua vez,

Tabela A.8: Classificação do total de reclamações no ano

Valor	Classificação	Proporção
1	Uma	80,08%
2 a 10	Algumas	17,04%
> 10	Muitas	2,88%

foram então avaliadas de acordo com a proporção de procedência observada, nos moldes descritos na Tabela A.9. Dessa forma, foram obtidas as sentenças finais para as variáveis em questão, conforme apresentado na Tabela A.10.

Tabela A.9: Enquadramento quanto ao encerramento e procedência

Encerradas como procedente	Enquadramento
100%	Todas procedentes
75% a 99%	Grande maioria procedente
51% a 74%	Maioria procedente
50%	Metade procedente
25% a 49%	Maioria improcedente
1% a 24%	Grande maioria improcedente
0%	Todas improcedentes

Tabela A.10: Sentenças para a quantidade e procedência das reclamações no último ano

Sentença resultante	Proporção
Cidadão só fez uma reclamação no último ano	80,08%
Cidadão fez algumas reclamações no último ano, ainda não encerradas	5,06%
Cidadão fez algumas reclamações no último ano, todas procedentes	1,58%
Cidadão fez algumas reclamações no último ano, a grande maioria procedente	0,09%
Cidadão fez algumas reclamações no último ano, a maioria procedente	0,36%
Cidadão fez algumas reclamações no último ano, metade procedentes	1,47%
Cidadão fez algumas reclamações no último ano, a maioria improcedente	1,26%
Cidadão fez algumas reclamações no último ano, a grande maioria improcedente	0,39%
Cidadão fez algumas reclamações no último ano, todas improcedentes	6,83%
Cidadão fez muitas reclamações no último ano, ainda não encerradas	0,11%
Cidadão fez muitas reclamações no último ano, todas procedentes	0,02%
Cidadão fez muitas reclamações no último ano, a grande maioria procedente	0,01%
Cidadão fez muitas reclamações no último ano, a maioria procedente	0,06%
Cidadão fez muitas reclamações no último ano, metade procedentes	0,06%
Cidadão fez muitas reclamações no último ano, a maioria improcedente	0,49%
Cidadão fez muitas reclamações no último ano, a grande maioria improcedente	1,42%
Cidadão fez muitas reclamações no último ano, todas improcedentes	0,71%

A.2.2 Transformações para o texto da resposta

As transformações unimodais efetuadas nas variáveis tabulares a serem incorporadas no texto da resposta do cidadão estão descritas nas tabelas apresentadas a seguir.

Segmento da Instituição Financeira

Para a variável categórica "Segmento da Instituição Financeira", a conversão em texto envolveu simplesmente o acréscimo da sequência "O segmento da entidade é " previamente ao rótulo do atributo, dando origem ao domínio apresentado na Tabela A.11, com as possíveis sentenças a serem incorporadas no texto da resposta.

Tabela A.11: Transformação para segmento da Instituição Financeira

Sentença resultante	**Proporção**
O segmento da entidade é Banco Múltiplo.	56,89%
O segmento da entidade é Caixa Econômica Federal.	13,12%
O segmento da entidade é Instituição de Pagamento.	12,43%
O segmento da entidade é Banco do Brasil - Banco Múltiplo.	6,81%
O segmento da entidade é Sociedade de Crédito, Financiamento e Investimento.	3,37%
O segmento da entidade é Administradora de Consórcio.	1,89%
O segmento da entidade é Banco Comercial.	1,38%
O segmento da entidade é Cooperativa de Crédito.	1,16%
O segmento da entidade é Instituição de Pagamento não sujeita a autorização pelo BCB.	1,00%
O segmento da entidade é Sociedade Corretora de TVM.	0,66%
O segmento da entidade é Sociedade de Crédito Direto.	0,57%
O segmento da entidade é Sociedade de Crédito ao Microempreendedor.	0,27%
O segmento da entidade é Sociedade Distribuidora de TVM.	0,13%
O segmento da entidade é Banco Múltiplo Cooperativo.	0,12%
O segmento da entidade é Sociedade de Empréstimo entre Pessoas.	0,04%
O segmento da entidade é Sociedade Corretora de Câmbio.	0,04%
O segmento da entidade é Sociedade de Arrendamento Mercantil.	0,03%
O segmento da entidade é Associação de Poupança e Empréstimo.	0,02%
O segmento da entidade é Banco de Investimento.	0,01%
O segmento da entidade é Banco de Câmbio.	0,01%
O segmento da entidade é Companhia Hipotecária.	0,01%
O segmento da entidade é BNDES.	0,01%
O segmento da entidade é Banco de Desenvolvimento.	0,01%
O segmento da entidade é Entidade Operadora Infraestrutura Mercado Financeiro - IMF.	0,01%
O segmento da entidade é Banco Comercial Estrangeiro - Filial no país.	0,00%
O segmento da entidade é Agência de Fomento.	0,00%
O segmento da entidade é Outros.	0,00%

