CLASSIFICAÇÃO DE DENÚNCIAS: UTILIZAÇÃO DE PROCESSAMENTO DE LINGUAGEM NATURAL PARA APRIMORAR ATIVIDADES DE OUVIDORIA

Eduardo Soares de Paiva

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Civil.

Orientador: Nelson Francisco Favilla Ebecken
D.Sc.

Rio de Janeiro
Março de 2023

CLASSIFICAÇÃO DE DENÚNCIAS: UTILIZAÇÃO DE PROCESSAMENTO DE LINGUAGEM NATURAL PARA APRIMORAR ATIVIDADES DE OUVIDORIA

Eduardo Soares de Paiva

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Orientador: Nelson Francisco Favilla Ebecken D.Sc.

Aprovada por: Prof. Nelson Francisco Favilla Ebecken
Prof. Rogério Pinto Espíndola
Prof. Mário Antônio Ribeiro Dantas
Prof. João Baptista do Oliveira Souza Filho
Prof. Elton Fernandes

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2023

Soares de Paiva, Eduardo

Classificação de Denúncias: utilização de Processamento de Linguagem Natural para aprimorar atividades de ouvidoria/Eduardo Soares de Paiva. – Rio de Janeiro: UFRJ/COPPE, 2023.

XI, 103 p.: il.; 29, 7cm.

Orientador: Nelson Francisco Favilla Ebecken D.Sc.

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Civil, 2023.

Referências Bibliográficas: p. 93 – 103.

1. Classificação de denúncias. 2. Processamento de linguagem Natural. 3. Classificação Textual. I. D.Sc., Nelson Francisco Favilla Ebecken. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

CLASSIFICAÇÃO DE DENÚNCIAS: UTILIZAÇÃO DE PROCESSAMENTO DE LINGUAGEM NATURAL PARA APRIMORAR ATIVIDADES DE OUVIDORIA

Eduardo Soares de Paiva

Março/2023

Orientador: Nelson Francisco Favilla Ebecken D.Sc.

Programa: Engenharia Civil

No Brasil, os cidadãos podem fazer denúncias sobre irregularidades na Administração Pública. Porém, para que essas denúncias sejam apuradas, elas precisam passar por uma triagem, que verifica se elas reúnem os requisitos mínimos para a apuração. Esse é um processo demorado, que precisa de automação. No entanto, essa triagem utiliza informações que não estão nos textos. Atualmente, os trabalhos que tratam de classificação textual não abordam a utilização de fontes de dados externas para a classificação. Dessa forma, essa pesquisa propõe e avalia diferentes soluções para esse problema. Sendo assim, nosso objetivo é desenvolver um modelo de classificação textual que prevê se uma denúncia deve ser classificada como apta ou não, a partir do seu conteúdo. Para isso, avaliamos diferentes arquiteturas de rede, a fim de identificar a mais apropriada para o problema de classificação textual em questão. Também propomos dois métodos de sumarização de denúncias, a fim de verificar se os textos sumarizados traziam ganhos para o processo de classificação. Uma metodologia de extração de variáveis, a partir do texto das denúncias, também foi proposta. Essa metodologia utilizava as variáveis extraídas para buscar outras informações em bases de dados externas. Por fim, o modelo proposto era composto por outros dois modelos: um que utilizava os dados textuais e a arquitetura BERT, e outro que utilizava as variáveis estruturadas, extraídas dos textos e de bases de dados externas, e o algoritmo lightGBM. Esse modelo já está em produção, fazendo a classificação de forma automática da parte das denúncias recebidas pelo Governo Federal. Sendo assim, o modelo já está trazendo benefícios para o processo de apuração de denúncias, e consequentemente para a sociedade como um todo.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

COMPLAINTS CLASSIFICATION: USING NATURAL LANGUAGE PROCESSING TO ENHANCE OMBUDSMAN ACTIVITIES

Eduardo Soares de Paiva

March/2023

Advisor: Nelson Francisco Favilla Ebecken D.Sc.

Department: Civil Engineering

In Brazil, citizens can report irregularities in public administration. However, for these complaints to be investigated, they need to be triaged. This screening checks if the complaints are consistent and have the minimum requirements to proceed with the investigation process. This is a time-consuming process that needs automation. However, this screening uses other information that is not present in the texts. Currently, papers about textual classification do not address the use of external data sources to carry out the classification. Thus, this research aims to propose and evaluate different solutions to this problem. Therefore, our objective is to develop a textual classification model to predict whether a complaint should be classified as able or not able, based on the content of the complaints and their attachments. For this, we evaluated different network architectures to identify the most appropriate one for the textual classification problem in question. We also propose two methods of summarizing complaints, to verify if the summarized texts improved the classification process. A methodology for extracting variables from the text of the complaint was also proposed. This methodology used the extracted variables to search for other information in external databases. Finally, the proposed classification model was composed of two other models: one using textual data and the BERT architecture, and another based on structured variables, extracted from texts and external databases, and the lightGBM algorithm. The proposed model is already in production, automatically classifying part of the complaints received by the federal government. Therefore, the model is already bringing benefits to the process of invastigating complaints, and consequently to society.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Este capítulo fornece uma visão geral da pesquisa, apresentando sua motivação, o problema abordado, os principais desafios, questões de pesquisa, os objetivos a serem alcançados, as bases de dados utilizadas e as métricas de avaliação.

1.1 Motivação

O conceito de democracia pode ser dividido em duas vertentes: a democracia representativa e a democracia participativa. A democracia representativa é aquela que é exercida quando os cidadãos escolhem, por meio do voto, os representantes que irão atuar nos seus interesses. Nesse caso, a representação é o exercício do poder político por meio do trabalho dos deputados, senadores, governadores, prefeitos, vereadores, entre outros agentes políticos que são eleitos periodicamente [8].

Já a democracia participativa se legitima na ideia de que todo poder político vem do povo, como está previsto no parágrafo único do artigo 1º da Constituição Federal, e na participação e controle social dos usuários de serviços públicos na Administração Pública direta e indireta, conforme o art. 37 da Constituição Federal [8].

Sendo assim, o controle social é uma forma de exercício da democracia. Nesse tipo de controle, o próprio cidadão tem a possibilidade de verificar a regularidade da atuação da Administração, a fim de impedir a prática de atos ilegítimos, lesivos aos indivíduos ou a coletividade ou que possibilitem a reparação dos danos decorrentes da prática de tais atos [9].

Visando operacionalizar essa modalidade de controle, foram instituídas as ouvidorias públicas. As ouvidorias públicas são canais que facilitam a interação entre os cidadãos e a Administração Pública, oferecendo assim uma forma para que o cidadão possa exercer o controle social.

No âmbito federal, esse papel é desempenhado pela Ouvidoria-Geral da União (OGU), ligada à Controladoria-Geral da União (CGU). A OGU é responsável por

receber, examinar e encaminhar denúncias, reclamações, elogios, sugestões, solicitações de informação e pedidos de simplificação referentes a procedimentos e ações de agentes públicos, órgãos e entidades do Poder Executivo Federal [10].

Dessa forma, a OGU recebe uma série de denúncias que precisam ser tratadas, analisadas e apuradas. Uma das primeiras atividades referentes ao tratamento das denúncias recebidas é a análise de aptidão das denúncias. Nessa análise de aptidão, verifica-se todo o material referente a uma determinada denúncias (textos das denúncias e anexos auxiliares) para definir se tal denúncia reúne os requisitos mínimos para o prosseguimento do rito apuratório.

O volume de material a ser analisado por denúncia, aliado à grande quantidade de denúncias que são recebidas, dificultam uma resposta tempestiva para essa análise de aptidão. Sendo assim, faz-se necessário um mecanismo automatizado para a realização de tais análises. Logo, o desenvolvimento de técnicas de processamento de linguagem natural capazes de realizar a classificação de uma denúncia como apta ou não, com base no seu conteúdo textual, torna-se um problema em aberto.

Esse problema consiste em um problema de classificação textual. No entanto, algumas peculiaridades distinguem o problema em questão dos casos de classificação textual já abordados na literatura, visto que, essa classificação deve levar em consideração outras informações que não estão presentes no conteúdo dos textos.

1.2 Definição do Problema

A Ouvidoria Geral da União (OGU) recebe as denúncias por meio de um sistema denominado Fala.BR[1]. Esse sistema permite que os cidadãos façam denúncias sobre irregularidades que estão acontecendo na Administração Pública Federal. Atualmente, a OGU recebe cerca de 380 denúncias por mês. Essas denúncias podem vir acompanhadas de arquivos anexos que são utilizados para acrescentar outras informações a respeito dos fatos narrados.

No entanto, para saber se tais denúncias possuem informações consistentes para serem apuradas, faz-se necessária a identificação, validação e análise das situações narradas. Essa checagem exige um grande esforço e dispêndio de tempo por parte de servidores da Ouvidoria Geral da União. Ao final dessa análise, as denúncias são classificadas como Aptas ou Não Aptas, sendo que, as denúncias Aptas seguem o rito normal de apuração.

A Figura 1.1 ilustra o fluxo de trabalho executado pelos servidores na tarefa de análise de aptidão de denúncias, sendo que, os números dentro dos círculos indicam a sequência das atividades a serem executadas pelos servidores.

[1]https://falabr.cgu.gov.br/

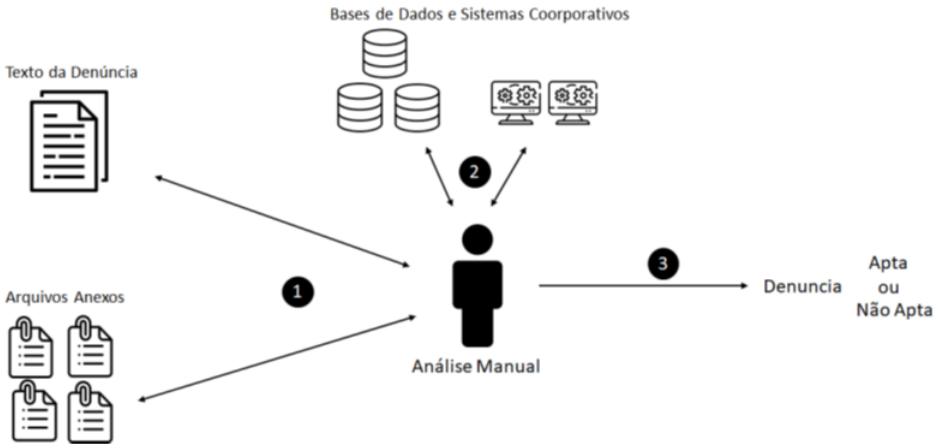

Figura 1.1: Fluxo de trabalho para análise de aptidão de denúncias

Conforme pode ser observado na Figura 1.1, ao analisar uma denúncia específica, o servidor deve ler o texto dessa denúncia, acessar e analisar cada arquivo anexo a essa denúncia. Esses arquivos anexos podem estar em diferentes formatos (planilhas, figuras, apresentações, arquivos textos e outros). Não há limite para o número de anexos de uma denúncia. A quantidade média de anexos nas denúncias com anexo é de aproximadamente 3, porém existem denúncias com até 50 arquivos anexos.