Tamanho do texto da resposta

A variável categórica "Tamanho do texto da resposta" deu origem ao domínio apresentado na Tabela A.12, contendo as possíveis sentenças a serem incorporadas no texto da resposta, geradas com base na distribuição dos dados no *dataset* de experimentos.

Tabela A.12: Transformação para o tamanho do texto da resposta

Valor	Sentença resultante	Proporção
< 697	Resposta muito curta	25%
697 a 1706	Resposta curta	25%
1707 a 2701	Resposta longa	25%
> 2701	Resposta muito longa	25%

Defasagens de data na resposta: Mínima e Máxima

Para as variáveis de defasagem de data na resposta, as transformações unimodais foram efetuadas em duas etapas. A primeira consistiu em classificar as variáveis numéricas "Defasagem mínima na resposta" e "Defasagem máxima na resposta" de acordo com julgamento de negócio (o mesmo aplicado sobre as datas nas reclamações), resultando no domínio apresentado na Tabela A.13.

Tabela A.13: Classificação das defasagens de data na reclamação

Valor	Descrição (eventos ocorridos...)	Classificação	Proporção	
			Defas. Mín.	Defas. Máx.
$<$ - 30	a mais de 30 dias	passado distante	34,69%	10,97%
-1 a -30	entre 1 e 30 dias	passado próximo	14,46%	12,37%
0	na data da demanda	atual	44,27%	46,94%
> 0	após a data da demanda	futuro	6,58%	29,72%

Na segunda etapa, foi feita comparação entre as variáveis "Defasagem mínima na resposta" e "Defasagem máxima na resposta", no intuito de definir se tratam-se de eventos concomitantes (considerados um único evento para fins de registro de data), ocorridos em períodos iguais (com a mesma classificação na Tabela A.13), ou em períodos distintos (classificação diferente na Tabela A.13), conforme o mesmo domínio descrito na Tabela A.5. Por fim, as sentenças finais obtidas para esses atributos se encontram elencadas na Tabela A.14. Como as mesmas transformações são realizadas para as defasagens de data na reclamação (Subseção A.2.1), tornou-se redundante converter as variáveis de defasagem de data global em texto, não sendo estas, portanto, consideradas na transformação unimodal.

Demandas Anteriores e Similaridade entre respostas

Como as variáveis numéricas de similaridade - "Similaridade entre reclamações" e "Similaridade entre respostas" - estão diretamente associadas à variável categórica "Demandas Anteriores", já que é necessário existir uma demanda anterior para avaliar a similaridade entre os textos da reclamação/resposta, a transformação unimodal com esses atributos

Tabela A.14: Sentenças para as defasagens de data da resposta

Sentença resultante	**Proporção**
Resposta aborda um evento em passado distante	5,42%
Resposta aborda um evento em passado próximo	4,57%
Resposta aborda um evento atual	42,58%
Resposta aborda um evento futuro	5,35%
Resposta aborda dois eventos em passado distante	5,55%
Resposta aborda dois eventos em passado próximo	2,02%
Resposta aborda dois eventos futuros	1,23%
Resposta aborda um evento em passado distante e outro em passado próximo	5,79%
Resposta aborda um evento em passado distante e outro atual	2,74%
Resposta aborda um evento em passado distante e outro futuro	15,19%
Resposta aborda um evento em passado próximo e outro atual	1,61%
Resposta aborda um evento em passado próximo e outro futuro	6,26%
Resposta aborda um evento atual e outro futuro	1,69%

foi realizada de forma conjunta. Assim, no âmbito das respostas, foi considerado o domínio apresentado na Tabela A.15, contendo as possíveis sentenças a serem incorporadas no texto da resposta, geradas com base na distribuição dos dados no *dataset* de experimentos.