Uma vez analisados os textos e os anexos às denúncias, o servidor deve checar as informações relatadas nesses documentos em bases de dados e sistemas corporativos. A partir dessas análises, ele conclui sobre a aptidão ou não das denúncias.

Sendo assim, o problema que essa pesquisa se propõe a tratar é: como criar um modelo capaz de predizer, a partir dos textos das denúncias e de seus respectivos anexos, se essas denúncias devem ser consideradas como aptas ou não?

Logo, o problema em questão é caracterizado como um problema de classificação textual e pode ser definido da seguinte forma:

1. Tem-se uma coleção de textos e anexos de denúncias $D = \{d_1, d_2, d_3, \ldots, d_k\}$, em que cada elemento d_i representa o texto de uma denúncia concatenado com o conteúdo dos seus anexos

2. Tem-se um conjunto de situações $S = \{s_1, s_2, s_3, \ldots, s_k\}$, em que cada elemento s_i representa a situação da denúncia d_i, sendo que s_i só pode assumir dois valores 0 ou 1, de forma que 0 indica que a denúncia foi considerada Não Apta, e 1 indica que a referida denúncia foi considerada Apta.

3. Partindo-se do pressuposto de que exista uma função desconhecida f, capaz de mapear $d_i \rightarrow s_i$ tal que $f(d_i) = s_i$, deseja-se encontrar uma função de aproximação $\hat{f}$ que permita estimar o valor de f para novas denúncias d.

No entanto, esse problema de classificação tem alguns dificultadores. O primeiro

é que a classificação deve levar em consideração não apenas o conteúdo das denúncias, mas também o conteúdo dos arquivos anexos, que podem estar em diferentes formatos.

O outro dificultador reside no fato de que essa classificação não pode se limitar ao conteúdo dos textos. Ela tem que ser capaz de extrair certas informações dos textos e correlacioná-las com outras bases de dados, a fim de fornecer mais parâmetros para a tomada de decisão.

Também é preciso considerar que os textos das denúncias têm características bem peculiares, que os diferem de outros tipos de textos. As denúncias são escritas por cidadãos de diferentes graus de instrução e a forma desses textos pode variar bastante. Isso exige um estudo para identificar a abordagem mais apropriada para esse problema de classificação textual.

Outra característica das denúncias é o fato delas possuírem textos grandes, porém, com muitas informações que não são relevantes para o processo de classificação.

1.3 Principais Desafios

Diante das características do problema a ser tratado, foram elencados alguns desafios que precisaram ser abordados por essa pesquisa. Esses desafios são:

- Necessidade de pré-processamento dos arquivos anexos às denúncias;
- Necessidade da identificação da arquitetura de rede mais apropriada para a classificação das denúncias;
- Necessidade do desenvolvimento de uma técnica de sumarização de denúncias;
- Necessidade de informações externas aos textos das denúncias.

1.3.1 Necessidade de pré-processamento dos arquivos anexos

O primeiro dificultador para o processo em questão é o fato dos textos não estarem todos agrupados em um único ponto de consulta. As denúncias costumam vir acompanhadas de arquivos anexos, que podem estar em diferentes formatos. Sendo assim, o processo a ser desenvolvido precisa ser capaz de tratar toda essa diversidade de formatos, a fim de possibilitar um procedimento único para todo o conteúdo da denúncia.

Logo, o pré-processamento dos textos não pode se limitar a fatores associados ao Processamento de Linguagem Natural. Nessa situação, a leitura e o tratamento dos diversos tipos de arquivos também são necessários, a fim de tornar o conteúdo desses anexos passíveis de serem processados pelo modelo a ser desenvolvido.

1.3.2 Necessidade de identificação da arquitetura de rede mais apropriada para a classificação de denúncias

O tamanho dos textos a serem classificados, além de trazerem complicações com relação ao tempo de processamento das informações, também é um dificultador para o desenvolvimento do classificador. Isso acontece porque os principais modelos de processamento de linguagem natural, atualmente considerados como o estado da arte, possuem restrições para o tamanho de entrada dos textos.

Também é preciso considerar que os textos das denúncias têm características bem peculiares, que os diferem de outros tipos de textos. As denúncias são escritas por cidadãos de diferentes graus de instrução e a forma desses textos podem variar bastante.

Por essas razões, um dos desafios a serem abordados por essa pesquisa é a identificação da arquitetura de rede mais apropriada para os textos em questão.

1.3.3 Necessidade de desenvolvimento de uma técnica de sumarização de denúncias

Outro desafio a ser transposto é o fato de muitas denúncias serem compostas por conjuntos textuais grandes, que possuem informações não relevantes para a classificação, prejudicando assim o processo como um todo.

Diante dessa situação, levantou-se a hipótese de que se forem aplicadas técnicas de sumarização de denúncias antes de se gerar os modelos de classificação de aptidão de denúncias, então, esses modelos gerarão melhores resultado.

Dessa forma, faz-se necessário o desenvolvimento de técnicas de sumarização que possam testar essa hipótese.

1.3.4 Necessidade de informações externas aos textos das denúncias

O quarto desafio é a necessidade de utilizar informações externas aos textos para se proceder a classificação.

Essa necessidade ocorre porque quando os servidores da OGU realizam a análise de aptidão, eles consultam outros sistemas para acessar informações que não estão contidas nos textos. O objetivo dessas consultas é obter novos dados que possam auxiliar na decisão sobre a aptidão das denúncias, ou seja, são utilizadas outras informações (que estão em bases de dados auxiliares) com o objetivo de validar os fatos narrados e quantificar os potenciais danos oriundas das irregularidades citadas.

Dessa forma, a solução a ser desenvolvida não pode seguir a linha adotada pelos modelos de classificação textual comumente desenvolvidos, que se prendem ao

conteúdo dos textos analisados.

Logo, a solução proposta deve ser capaz de extrair um conjunto de informações dos textos das denúncias e utilizar essas informações para buscar novos elementos, em outras bases de dados, que possam auxiliar no processo de classificação dessas denúncias.

1.4 Questões de Pesquisa

Diante dos desafios apresentados (Seção 1.3), foram formuladas três questões de pesquisa a serem respondidas durante os estudos:

- **Questão 1:** Qual é a arquitetura de rede mais apropriada para o problema de classificação de aptidão de denúncias?
- **Questão 2:** Como extrair as principais informações dos textos das denúncias, a fim de utilizá-las no processo de classificação textual?
- **Questão 3:** como considerar informações externas aos textos das denúncias no processo de classificação de aptidão de denúncias?

1.5 Objetivos

Essa pesquisa se propõe a estudar e desenvolver técnicas capazes de endereçar o problema descrito na Seção 1.2, a fim de automatizar a classificação de aptidão das denúncias, e incorporar novas formas de abordagens para as dificuldades apresentadas. Sendo assim, os objetivos dessa pesquisa são:

Objetivo Geral: desenvolver um modelo de classificação textual capaz de prever se uma denúncia deve ser classificada como apta ou não, a partir do conteúdo das denúncias e de seus anexos.

Objetivos Específicos:

- Identificar a arquitetura de rede mais apropriada para a classificação de aptidão de denúncias.
- Desenvolver uma metodologia de sumarização de textos capaz de extrair as principais informações dos textos das denúncias.
- Avaliar se a utilização de textos sumarizados de denúncias, para a geração dos modelos, melhora o desempenho dos classificadores.

- Desenvolver uma metodologia de extração de informações capaz de identificar elementos nos textos das denúncias e buscar por mais informações em bases de dados auxiliares.

- Desenvolver um modelo de classificação textual que considere não só o conteúdo dos textos, mas também dados estruturados relacionados a esse conteúdo.

1.6 Descrição da Base de Dados

Os textos das denúncias possuem algumas características que precisam ser consideradas durante a pesquisa. Sendo assim, realizou-se uma análise descritiva em um conjunto de 4482 denúncias a fim de se identificar os fatores que deveriam ser levados em consideração durante os estudos.

Esses textos (textos originais concatenados com o texto dos anexos) apresentam uma grande variação de tamanhos. A menor denúncia possui apenas 1 palavra, e a maior denúncia possui mais de 2 milhões de palavras. No entanto, o tamanho médio das denúncias é de 10.680 palavras, e a mediana é de 247 palavras.

A grande distância entre a média e a mediana demonstra que há *outliers* no tamanho das denúncias, o que faz com que a média fique deslocada em relação a mediana. Essa mesma conclusão também pode ser alcançada a partir do histograma apresentado na Figura 1.2. Esse histograma apresenta o número de palavras por denúncias. O eixo horizontal representa "bins" de tamanho 250 e o eixo vertical representa o número de denúncias que possuem a quantidade de palavras que se enquadram no "bin" em questão. Sendo assim, a primeira coluna representa a quantidade de denúncias com número de palavras que varia de zero a 250, a segunda coluna representa a quantidade de denúncias que possuem de 251 a 500 palavras, e assim sucessivamente.

Cabe ressaltar que esse histograma só considera denúncias com até 2000 palavras por questões de legibilidade do gráfico.

Pela análise desse histograma, percebe-se que a maioria das denúncias possuem menos que 500 palavras, e que as denúncias maiores são menos frequentes.

A Figura 1.3 demonstra a divisão dos tamanhos das denúncias (em quantidade de palavras) por quartis e também evidencia essa disparidade no número de palavras por denúncias.

Apesar dos textos terem um valor mediano de 247 palavras, não se pode desconsiderar as denúncias maiores, pois elas representam uma grande quantidade de denúncias em valores absolutos. Por exemplo, analisando o 3º quartil (apresentado na Figura 1.3), identifica-se que 25% das denúncias possuem mais de 2048 palavras.

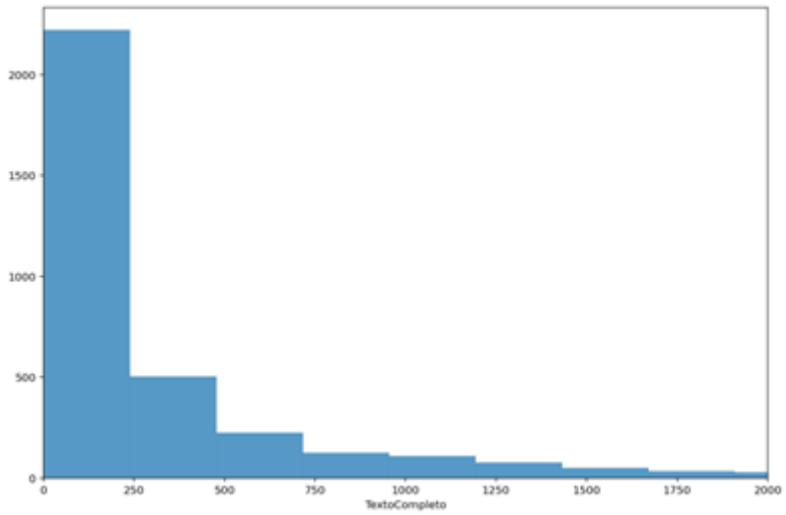

Figura 1.2: Histograma com o número de palavras por denúncia

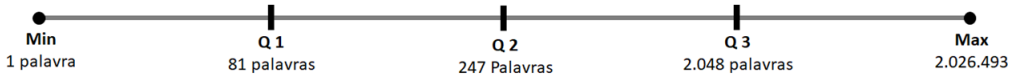

Figura 1.3: Quantidade de palavras por quartis

Logo, como a base em questão é composta por 4482 denúncias, isso corresponde a 1.120 denúncias.