Tabela A.15: Transformação para a similaridade das respostas

Valor das variáveis tabulares		**Sentença resultante**	**Prop.**
Demanda Anterior	**Similaridade respostas**		
Não	-	Única resposta no último mês	88,57%
Sim	0,00 a 0,33	Outra resposta muito diferente no último mês	2,86%
Sim	0,34 a 0,58	Outra resposta diferente no último mês	2,86%
Sim	0,59 a 0,93	Outra resposta parecida no último mês	2,86%
Sim	0,94 a 1,00	Outra resposta muito parecida no último mês	2,86%

Apêndice B

Testes estatísticos

B.1 Resultados da validação cruzada

Tabela B.1: Resultados da validação cruzada com 5 *folds*

Modelo	**PRAUC (Área sobre a curva precisão-revocação)**						
	***Fold*1**	***Fold*2**	***Fold*3**	***Fold*4**	***Fold*5**	**Méd. ($\bar{x}$)**	**Desv. Pad. (s)**
$\text{BCB}_{\text{notab}}$	0,7002	0,6947	0,7016	0,7014	0,6954	0,6987	0,0033
BCB	0,7112	0,7052	0,7107	0,7097	0,7044	0,7082	0,0032
Proposto	0,7156	0,7119	0,7199	0,7151	0,7079	0,7141	0,0045
$\text{Prop}_{\text{concat}}$	0,7185	0,7144	0,7218	0,7142	0,7103	0,7158	0,0044
$\text{Prop}_{\text{alltext}}$	0,7207	0,7135	0,7222	0,7176	0,7149	0,7178	0,0037

B.2 Teste *t* para variâncias desiguais (*Welch's t-test*)

No intuito de verificar estatisticamente se o desempenho médio de dois modelos é diferente, foi realizado, com base nas recomendações de [84], o teste t para variâncias desiguais, também conhecido como *Welch's t-Test*. Para isso foi traçada a seguinte Hipótese Nula (H_0): "O desempenho médio obtido na validação cruzada com 5 *folds* foi o mesmo para ambos os modelos", de modo que $H_0 : \mu_1 = \mu_2$.

Os valores da média ($\bar{x}$), do desvio padrão (s) e do tamanho da amostra (n) - equivalente ao número de *folds* considerados na validação cruzada - foram obtidos da Tabela B.1. Para a aplicação do teste, foram utilizadas as equações Equação B.1 e Equação B.2, adaptadas de [84], referentes ao cálculo do teste estatístico (t') e do grau de liberdade (v), respectivamente, considerando variâncias desiguais para as duas amostras, e assumido um nível de significância de 5% ($\alpha = 0,05$) .

$$t' = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{B.1}$$

$$v = \frac{(\frac{1}{n_1} + \frac{u}{n_2})^2}{\frac{1}{n_1^2(n_1-1)} + \frac{u^2}{n_2^2(n_2-1)}}, \text{ em que } u = \frac{s_2^2}{s_1^2} \tag{B.2}$$

B.2.1 Modelo BCB x Modelo BCB$_{notab}$

O primeiro teste buscou averiguar estatisticamente se o desempenho médio do Modelo BCB foi igual ao do Modelo BCB$_{notab}$, sendo obtidos os seguintes resultados:

Modelo BCB: $\bar{x}_1 = 0,7082$, $s_1 = 0,0032$ e $n_1 = 5$

Modelo BCB$_{notab}$: $\bar{x}_2 = 0,6987$, $s_2 = 0,0033$ e $n_2 = 5$

Teste estatístico: $t' = \frac{0,7082-0,6987}{\sqrt{\frac{0,0032^2}{5} + \frac{0,0033^2}{5}}} = 4,6307$

Grau de liberdade: $u = \frac{0,0033^2}{0,0032^2} = 1,0983; v = \frac{(\frac{1}{5} + \frac{1,0983}{5})^2}{\frac{1}{5^2(5-1)} + \frac{1,0983^2}{5^2(5-1)}} = 7,9825$