Quanto aos rótulos (informação que diz se a denúncia deve ser considerada como apta ou não), foi feito um processo de rotulagem manual pelos servidores da OGU. Essa rotulagem estava integrada ao processo de trabalho desses servidores e obedecia as seguintes regras:

- Os rótulos devem ser valores numéricos entre 10 e 100, sendo que, esses valores devem ser múltiplos de 10 (10, 20, 30, ..., 100).
- Quanto maior for o valor do rótulo, maior é a confiança que o servidor tem de que a denúncia deve ser considerada como apta. Da mesma forma, quanto menor for o valor do rótulo, menor é a confiança que o servidor tem de que aquela denúncia deve ser considerada como apta.
- Os rótulos menores ou iguais a 50 caracterizam uma denúncia como não apta, e os rótulos maiores ou iguais a 60 caracterizam uma denúncia como apta.

Dessa forma, valores mais extremos caracterizam situações em que o servidor não teve dúvida para definir sua decisão, enquanto, valores centrais representam situações em que o servidor não teve muita convicção para classificar a denúncia.

Analisando-se os rótulos das denúncias, constatou-se que cerca de 70% das denúncias eram consideradas como não aptas, enquanto 30% eram consideradas como aptas.

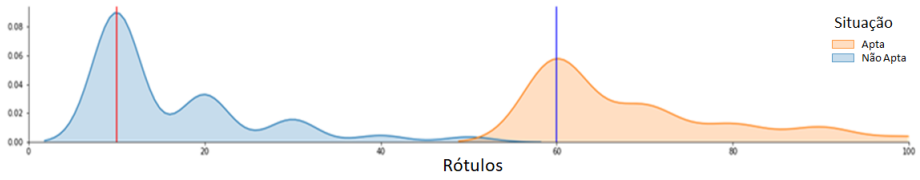

Figura 1.4: Gráfico de densidade por rótulos

A Figura 1.4 apresenta o gráfico de densidade da quantidade de denúncias por rótulo, sendo que, o eixo horizontal indica o grau de aptidão informado pelo servidor responsável pela rotulação. Conforme pode ser observado, as denúncias são classificadas como não aptas com alta confiança por parte dos servidores, pois a maior parte das denúncias não aptas receberam o valor 10.

No entanto, a maior parte das denúncias classificadas como aptas geraram dúvidas nos servidores que fizeram a classificação, pois, a maioria dessas denúncias foram classificadas com o menor valor possível para serem consideradas como aptas (60).

Esse fato evidencia um dificultador para o classificador a ser desenvolvido, uma vez que, a classificação apresenta um grau de subjetividade que gera dúvidas na própria classificação humana. Assim, provavelmente o algoritmo a ser desenvolvido também terá essa incerteza quanto a aptidão das denúncias.

Esse fator pode ser um indicativo de que ao invés de uma classificação binária, seria mais indicado o algoritmo fornecer a probabilidade da denúncia ser classificada como apta ou não apta. Isso permitiria que o modelo pudesse ser otimizado sem haver o estabelecimento de um ponto de corte, possibilitando assim, que a área de negócio decida qual seria o percentual mais apropriado para se classificar uma denúncia como apta ou não.

Esse fato também sugere que a métrica a ser utilizada para avaliar o modelo seja uma métrica baseada em probabilidades.

1.7 Métricas de avaliação utilizadas

Pelas razões expostas na Seção 1.6, os experimentos realizados nessa pesquisa foram guiados por duas métricas específicas: área sob a curva ROC (ROC-AUC) e Área sob a curva Precision-Recall (AUPRC).

A curva ROC (*Receiver operating characteristic*) mede o desempenho de classificadores binários para diferentes pontos de corte. A ROC é uma curva de probabili-

dade, e a área sob essa curva representa a capacidade do modelo distinguir as duas classes do problema em questão.

Já a ROC-AUC mede a área sob a curva ROC, ou seja, essa métrica mede a área abaixo da curva do gráfico entre a taxa de verdadeiros positivos (casos que foram corretamente classificados como sendo de uma determinada classe) e a taxa de falsos positivos (casos que foram erroneamente classificados como sendo de uma determinada classe).

Dessa forma, quanto maior o valor dessa métrica, melhor é a performance do modelo, sendo que, o seu valor máximo é um. Cabe ressaltar que, um modelo que gere uma classificação totalmente aleatória obtém um valor de 0,5 para a ROC-AUC. Logo, qualquer valor superior a 0,5 já representa uma classificação melhor do que uma escolha aleatória.

Já a Área sob a curva Precision-Recall (AUPRC) sumariza a curva de precisão-recall com a média ponderada alcançada em diferentes pontos de corte. Essa métrica também varia entre zero e um, sendo que quanto maior ela for, melhor é a performance do classificador.

Essa pesquisa utilizou a ROC-AUC como métrica principal e a AUPRC como métrica secundária. Portanto, apesar das análises serem guiadas pala métrica ROC-ACU, a AUPRC é utilizada como um segundo balizador, a fim de possibilitar uma outra visão para os resultados obtidos.

1.8 Organização da tese

O restante do texto dessa tese está organizado da seguinte forma: o Capítulo 2 apresenta uma fundamentação teórica sobre a utilização de técnicas de *deep learning* para o Processamento de Linguagem Natural. O Capítulo 3 mostra os estudos realizados para a identificação da arquitetura mais apropriada para a classificação textual de aptidão de denúncias. O Capítulo 4 expõe a proposta e a avaliação de métodos de sumarização de denúncias. Já o Capítulo 5 descreve a proposta de obtenção de variáveis estruturadas a partir de dados textuais e a utilização dessas variáveis para a criação de um modelo combinado de classificação textual. Finalmente o Capítulo 6 apresenta as conclusões da tese.

Capítulo 2

Deep Learning para o Processamento de Linguagem Natural

Esse capítulo apresenta uma visão sobre as principais técnicas de processamento de linguagem natural baseadas em *Deep Learning*.

Atualmente, a maioria dos dados disponíveis para análise se encontra em formato textual. Logo, essa é uma área de pesquisa que tem recebido bastante atenção nos últimos anos. As atividades de Processamento de Linguagem Natural podem ser utilizadas para a classificação textual [11], análise de sentimentos [12], tradução automática de textos [13], sumarização de textos [14], dentre outras.

Estudos demonstram evoluções nas formas de representação textual, e no modo de processamento dos textos. Quanto à representação textual, foram desenvolvidas técnicas capazes não só de representar os textos, como também de capturar informações semânticas e sintáticas das palavras [15], [16], [7], [4].

As pesquisas na área de processamento de textos têm apontado para a utilização de algoritmos de *Deep Learning* (redes neurais profundas). Nesse contexto, as redes *Convolutional Neural Network-CNN* [17], *Recurrent Neural Network-RNN* [18] e as baseadas na arquitetura Transformers [3] têm demonstrado bons resultados.

Sendo assim, esse trabalho faz um levantamento de tais técnicas a fim de fornecer um panorama sobre os principais direcionamentos utilizados no processamento de textos.

O restante desse Capítulo está dividido da seguinte forma: a Seção 2.1 apresenta um estudo sobre a representação de palavras. As Seções 2.2, 2.3 e 2.4 descrevem as redes neurais convolucionais, as redes neurais recorrentes e a arquitetura *Transformer*, respectivamente. Já a Seção 2.5 demonstra o funcionamento e a utilização da arquitetura *Bidirectional Encoder Representations from Transformers* (BERT) e a

Seção 2.6 apresenta outros modelos que tentam tratar limitações do modelo BERT. Finalmente, a Seção 2.7 faz uma breve conclusão do estudo.

2.1 Representação de Palavras

Os modelos de redes neurais não trabalham com o texto bruto, eles esperam como entrada vetores numéricos [19]. Logo, antes de se iniciar o processamento de textos com redes neurais, é necessário realizar um processo de vetorização textual.

Atualmente, a forma mais usual de se representar palavras como vetores numéricos é pela utilização de *word embeddings*. *Word embedding* é uma forma de representação textual que utiliza vetores densos[1], de baixa dimensionalidade, cujos valores são aprendidos a partir do próprio texto. O processo de *word embedding* utiliza a vizinhança de cada uma das palavras do texto para formular a representação das palavras. Isso permite a criação de um vetor denso que representa a projeção de cada palavra. Dessa forma, o *word embedding* representa as coordenadas da palavra em um espaço vetorial que foi aprendido a partir do texto. Sendo assim, os relacionamentos geométricos entre os vetores de palavras devem refletir os relacionamentos semânticos entre essas palavras [20]. Utilizando-se essa abordagem, pode-se comparar a relação entre duas palavras quaisquer a partir da comparação dos vetores que as representam.

Esse tipo de representação surgiu da necessidade de se expressar as características de similaridade entre as palavras, de forma que isso pudesse ser aproveitado no contexto das aplicações. Sendo assim, passou-se a explorar o conceito de modelagem estatística da linguagem. Esse modelo permite a previsão da palavra seguinte, levando-se em consideração as anteriores [21]. Logo, pode-se prever a palavra seguinte de um determinado texto com base nas palavras que ocorreram anteriormente nesse mesmo texto.

Essa constatação já havia sido feita por linguistas. HARRIS [22] já afirmava que as palavras vizinhas estavam relacionadas semanticamente, logo, palavras semelhantes ocorreriam em contextos semelhantes. Nesse mesmo sentido, FIRTH [23] afirma que é possível conhecer uma palavra através das palavras que a acompanham (*"You shall know a word by the company it keeps!"*). Dessa forma, conclui-se que as palavras não ocorrem em contextos independentes, uma palavra sempre está relacionada com as palavras que vêm antes e depois dela.

Sendo assim, pode-se destacar duas características importantes do modelo de linguagem:

[1]Um vetor denso é um vetor cujas posições são preenchidas com valores diferentes de zero. Em contrapartida, um vetor esparso é aquele cuja maioria das posições são preenchidas com o valor zero.

- A probabilidade de se encontrar uma palavra em um determinado texto é uma função da ocorrência de todas as palavras anteriores;
- Uma palavra sempre está relacionada com as suas vizinhanças.

Logo, a partir dessas premissas, formulou-se a seguinte expressão, indicada na Equação 2.1, para representar a probabilidade de ocorrência de uma determinada palavra em uma sequência de dados textuais. Desse modo, para se obter a probabilidade de ocorrência de uma palavra localizada na posição t, deve-se considerar a probabilidade de ocorrência de todas as palavras que apareceram nas $t-1$ posições anteriores (razão pela qual o valor de K, apresentado na Equação 2.1, varia entre 1 e $t-1$).

$$p(w^{(t)}) = \prod_{k=1}^{k=t-1} p(w^{(k)}|\{w^1, ..., w^{k-1}\}) \qquad [21] \qquad (2.1)$$

Logo, a probabilidade de ocorrência de uma determinada palavra em um texto será igual ao produto das probabilidades de ocorrência das palavras que ocorreram antes dela nesse mesmo texto.

No entanto, essa é apenas uma formulação teórica, pois na maioria das situações não é possível utilizar todas as palavras anteriores (desde o início do texto) para se prever a próxima. Então, para tornar essa premissa viável de ser aplicada em situações práticas, faz-se uma aproximação, e em vez de se considerar todas as palavras para o cálculo da probabilidade da palavra seguinte, realiza-se esse cálculo com base em uma janela de tamanho fixo. Assim, calcula-se a probabilidade de uma palavra com base nas n palavras anteriores (onde n é o tamanho da janela a ser considerada). Dessa forma, a Equação 2.1 pode ser simplificada para a Equação 2.2:

$$p(w^{(t)}) \approx \prod_{k=t-n+1}^{k=t-1} p(w^{(k)}|\{w^{k-1}, ..., w^{k-n}\}) \qquad [21] \qquad (2.2)$$

Isso permite a modelagem de qualquer palavra do texto em função de um número fixo de parâmetros, o que torna tal representação muito mais concisa. A partir dessa consideração, torna-se possível desenvolver uma série de modelos de *word embeddings* que são utilizados para a modelagem textual. Existem vários modelos de *word embeddings* que se utilizam dessa premissa. Esse estudo apresenta dois exemplos desses tipos de modelos: o word2Vec [15] e [16] e o GloVe [7].

2.1.1 Word2Vec

Apesar de algumas tentativas anteriores, as primeiras iniciativas que apresentaram resultados satisfatórios, para a representação de palavras como vetores densos, foram

as propostas em [15] e [16]. Esses trabalhos apresentam a representação distribuída de palavras obtida a partir da utilização de uma rede neural de duas camadas. Tal modelo ficou conhecido como o Word2Vec, dividindo-se em duas variantes: *skip-gram* e *Continuous Bag of Words* (CBOW).

- **Skip-gram**

O modelo *skip-gram* [16] é a variante do word2vec que prevê o contexto baseado na palavra corrente, ou seja, dada uma palavra de entrada, ele tenta prever as palavras que aparecem antes e depois dessa palavra de entrada. Para isso, esse modelo se utiliza de uma rede neural de duas camadas que é treinada com um determinado corpus, ou seja, o próprio texto já oferece a entrada (uma palavra central) e os valores de referência que devem ser utilizados no processo de treinamento (palavras vizinhas), constituindo-se assim em um processo auto-supervisionado. Logo, a entrada dessa rede neural é a representação da palavra central, e a saída são as as representações das palavras vizinhas, que se pretende prever. Nessa situação, o processo de *word embedding* é representado pelos pesos da camada escondida da rede neural.

A Figura 2.1 ilustra a estrutura da rede neural utilizada para encontrar a representação das palavras empregando a variante skip-gram do modelo word2vec, sendo que, nesse caso, V é o tamanho do vocabulário[2], N é a quantidade de neurônios da camada escondida, x e y são vetores *one-hot encoding*[3] que representam as entradas e saídas do modelo, respectivamente, W_{VN} é a matriz de pesos entre a camada de entrada e a camada escondida, onde a *iésima* linha da matriz representa o peso correspondente da *iésima* palavra do vocabulário, a matriz W'_{VN} representa os pesos entre a camada escondida e a camada de saída e C está relacionado ao tamanho do contexto (quantidade de vizinhos da palavra alvo). Os vetores da *word embedings* estão contidos na matriz de pesos W_{VN}.

Sendo assim, considerando a frase *"esse é um exemplo de sentença"*, dada a palavra central *"exemplo"*, o modelo tenta prever as palavras do contexto: *"esse"*, *"é"*, *"um"*, *"de"* e *"sentença"*.

O algoritmo funciona da seguinte forma: primeiro gera-se o vetor *one-hot encoding* da palavra central, depois obtém-se o vetor de pesos correspondente à palavra em questão (*word embeding* dessa palavra central). O passo seguinte é a geração do *score* z, sendo que esse *score* é obtido pela multiplicação do vetor de entrada pelo vetor de pesos (como a entrada é um vetor esparso, *one-hot encoding*, quando é feita a multiplicação da entrada pela matriz de pesos, apenas a linha correspondente à palavra de entrada em questão é considerada). Posteriormente, é feito o

[2]Vocabulário: conjunto de palavras (ou tokens) que devem ser consideradas no processamento do texto.

[3]Vetor *one-hot encoding* é um vetor em que todas as suas posições são preenchidas com o valor zero, exceto a posição da palavra que ele está representando, que é preenchida com o valor 1.

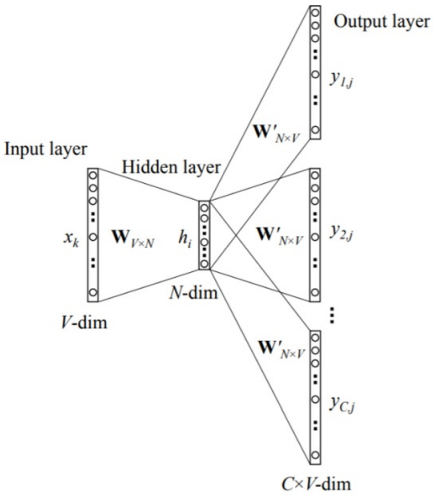

Figura 2.1: Estrutura de Funcionamento do word2vec - skip-gram [1].

cálculo da probabilidade das palavras de saída, para isso, utiliza-se a função *softmax*[4] ($y = softmax(z)$). Depois, compara-se o vetor de probabilidades geradas com o vetor de probabilidades reais. Isso possibilita a atualização dos pesos pelo método do gradiente e repete-se esse processo de forma iterativa até o fim do treinamento.

A probabilidade, obtida pela aplicação da função softmax, é dada pela Equação 2.3. Nessa Equação, v_O indica a representação da palavra de saída ($w^{(t)}$) e μ_I é a representação da palavra de entrada ($w^{(t+k)}$).

$$p(w^{(t+k)}|w^{(t)}) = p(v_O|\mu_I) = \frac{exp(v_O^T \mu_I)}{\sum_{j=1}^{M} exp(v_J^T \mu_I)} \qquad [15] \qquad (2.3)$$

Logo, obtêm-se a probabilidade de uma determinada palavra $(t + k)$, a partir de uma palavra t, pela utilização da função softmax aplicada ao produto de dois vetores u e v.

- ***Continuous Bag of Words* (CBOW)**

O modelo CBOW [16] funciona de forma análoga ao Skip-gram. No entanto, nessa versão do Word2vec a representação da palavra é obtida pelo ajuste dos parâmetros do modelo que tenta fazer a previsão de uma determinada palavra a partir do seu contexto (palavras que aparecem na sua vizinhança), conforme apresentado na Figura 2.2.

Sendo assim, considerando a frase *"esse é um exemplo de sentença"*, dadas as palavras do contexto: *"esse"*, *"é"*, *"um"* e *"de"* e *"sentença"*, o modelo tenta prever a palavra central *"exemplo"*.

[4]A função *softmax* recebe como entrada um vetor de valores reais e fornece como saída um vetor com valores entre 0 e 1 e a soma desses valores é igual a 1.

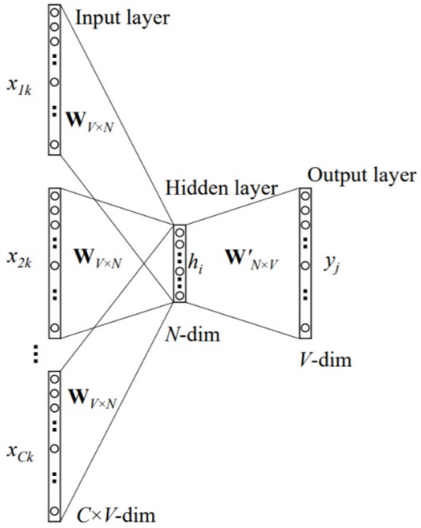

Figura 2.2: Estrutura de Funcionamento do Word2Vec – CBOW. [1].

Nesse caso, as entradas são vetores *one-hot encoding*, que representam cada uma das palavras de contexto, e a saída é o vetor *one-hot encoding* que faz a previsão da palavra central. Cabe ressaltar que o algoritmo do CBOW funciona de forma análoga ao Skip-gram, sendo que, a única diferença é que no caso do CBOW as entradas e saídas da rede neural são invertidas, ou seja, as palavras de contexto funcionam como entrada da rede neural e a palavra central é a saída da rede.

2.1.2 GloVe

O modelo Global Vectors for Word Representation - GloVe [7] trabalha com as probabilidades de coocorrência de palavras em um texto para incorporá-las em vetores significativos. Assim, ele considera a frequência em que uma palavra j aparece no contexto de uma palavra i em todo o texto. Dessa forma, considera-se X a matriz de coocorrência, cujo elemento X_{ij} é o número de vezes que a palavra j aparece no contexto da palavra i.

A probabilidade de coocorrência de uma palavra j ocorrer com uma palavra i é a razão entre o número de vezes que a palavra j ocorre no contexto da palavra i sobre o número de vezes que qualquer palavra aparece no contexto da palavra i, conforme apresentada na Equação 2.4.

$$P_{ij} = P(j|i) = \frac{X_{ij}}{\sum_{\ i\ \in\ context} X_{ik}} \qquad [7] \qquad (2.4)$$

Dessa forma, o GloVe verifica a razão entre as probabilidades de coocorrências para extrair o significado interno das palavras. PENNINGTON *et al.* [7] utilizam a Tabela 2.1 para exemplificar essa ideia.

Probabilidade e Razão	k = solid	k = gas	k = water	k = fashion
P(k/ice)	$1,9 \times 10^{-4}$	$6,6 \times 10^{-5}$	$3,0 \times 10^{-3}$	$1,7 \times 10^{-5}$
P(k/steam)	$2,2 \times 10^{-5}$	$7,8 \times 10^{-4}$	$2,2 \times 10^{-3}$	$1,8 \times 10^{-5}$
P(k/ice)/P(k/steam)	$8,9$	$8,5 \times 10^{-2}$	$1,36$	$0,96$

Tabela 2.1: Probabilidades de Coocorrências [7]

PENNINGTON *et al.* [7] explicam que as duas primeiras linhas da tabela mostram as probabilidades das palavras *"solid"*, *"gas"*, *"water"* e *"fashion"* ocorrerem no contexto das palavras *"ice"* e *"steam"*. Já a última linha mostra a razão de probabilidades.

Para palavras relacionadas a *"ice"*, mas não a *"steam"*, como *"solid"*, a razão é alta. Por outro lado, para palavras relacionadas a *"steam"*, mas não a *"ice"*, como *"gas"*, a razão é baixa e, para palavras relacionadas a ambos ou a nenhum deles, como *"water"* e *"fashion"*, respectivamente, a razão é próxima de 1.

Dado que as razões de probabilidades de coocorrência capturam informações relevantes sobre o relacionamento das palavras, o modelo GloVe visa obter uma função F que prevê essas proporções a partir de dois vetores de palavras w_i e w_j e de um vetor de palavra de contexto w_k, conforme apresentado na Equação 2.5.

$$F(w_i, w_j, \tilde{w_k}) = \frac{P_{ik}}{P_{jk}} \qquad [7] \tag{2.5}$$

Sendo assim, o modelo aprende as representações dos vetores de palavras w_i, w_j e w_k para alimentar a função F que busca prever corretamente as proporções de probabilidades.