Valor teórico (t-valor): $t_{val} = t_{\alpha,v} = t_{0.05,7.9825} = 2,3069$

Nível descritivo (p-valor): $p_{val} = P(t > |2,3069|) = 0,0017$

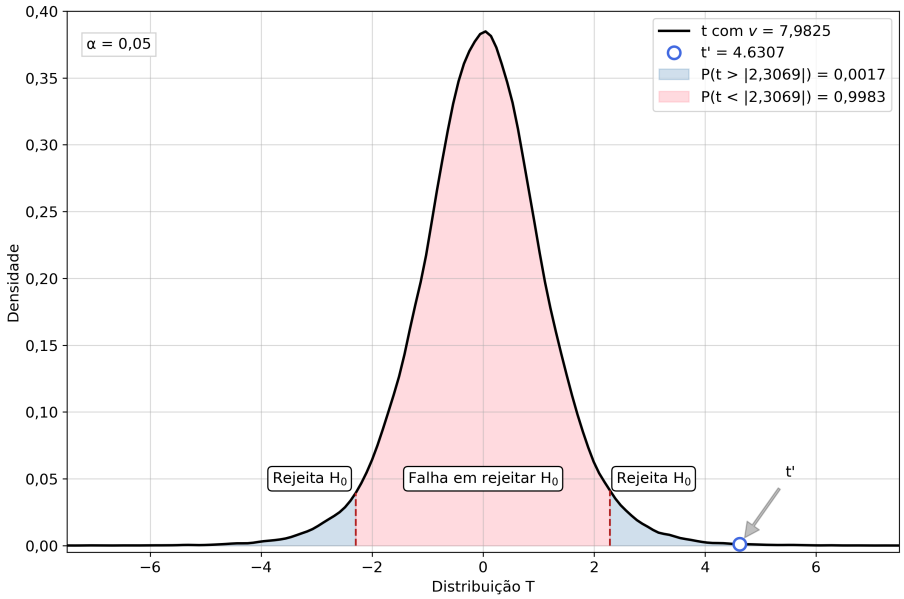


Figura B.1: Welch's t-Test: BCB x BCB$_{notab}$ (Fonte: elaborado pelo autor).

Como $t' > t_{val}$ (pois $4,6307 > 2,3069$) e $\alpha > p_{val}$ (pois $0,0500 > 0,0017$), conforme ilustra a Figura B.1, **podemos refutar a hipótese nula H_0 de que o desempenho médio do Modelo BCB foi igual ao do Modelo BCB$_{notab}$.**

B.2.2 Modelo Proposto x Modelo BCB

O segundo teste buscou averiguar estatisticamente se o desempenho médio do Modelo Proposto foi igual ao do Modelo BCB, sendo obtidos os seguintes resultados:

Modelo Proposto: $\bar{x}_1 = 0,7141,\ s_1 = 0,0045$ e $n_1 = 5$

Modelo BCB: $\bar{x}_2 = 0,7082,\ s_2 = 0,0032$ e $n_2 = 5$

Teste estatístico: $t' = \frac{0,7141 - 0,7082}{\sqrt{\frac{0,0045^2}{5} + \frac{0,0032^2}{5}}} = 2,3723$

Grau de liberdade: $u = \frac{0,0032^2}{0,0045^2} = 0,5085; v = \frac{(\frac{1}{5} + \frac{0,5085}{5})^2}{\frac{1}{5^2(5-1)} + \frac{0,5085^2}{5^2(5-1)}} = 7,2320$