O GloVe prediz as palavras circundantes maximizando a probabilidade de uma palavra de contexto ocorrer, a partir de uma palavra central, utilizando para isso uma regressão logística.

2.2 *Convolution Neural Network*

As redes neurais convolucionais (CNNs) são largamente empregadas em atividades de visão computacional, obtendo excelentes resultados em tais tarefas [24], [25]. Esse sucesso se dá principalmente pelo fato do seu processamento ser realizado de forma convolucional[5], o que é muito eficaz para casos em que os dados tenham alguma relação topológica com os seus vizinhos, como ocorre em imagens. Essa propriedade permite a extração de características que são capazes de definir os dados que são utilizados como entrada.

[5]A convolução é uma operação matemática entre duas funções g e h, produzindo uma terceira função que normalmente é vista como uma versão modificada de uma das funções originais[26].

Tal propriedade, que torna as CNNs tão eficazes para tarefas de visão computacional, também é útil para a extração de características importantes em dados sequenciais, pois, via de regra, os dados pertencentes a uma sequência textual também compartilham características com seus elementos vizinhos. Um texto pode ser visto como uma sequência de palavras. Logo, assim como em uma imagem um pixel compartilha algumas características em comum com os seus pixels vizinhos, as palavras de um texto também guardam algum tipo de relação com as demais palavras que aparecem nas suas proximidades. Sendo assim, pode-se utilizar esse tipo de rede para treinar modelos capazes de desempenhar tarefas de processamento de linguagem natural, conforme pode ser observado em [27], [28] e [29].

No entanto, enquanto nas tarefas de visão computacional os dados são tratados de forma bidimensional (largura e altura da imagem), nas tarefas de Processamento de Linguagem Natural com CNNs, os dados aparecem de forma unidimensional, onde a única dimensão tratada é a dimensão temporal. Logo, cada palavra do texto é considerada como o valor correspondente a um determinado instante de tempo. Sendo assim, para o caso de processamento de textos, ao invés de se utilizar convoluções 2D (como é o caso do processamento de imagens), utiliza-se convolução 1D.

A Figura 2.3 ilustra o processo de convolução 1D para uma determinada sequência de *tokens*. A convolução começa capturando os primeiros *tokens* da sequência da entrada, considerando-se o tamanho do *kernel* do filtro. A partir dessa captura, faz-se a multiplicação desses valores pelos filtros e obtém-se a saída correspondente a essa parte da entrada, sendo que esse processo se repete até que se chegue ao final da sequência de entrada. A CNN unidimensional é invariante para translações (assim como a bidimensional), o que significa que certas sequências podem ser reconhecidas mesmo que apareçam em posições diferentes. Isso pode ser útil para a identificação de determinados padrões no texto.

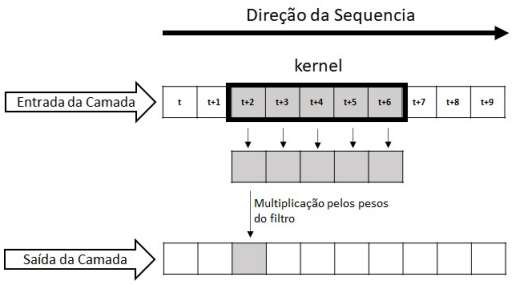

Figura 2.3: Processo de Convolução 1D.

Essa capacidade das camadas de convolução 1D reconhecerem padrões locais em uma sequência se dá pelo fato de que a mesma transformação de entrada é executada em cada janela. Logo, o padrão aprendido em uma determinada posição de uma

sentença pode ser reconhecido posteriormente em uma posição diferente. Por essa razão, diz-se que a convolução é invariante a translações. Dessa forma, caso se tenha um filtro sensível[6] a uma determinada sequência de palavras, ele será ativado sempre que tal sequência aparecer no texto, independentemente de sua posição dentro da frase.

Associada à camada de convolução, utiliza-se uma camada de *pooling*, cujo objetivo é diminuir a dimensionalidade dos dados de entrada e tornar o modelo pouco sensível a operações de rotação e de translação, fazendo uma espécie de subamostragem. A operação de *pooling* tem algumas variantes no critério de subamostragem, como por exemplo, *pooling* pela média ou *pooling* pelo valor máximo.

Sendo assim, as camadas de convolução e de *pooling* trabalham de forma combinada, com o objetivo de extrair um conjunto de características capazes de identificar algumas propriedades do conjunto de dados de entrada.

2.3 *Long Short-Term Memory Networks*

Uma das principais características das redes neurais convencionais (*feedforward*) é que elas não têm memória. Cada entrada apresentada a elas é processada independentemente, sem que nenhum estado seja mantido entre as entradas. Dessa forma, para se processar uma sequência de dados, é necessário mostrar a sequência inteira à rede de uma só vez, transformando-a em um único ponto de dado.

No entanto, muitas vezes é importante se ter alguma informação a respeito do que foi processado nos instantes anteriores, para utilizá-la no processamento atual. Essa característica é especialmente importante para o processamento de textos, visto que, muitas das vezes, o significado de uma determinada palavra depende das palavras que foram processadas anteriormente.

As redes neurais recorrentes (*recurrent neural network* - RNN) [18] surgiram com o objetivo de preencher essa lacuna. Esse tipo de rede neural processa sequências de dados e mantêm as informações referentes a esse processamento. Nas RNNs há uma espécie de realimentação a partir da sua saída, o que permite com que as variáveis de estado possam ser mantidas para auxiliar nas próximas previsões.

No entanto, as redes neurais recorrentes tradicionais apresentam o problema da dissipação do gradiente (*vanish gradient*) e da explosão do gradiente [30]. Esse tipo de problema dificulta o processo de treinamento da rede, pois, durante a fase de retropropagação do gradiente, o sinal do gradiente acaba sendo multiplicado várias vezes pela matriz de peso associada às conexões entre os neurônios da camada recorrente. Isso faz com que o valor dessas multiplicações tenda para zero ou para

[6]Nesse contexto, entende-se como um filtro sensível, aquele que é capaz de capturar algum tipo específico de padrão nos dados que estão sendo processados.

valores muito altos, o que, em ambos os casos, é prejudicial para o processo de treinamento da rede.

PASCANU *et al.* [30] demonstram esse problema. Basicamente, se a matriz for formada por valores pequenos (se o autovalor principal da matriz de peso for menor que 1), isso fará com que o valor do gradiente fique tão pequeno a ponto de paralisar o processo de treinamento. Esse fenômeno é conhecido como dissipação do gradiente (ou *vanishing gradient*). Por outro lado, se os pesos dessa matriz forem altos (autovalor principal da matriz de peso maior que 1), o sinal de gradiente pode ficar tão grande a ponto de fazer com que o processo de treinamento não convirja. Esse fenômeno é conhecido como explosão do gradiente (ou *exploding gradient*).

Essas limitações motivaram o surgimento das redes LSTM [31]. Os neurônios de uma camada LSTM possuem uma célula de memória que é composta por quatro elementos principais: um portão de entrada, um portão com uma conexão auto recorrente, um portão de esquecimento e um portão de saída. A conexão auto recorrente tem um peso de 1 e permite que o estado da célula de memória permaneça constante de um passo para outro. Os portões servem para regular as interações entre a própria célula de memória. A porta de entrada pode permitir que o sinal recebido altere o estado da célula de memória ou bloqueie-o. Já a porta de saída pode permitir que o estado da célula de memória tenha efeito sobre outros neurônios ou não. O portão de esquecimento trata a conexão recorrente da célula de memória, permitindo que a célula se lembre ou esqueça do seu estado anterior, conforme necessário. A Figura 2.4 ilustra uma célula LSTM.

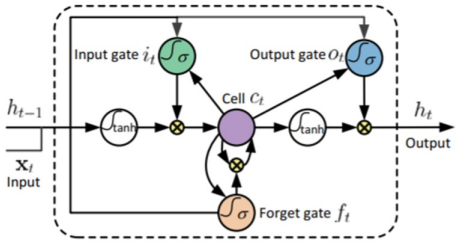

Figura 2.4: Neurônio LSTM [2].

Esse tipo de rede não sofre com os problemas da degradação e explosão do gradiente pelo fato de se gerar um fluxo contínuo para o cálculo do gradiente, pois, o cálculo do gradiente em relação ao espaço interno de memória não é retropropagado para esses parâmetros. Dessa forma, o erro retropropagado passa direto pela camada, não desaparecendo, nem explodindo, para sequências mais longas. Essa característica faz com que essa topologia melhore os resultados quando se precisa de memórias mais longas[7].

[7]Entende-se como necessidade de memórias mais longas as situações em que o texto processado é

Por tais características, as redes LSTMs vêm sendo largamente utilizadas nas tarefas de processamento de linguagem natural, pois a topologia da rede LSTM permite que sejam guardadas algumas informações referentes as palavras anteriores, sem sofrer os problemas de treinamento das RNNs tradicionais, o que pode ser útil para o processamento da palavra atual.

2.4 Arquitetura *Transformer*

Outra arquitetura de rede que tem ganhado relevância no contexto de processamento de linguagem natural é a arquitetura *Transformer*, que é composta por duas partes: um codificador e um decodificador. Porém, o conceito de codificadores e decodificadores é anterior à arquitetura *transformer*, um exemplo disso é a arquitetura proposta em [32]. CHO *et al.* [32] propõem duas redes neurais recorrentes (RNN) que atuam como um codificador e um decodificador.

O codificador mapeia uma sequência fonte de comprimento variável para um vetor de comprimento fixo e o decodificador mapeia essa representação vetorial de volta para uma sequência alvo de comprimento variável.

Na arquitetura proposta por CHO *et al.* [32], as duas redes (codificador e decodificador) são treinadas em conjunto para maximizar a probabilidade condicional da obtenção da sequência de destino, dada uma sequência de origem.

Nessa arquitetura, o codificador é responsável por receber sequências textuais e o decodificador utiliza a saída desse codificador para produzir respostas para os textos de entrada. Esse princípio é útil para problemas de tradução automática de textos, pois, nesse tipo de problema, deseja-se mapear uma sequência de palavras (que forma uma sentença) em uma outra sequência de palavras que forma uma sentença com a mesma carga semântica da sequência de entrada, mas escrita em outro idioma.

Utilizando esse princípio de codificadores e decodificadores, VASWANI *et al.* [3] propõem uma nova arquitetura, denominada *Transformers*, que é centrada em mecanismos de atenção. A Figura 2.5 apresenta a arquitetura *Transformers*, sendo que, a parte esquerda representa o codificador e a parte direita representa o decodificador.

A arquitetura apresentada na Figura 2.5 recebe como entrada, na parte inicial do codificador, uma frase inteira, e fornece como saída, na parte final do decodificador a probabilidade das próximas palavras a serem preditas.

Cabe ressaltar que as saídas do decodificador também são utilizadas para retroalimentar o processo, funcionando também como entrada na parte inicial do decodificador. Dessa forma, o decodificador também recebe como entrada uma frase

muito grande e as células de memória precisam armazenar muitas informações relativas às palavras anteriores.