Valor teórico (t-valor): $t_{val} = t_{\alpha,v} = t_{0.05,7.2320} = 2,3493$

Nível descritivo (p-valor): $p_{val} = P(t > |2,3493|) = 0,0483$

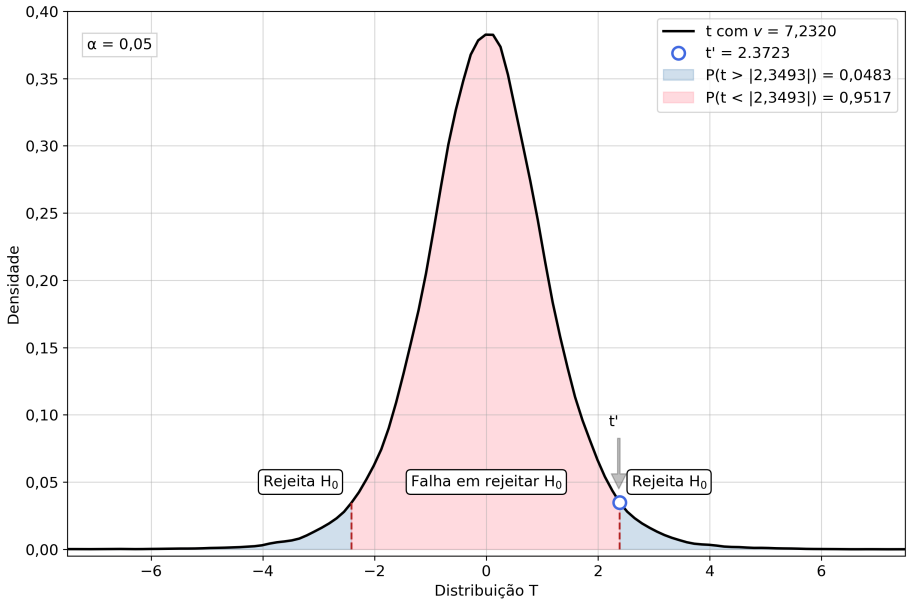


Figura B.2: Welch's t-Test: Proposto x BCB (Fonte: elaborado pelo autor).

Como $t' > t_{val}$ (pois $2,3723 > 2,3493$) e $\alpha > p_{val}$ (pois $0,0500 > 0,0483$), conforme ilustra a Figura B.2, **podemos refutar a hipótese nula H_0 de que o desempenho médio do Modelo Proposto foi igual ao do Modelo BCB**.

B.2.3 Modelo Proposto$_{concat}$ x Modelo Proposto

O terceiro teste buscou averiguar estatisticamente se o desempenho médio do Modelo Proposto$_{concat}$ foi igual ao do Modelo Proposto, sendo obtidos os seguintes resultados:

Modelo Proposto$_{concat}$: $\bar{x}_1 = 0,7158$, $s_1 = 0,0044$ e $n_1 = 5$

Modelo Proposto: $\bar{x}_2 = 0,7141$, $s_2 = 0,0045$ e $n_2 = 5$

Teste estatístico: $t' = \frac{0,7578 - 0,7141}{\sqrt{\frac{0,0044^2}{5} + \frac{0,0045^2}{5}}} = 0,6201$

Grau de liberdade: $u = \frac{0,0045^2}{0,0044^2} = 1,0248; v = \frac{(\frac{1}{5} + \frac{1,0248}{5})^2}{\frac{1}{5^2(5-1)} + \frac{1,0248^2}{5^2(5-1)}} = 7,9988$

Valor teórico (t-valor): $t_{val} = t_{\alpha,v} = t_{0.05,7.9988} = 2,3061$

Nível descritivo (p-valor): $p_{val} = P(t > |2,3061|) = 0,5525$

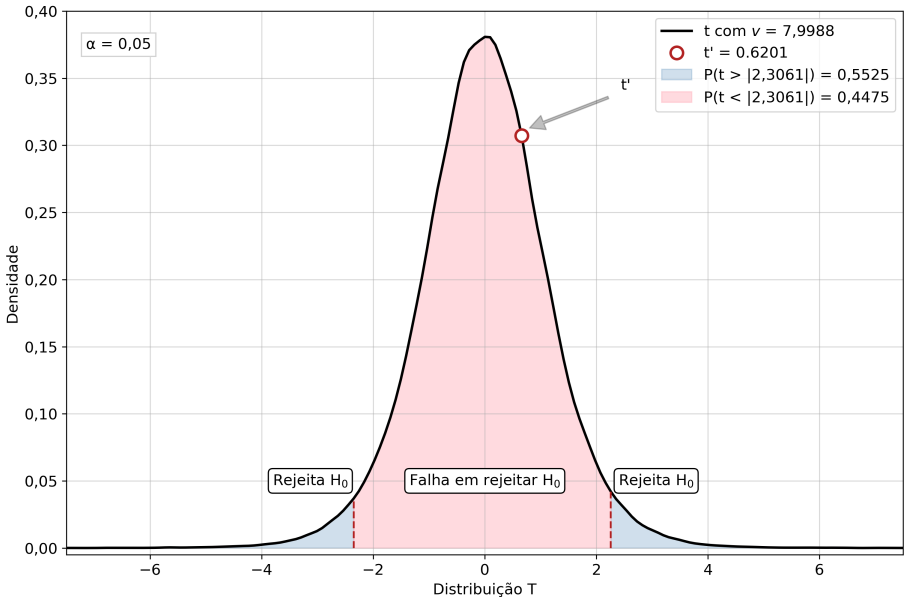


Figura B.3: Welch's t-Test: Proposto$_{concat}$ x Proposto (Fonte: elaborado pelo autor).