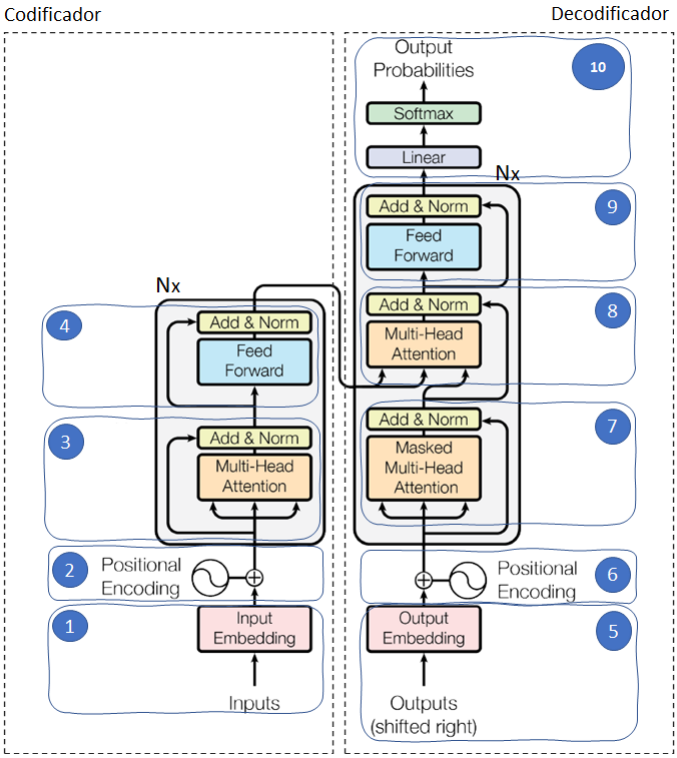

Figura 2.5: Arquitetura Transformer, adaptado de [3].

inteira. Assim, a frase é passada para o codificador, faz-se uma série de processamentos e depois vai para o decodificador. Então, o decodificador faz as previsões, que retornam as probabilidades de cada uma das palavras. Essas probabilidades são utilizadas para realimentar a entrada do decodificador. Portanto, as palavras previstas são utilizadas como entrada para novas previsões.

As próximas subseções detalham o funcionamento de cada um dos componentes da arquitetura *transformer*.

2.4.1 *Input Embedding*

A subcamada *Input Embedding* é a camada de entrada do codificador e tem a função de receber as palavras de entrada e convertê-las em vetores. Essa subcamada é representada pela estrutura marcada com o número 1 na Figura 2.5.

2.4.2 *Positional Encoding*

O *Positional Encoding* é utilizado para marcar a posição das palavras dentro das frases. As arquiteturas baseadas em convoluções (CNNs) ou em recorrências (RNNs) não precisam desse tipo de recurso. No caso das convoluções, as palavras sequenciais ficam juntas no mesmo filtro, ou seja, o posicionamento das palavras é mantido. As RNNs também mantêm a ordem das palavras, de acordo com a ordem de entrada

dessas palavras. Sendo assim, essas duas arquiteturas permitem a manutenção da ordem sequencial das palavras.

Já na arquitetura *Transformer*, faz-se necessário um marcador de posição, que aplica uma fórmula matemática, na matriz de *embbeding*, para que seja possível a identificação da posição de cada palavra no texto.

VASWANI *et al.* [3] resolvem essa questão utilizando as funções seno e cosseno para gerar diferentes frequências para a codificação posicional (PE). O *Positional Encoding*, representado pela estrutura de número 2 na arquitetura apresentada na Figura 2.5, cria outra matriz de *embbeding* adicionando um valor posicional, que marca a ordem das palavras. As Equações 2.6 e 2.7 indicam como se dá a obtenção do *positional encoding*.

$$PE_{(pos,\ 2i)} = \sin\left(\frac{Pos}{100000^{2i/d_{model}}}\right) \qquad [3] \tag{2.6}$$

$$PE_{(pos,\ 2i+1)} = \cos\left(\frac{Pos}{100000^{2i/d_{model}}}\right) \qquad [3] \tag{2.7}$$

Como pode ser observado, para as posições pares ($2i$) utiliza-se a função seno (equação 2.6), e para as posições ímpares ($2i+1$) utiliza-se a função cosseno (equação 2.7).

Logo, a função dessa subcamada é adicionar um valor de codificação posicional ao *embedding* de entrada, a fim de possibilitar a recuperação da posição da palavra dentro da frase.

2.4.3 *Multi-Head Attention*

As subcamadas *Multi-Head Attention*, representadas pelos números 3 e 8 na Figura 2.5, recebem como entrada um vetor de *embedding* que é dividido em três fluxos iguais. Logo, essa camada recebe três entradas, sendo que essas entradas são idênticas.

Esse componente é responsável por implementar o mecanismo de atenção. O mecanismo de atenção é um recurso que permite que uma determinada região do texto receba um "foco" maior, fazendo com que as demais informações fiquem "desfocadas". VASWANI *et al.* [3] definem esse tipo de mecanismo como uma função que realiza o mapeamento de uma *Query* (Q) e um conjunto de pares de chave-valor (K, V) para uma saída. A *Query* (Q), as chaves (K), os valores (V) e a saída são matrizes formadas pelos vetores das palavras da frase de entrada[8]. A saída é calculada por uma soma ponderada dos vetores de chave, sendo que, o peso atribuído a

[8]Cabe ressaltar que essas matrizes só serão a representação da frase de entrada no caso da primeira camada. Nas demais camadas, essas matrizes serão a saída da camada imediatamente anterior.

cada valor é calculado por uma função de compatibilidade da *Query* com a chave correspondente.

Essa atividade é executada por uma estrutura denominada *Scaled dot-product attention*, cuja funcionalidade é descrita abaixo

- ***Scaled dot-product attention***

O *scaled dot-product attention* trata a relação da frase original com suas próprias palavras. Sendo assim, esse mecanismo utiliza duas sentenças iguais A e B, e calcula como cada elemento de A está relacionado a cada elemento de B. Após isso, recombina-se A de acordo com essa relação. Portanto, esse tipo de mecanismo identifica palavras relacionadas dentro de uma mesma frase.

Sendo assim, a partir de uma sequência A (com um contexto B), obtém-se uma nova sequência na qual cada elemento é uma mistura dos elementos de A que estão relacionados com B. Assim, tem-se uma matriz que faz a combinação da frase com ela mesma.

Dessa forma, um produto escalar indica a similaridade entre dois vetores (no caso duas palavras). Assim, considerando u e v dois vetores de *embeddings* que representam duas palavras em uma mesma frase, quanto mais distante u está de v, menor similaridade entre essas palavras. Da mesma forma, quanto mais próximos estes vetores estiverem, maior a similaridade entre as palavras. Logo, a partir do produto escalar, realizado para todos os pares de palavras da frase, tem-se a relação existente entre cada par de palavra dessa frase.

A Equação 2.8 representa a operação executada pelo *scaled dot-product attention*, sendo que, Q, K e V são iguais e representam a frase de entrada[9].

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad [3] \qquad (2.8)$$

A Figura 2.6 representa graficamente a Equação 2.8. Nesse mecanismo, primeiro faz-se a multiplicação de Q e K (QK^T, sendo que K^T representa a transposta da matriz K). No denominador da função *softmax*, d_k indica a dimensão do *embbeding* que representa os tokens. Essa divisão é utilizada para se alterar a escala do resultado da multiplicação das matrizes. Posteriormente, aplica-se a função *softmax* (que fornece um valor entre 0 e 1). Por fim, faz-se a multiplicação pela matriz V.

Na Equação 2.8, a multiplicação de Q por K indica a autocorrelação das palavras da frase (pois, Q e K representam uma mesma frase). Logo, essa multiplicação retorna uma matriz que indica como as palavras da frase estão relacionadas entre si. O retorno da multiplicação QK é uma matriz, que após submetida a função *softmax*, é multiplicada por V e origina uma nova matriz.

[9]Cabe ressaltar que Q, K e V só representam a frase de entrada na primeira camada, nas demais camadas, essas matrizes representaram a saída da camada anterior.

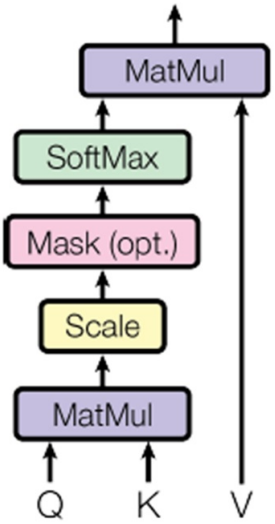

Figura 2.6: *Scaled dot-product attention* [3].

Para exemplificar a forma como o *scaled dot-product attention* funciona, utilizou-se frase: "eu gosto de café". Considerando que cada uma das palavras está sendo representada por um vetor de 5 posições, os vetores das palavras podem ser ilustrados conforme disposto na Figura 2.7(a). Já a frase completa é representada pela junção desses vetores, conforme apresentado na Figura 2.7(b). Desse modo, a frase será representada por uma matriz com 4 linhas (cada uma referente a uma das palavras da frase) e com 5 colunas (cada coluna referente a uma das dimensões utilizadas para representar as palavras). Cabe ressaltar que as 3 entradas do *scaled dot-product attention* são cópias dessa matriz que representa a frase inteira e essas entradas são chamadas de Q, K e V.

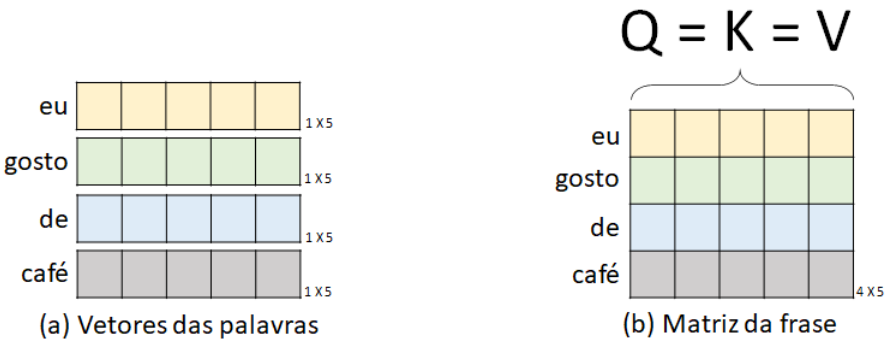

Figura 2.7: Representação da Frase.

A primeira parte do *scaled dot-product attention* faz a multiplicação entre as matrizes Q e K^T. Nessa multiplicação, cada linha da matriz Q é associada com uma coluna da matriz K^T. Assim, cada vetor que representa uma palavra na matriz Q é multiplicado por um vetor que representa uma palavra na matriz K^T, sendo que, o

resultado dessa multiplicação expressa o grau de associação de cada uma das palavras com todas as demais palavras que compõem a frase de entrada. Dessa forma, ao final, tem-se como resultado uma matriz quadrada, cuja dimensão coincide com o número de palavras da frase. Essa matriz de saída fornece o auto relacionamento das palavras de uma mesma frase. Assim, torna-se possível identificar palavras que possuem ligações maiores e sentidos semelhantes. A Figura 2.8 ilustra esse processo.