Como $t' < t_{val}$ (pois $0,6201 < 2,3061$) e $\alpha < p_{val}$ (pois $0,0500 < 0,5525$), conforme ilustra a Figura B.3, **não podemos refutar a hipótese nula H_0 de que o desempenho médio do Modelo Proposto$_{concat}$ foi igual ao do Modelo Proposto**.

B.2.4 Modelo Proposto$_{alltext}$ x Modelo Proposto

O quarto teste buscou averiguar estatisticamente se o desempenho médio do Modelo Proposto$_{alltext}$ foi igual ao do Modelo Proposto, sendo obtidos os seguintes resultados:

Modelo Proposto$_{alltext}$: $\bar{x}_1 = 0,7178$, $s_1 = 0,0037$ e $n_1 = 5$

Modelo Proposto: $\bar{x}_2 = 0,7141$, $s_2 = 0,0045$ e $n_2 = 5$

Teste estatístico: $t' = \frac{0,7178-0,7141}{\sqrt{\frac{0,0037^2}{5}+\frac{0,0045^2}{5}}} = 1,4165$

Grau de liberdade: $u = \frac{0,0045^2}{0,0037^2} = 1,4637; v = \frac{(\frac{1}{5}+\frac{1,4637}{5})^2}{\frac{1}{5^2(5-1)}+\frac{1,4637^2}{5^2(5-1)}} = 7,7263$

Valor teórico (t-valor): $t_{val} = t_{\alpha,v} = t_{0.05,7.7263} = 2,3203$

Nível descritivo (p-valor): $p_{val} = P(t > |2,3203|) = 0,1957$

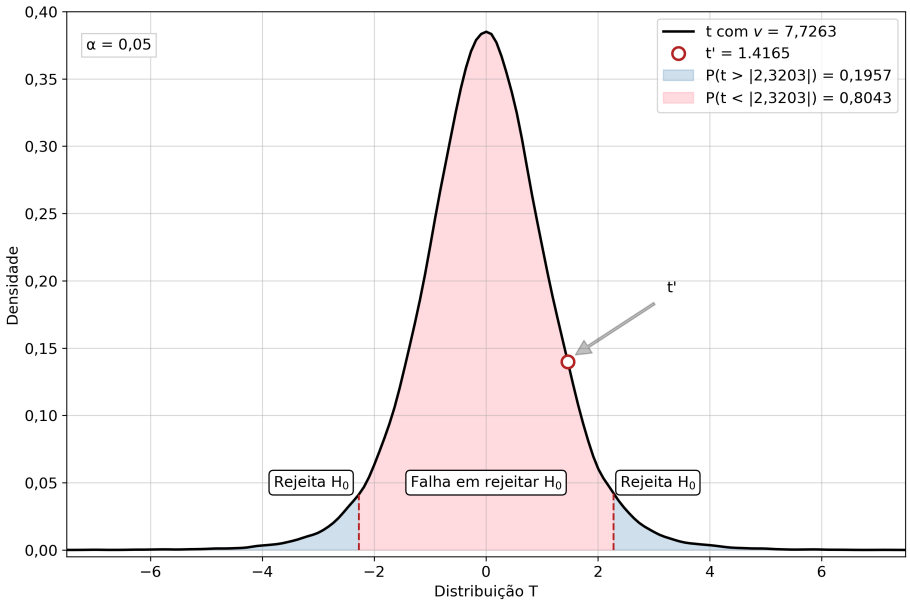


Figura B.4: Welch's t-Test: Proposto$_{alltext}$ x Proposto (Fonte: elaborado pelo autor).

Como $t' < t_{val}$ (pois $1,4165 < 2,3203$) e $\alpha < p_{val}$ (pois $0,0500 < 0,1957$), conforme ilustra a Figura B.4, **não podemos refutar a hipótese nula H_0 de que o desempenho médio do Modelo Proposto$_{alltext}$ foi igual ao do Modelo Proposto**.