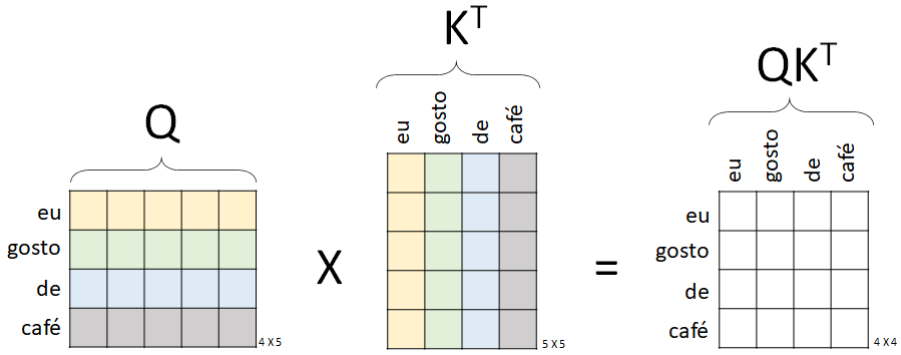

Figura 2.8: Cálculo do relacionamento entre as palavras.

O resultado da multiplicação é dividido pela raiz da dimensão e aplica-se a função $softmax$, fazendo assim, com que todos os valores fiquem compreendidos entre 0 e 1. O passo final é a multiplicação do resultado da função $softmax$ com a matriz V. Ou seja, a saída do *scaled dot-product attention* é uma matriz com as mesmas dimensões da matriz de entrada. A Figura 2.9 ilustra esse processo.

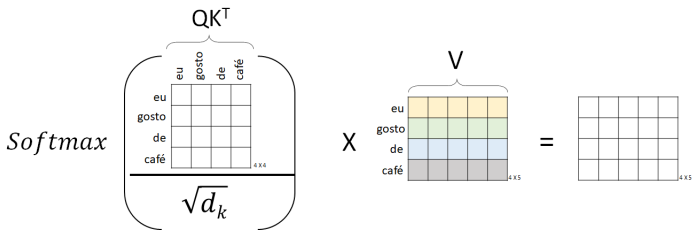

Figura 2.9: Saída do *scaled dot-product attention*.

- **Projeções Lineares no *Multi-Head attention Layer***

VASWANI *et al.* [3] identificaram que, ao invés de executar uma única função de atenção com todo o texto de entrada, seria melhor trabalhar com projeções lineares dessa entrada. Sendo assim, foi proposta a utilização de um número h de projeções (que são aplicadas nos textos de entrada), conforme demonstrado na Figura 2.10.

O conceito de projeções lineares permite a aplicação de mecanismos de atenção em diferentes subespaços. Assim, diferentes projeções lineares possibilitam o processamento de informações de diferentes subespaços em diferentes posições.

Dessa forma, antes de se aplicar o produto escalar, realiza-se a divisão das entradas em subespaços. O objetivo dessa divisão é reduzir a dimensionalidade dos vetores envolvidos, simplificando o processo de treinamento da rede.

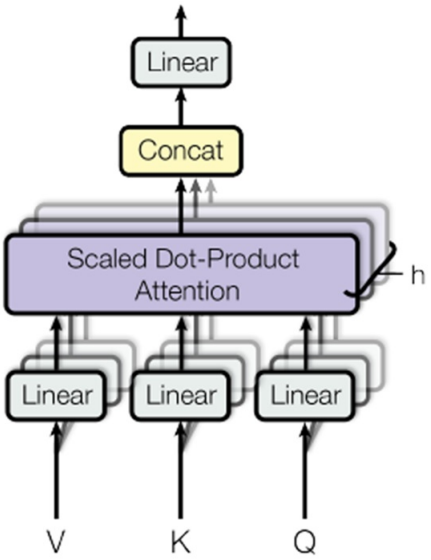

Figura 2.10: Projeções Lineares no *Multi-Head attention Layer* [3].

2.4.4 *Feedforward Network*

As camadas *Feed Forward*, representadas pelo número 4 na Figura 2.5, são responsáveis pela aplicação de duas transformações (duas camadas densas), conforme pode ser observado na Equação 2.9.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \qquad [3] \qquad (2.9)$$

2.4.5 Camada de Normalização (*Add & Norm*)

Cada subcamada de atenção e cada subcamada *Feed Forward* do *Transformer* é seguida por uma subcamada de normalização (*Add & Norm*). Essas subcamadas aparecem após várias estruturas da arquitetura *Transformers*, como pode ser observado nas estruturas representadas pelos números 3, 4, 7, 8 e 9 na Figura 2.5.

O objetivo da camada *Add & Norm* é evitar o processo de dissipação do gradiente, o que ajuda durante o processo de *backpropagation*.

Como pode ser observado na arquitetura, a camada *Add & Norm* sempre recebe uma cópia das frases antes delas entrarem nos mecanismos de atenção (ou nas camadas *Feed Forward*). Isso permite que, além dos valores processados, os valores originais também sejam mantidos. Dessa forma, tem-se a versão original e a versão transformada dos textos.

Essa camada de normalização contém uma função de adição e um processo de normalização de camada. A função *Add* processa as conexões residuais que vêm da entrada da subcamada. Essa operação garante que informações críticas não sejam perdidas. A Equação 2.10 descreve o funcionamento dessa camada, sendo que, x é a representação *embedding* do fluxo antes de passar pela camada anterior e $sublayer(x)$ representa a saída da camada imediatamente anterior à camada de normalização (uma camada de atenção ou *Feed Forward*).

$$Add\&Norm(x) = LayerNorm(x + Sublayer(x)) \qquad [3] \qquad (2.10)$$

2.4.6 *Masked Multi-head Attention*

O *Masked Multi-head Attention* é uma estrutura que só está presente no decodificador. Essa estrutura funciona de maneira similar ao *Multi-head Attention*, a única diferença é que ela aplica uma máscara na matriz de entrada. A função dessa máscara é mascarar as palavras localizadas após a palavra que está sendo tratada. Dessa forma, a arquitetura aprende a prever essas palavras futuras que foram mascaradas.

Cabe ressaltar que a subcamada *Masked Multi-head Attention*, representada pelo número 7 na Figura 2.5, também é seguida por uma subcamada de normalização, que funciona de forma idêntica à descrita na Subseção 2.4.5.

2.4.7 Última camada Linear e Camada *Softmax*

Na saída do decodificador há uma última camada densa, essa camada tem o tamanho do vocabulário. Finalmente, o último componente da arquitetura é uma função *softmax*, que é aplicada com o objetivo de gerar as probabilidades de cada uma das palavras. Essas estruturas são representadas pelo número 10 na Figura 2.5.

2.5 *Bidirectional Encoder Representations from Transformers* (BERT)

BERT (*Bidirectional Encoder Representations from Transformers*) [4] é um modelo de representação de linguagem baseado na arquitetura *Transformer* [3]. Esse modelo é considerado bidirecional pelo fato das palavras serem representadas com base tanto no contexto à esquerda quanto no contexto à direita. Tais representações são obtidas a partir de treinamentos em bases de dados textuais não rotuladas.

A partir do modelo do BERT, surgiu uma série de modelos que apresentam algumas variantes em relação ao modelo original: ROBERTA [33], ALBERT [34], DistilBERT [35] e outros. No entanto, este estudo ficará restrito ao BERT.

2.5.1 Modelos Anteriores

Durante o processo de evolução das técnicas de processamento de linguagem natural, muitos estudos apresentaram resultados satisfatórios, inspirando assim o surgimento de novos modelos. Dentre os modelos que precederam o BERT pode-se destacar o Elmo [36] e o OpenAI GPT [37].

- **ELMO**

O ELMO [36] não implementa o conceito de *Transformers*, nem de transferência de aprendizagem. Esse modelo apresenta duas características principais: é considerado pseudo-bidirecional e utiliza redes neurais recorrentes.

A arquitetura ELMO é classificada como pseudo bidirecional porque, apesar de considerar tanto o contexto à esquerda quanto o contexto à direita das palavras, ela não considera a sentença inteira de uma única vez. A sentença não passa por um núcleo único que processa todo o texto de forma conjunta. Sendo assim, a sentença é dividida em duas partes: uma à esquerda da palavra que está sendo considerada e outra à direita dessa palavra.

A Figura 2.11 ilustra a arquitetura ELMO, sendo que, cada elemento E_i representa uma frase de entrada. Como pode ser observado, cada frase é repassada tanto para a parte esquerda quanto para a parte direita da arquitetura. Ao final do processo, os resultados desses dois núcleos de processamento são concatenados em uma única saída.

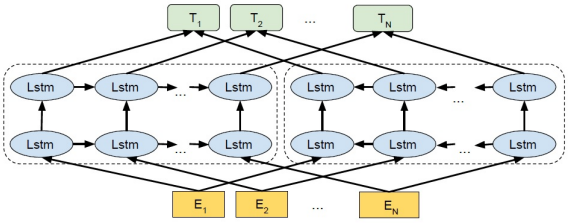

Figura 2.11: Arquitetura ELMO [4].

Cabe ressaltar que as setas internas nos dois núcleos de processamento, apresentados na Figura 2.11, representam a direção em que as informações são processadas.

O fato dessa arquitetura utilizar células recorrentes (mais especificamente células LSTM) representa uma desvantagem para esse tipo de processamento. Essa desvantagem ocorre porque as LSTMs fazem sucessivas chamadas recursivas, e à medida que as camadas ficam mais profundas, a informação tende a ser perdida.

- **OpenAI GPT**

A arquitetura OPEN AI GPT [37], apresentada na Figura 2.12, é baseada em *Transformers*. No entanto, ela só considera o contexto à esquerda dos *tokens*. Sendo assim, a principal deficiência desse modelo é que ele não implementa o conceito bidirecional.

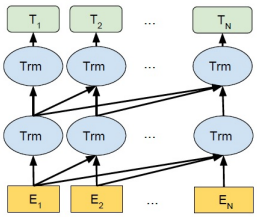

Figura 2.12: Arquitetura GPT-1 [4].

2.5.2 Arquitetura BERT

O BERT [4] combina as características dos modelos anteriores: utiliza a arquitetura *Transformer* e é bidirecional. A bidirecionalidade é obtida pelo fato das frases serem processadas tanto em relação aos dados à esquerda quanto à direita. Dessa forma, durante o treinamento, uma frase é processada tanto para a direita quanto para a esquerda, porém em um mesmo centro de processamento.

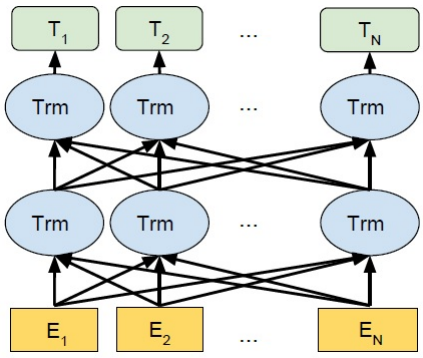

Figura 2.13: Arquitetura BERT [4].

O BERT é composto por uma série de codificadores (*Transformers*) empilhados. Dessa forma, o BERT utiliza apenas uma parte da arquitetura *Transformer*. Esses codificadores empilhados permitem que as sentenças sejam codificadas de maneira mais efetiva, possibilitando uma melhor composição dessas sentenças, de acordo com as relações entre as palavras.

O BERT possui duas arquiteturas básicas, o *BERT BASE* e o *BERT LARGE*, sendo que a principal diferença entre essas arquiteturas é o número de codificadores empilhados. O *BERT BASE* é constituído por uma sequência de 12 codificadores, enquanto o *BERT LARGE* é composto por 24 codificadores empilhados.

Apesar dessa diferença principal, as variantes do BERT podem ser definidas pela combinação de 3 parâmetros: L, H e A. O parâmetro L indica o número de camadas de codificadores. Já o parâmetro H indica a dimensão dos *embbedings* gerados (número de unidades na última camada oculta da arquitetura) e o parâmetro A indica o número de *self-attentions heads*. A Figura 2.14 apresenta a composição dos modelos *Base* e *Large* do BERT. Cabe ressaltar que o modelo *Base* possui cerca

de 110 milhões de parâmetros, enquanto o modelo *Large* 340 milhões.

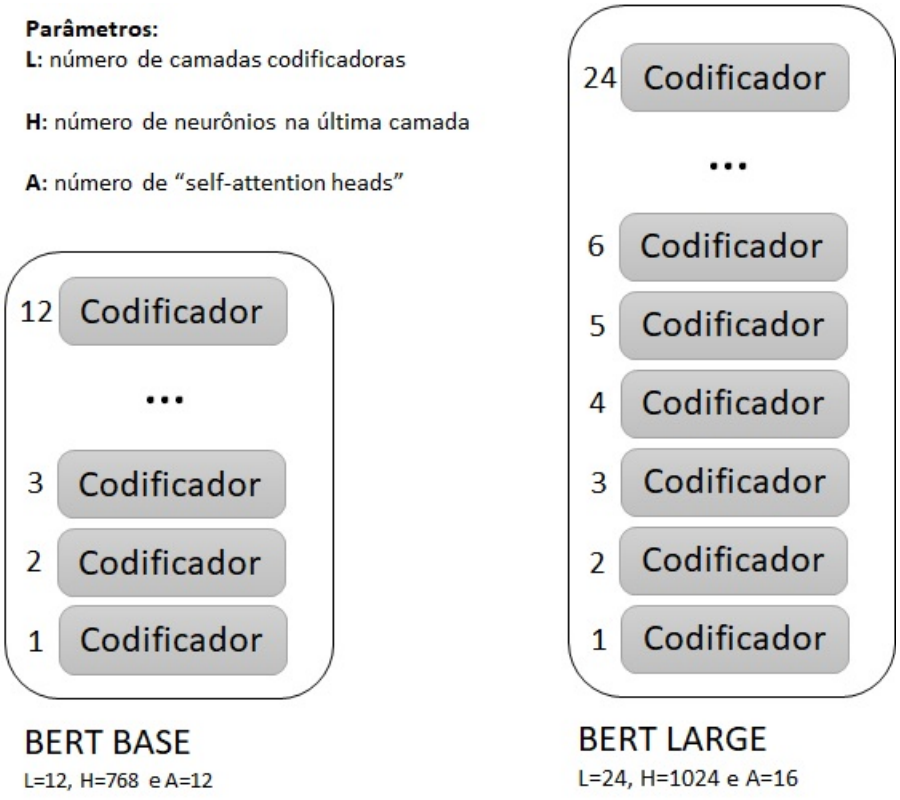

Figura 2.14: Composição dos modelos BERT.

2.5.3 Treinamento do BERT

O modelo original do BERT foi pré-treinado utilizando o *BookCorpus* (corpus com textos de livros com cerca de 800 milhões de palavras) e a Wikipedia em inglês (com cerca de 2,5 milhões de palavras). O pré-treinamento utilizou 16 TPUs[10] e demorou cerca de 3 dias, ou seja, esse processo é complexo e exige um grande poder computacional para executá-lo.

O pré-treinamento do BERT é composto por duas tarefas: o *Masked Language Modeling* (MLM) e o *Next Sentence Prediction* (NSP).

- ***Masked Language Modeling*** **(MLM)**

Na tarefa *Masked Language Modeling*, realiza-se um mascaramento aleatório de uma percentagem do *tokens* de entrada para que o modelo possa aprender como predizê-los. Dessa forma, substitui-se a palavra a ser predita, colocando-se o *token [MASK]* no lugar, e tenta-se obter o *embedding* desse *token* na saída, conforme ilustrado na Figura 2.15.

[10]Tensor Processing Unit-TPU é uma unidade de processamento de tensores desenvolvidas pelo Google para acelerar o processamento em aplicações de *Deep Learning*.

DEVLIN *et al.* [4] deixam claro que, diferentemente dos *autoencoders* de eliminação de ruído [38], o objetivo dessa tarefa não é reconstruir toda a sequência de palavras de entrada, mas apenas fazer a previsão da palavra mascarada.

Cabe ressaltar que esse token $[MASK]$ não é empregado em todas as iterações, ele é utilizado em apenas 15% das palavras. Ou seja, apenas 15% das palavras são substituídas, sendo que, tal substituição obedece a seguinte regra: em 80% dos casos, substitui-se pelo token $[MASK]$, em 10% dos casos, substitui-se por um token aleatório, e em 10% dos casos, não se modifica o token.

DEVLIN *et al.* [4] informam que a vantagem desse procedimento é que o codificador do *Transformer* não sabe quais palavras ele terá que prever, e nem quais foram substituídas por palavras aleatórias, então, ele é forçado a manter uma representação contextual distributiva de cada *token* de entrada. Além disso, como a substituição aleatória ocorre em uma pequena percentagem dos *tokens*, isso não prejudica a capacidade do modelo em compreender a linguagem.

A Figura 2.15 ilustra a tarefa de treinamento denominada MLM. Nessa figura, utiliza-se uma frase como entrada e aplica-se um mascaramento aleatório. Sendo assim, no exemplo em questão, a palavra correr foi substituída pelo token $[MASK]$, logo, deseja-se fazer a previsão dessa palavra. Ao final do processo aplica-se uma camada totalmente conectada (*feed forward neural network*), seguida de uma função *softmax* para gerar a probabilidade para cada uma das classes (possíveis *tokens*). Sendo assim, tem-se um problema de classificação em que se retorna uma probabilidade para cada um dos *tokens* do vocabulário, sendo que, o objetivo é que o *token* mascarado (correr) seja o *token* de maior probabilidade.

- ***Next Sentence Prediction* (NSP)**

A segunda tarefa de treinamento do BERT é a Predição da Próxima Sentença. A entrada consiste em duas frases e o modelo deve dizer se a segunda sentença é sequência da primeira ou não.

Dessa forma, essa tarefa consiste em uma classificação binária, sendo que, sempre que a segunda sentença for sequência da primeira, o rótulo é positivo; e no caso contrário, o rótulo é negativo.

A tarefa utiliza dois tokens especiais: $[CLS]$ e $[SEP]$. O $[CLS]$ é um token de classificação binário que é adicionado ao início da primeira frase para prever se a segunda frase é sequência da primeira. Já o $[SEP]$ é um token de separação que sinaliza o fim de uma frase. Sendo assim, a primeira frase é separada da segunda pelo token $[SEP]$.

As duas tarefas (MLM e NSP) são executadas simultaneamente, sendo que, as sequências de entrada devem ser previamente preparadas. A Figura 2.16 ilustra esse preparo.

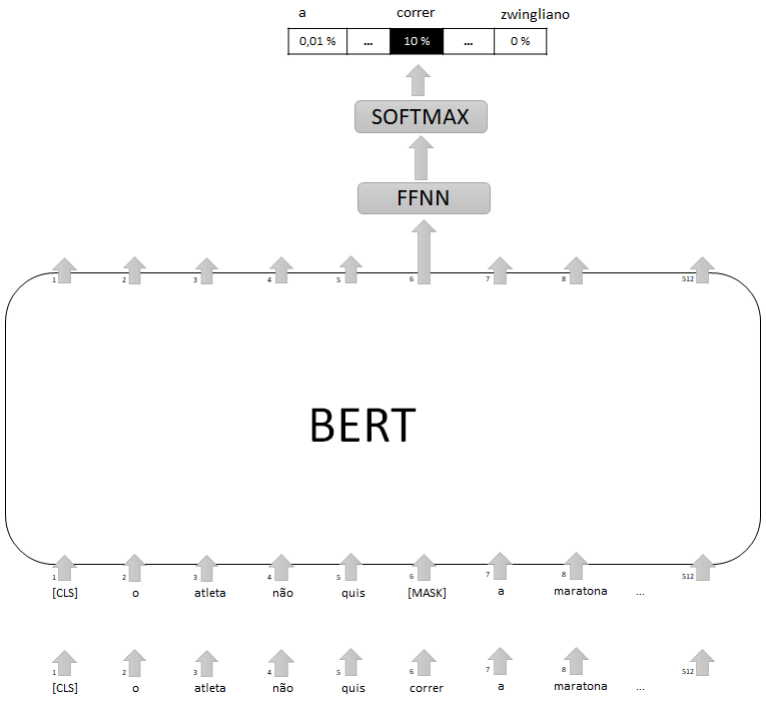

Figura 2.15: Treinamento "Masked Language Modeling".

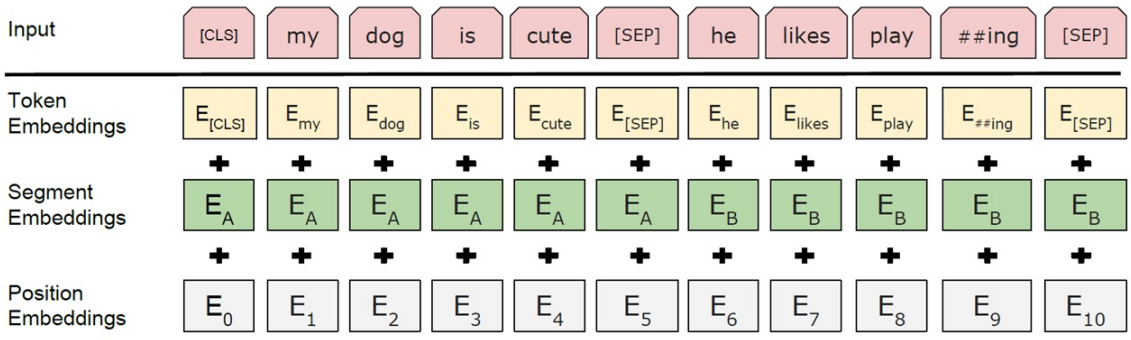

Figura 2.16: Entrada do modelo BERT [4].

Conforme pode ser observado, os *embeddings* de entrada são obtidos pela soma dos *tokens embeddings* com os *segment embeddings* e os *embeddings* de codificação posicional. Essas operações podem ser resumidas da seguinte forma:

1. A sequência de palavras é dividida em *tokens*,
2. Substitui-se aleatoriamente alguns tokens pelo token $[MASK]$.
3. Um *token* de classificação $[CLS]$ é inserido no início de uma das sentenças e um *token* $[SEP]$ separa a primeira sentença da segunda.
4. O *embedding* das sentenças é adicionado ao *embedding* de *tokens*, de modo que a primeira frase tenha um valor de *embedding* diferente da segunda frase.
5. A codificação posicional é aprendida, sendo que, nesse caso não se utiliza a codificação posicional seno-cosseno do *transformer* original.

Na tarefa de NSP, busca-se um entendimento de nível superior, passando do nível de palavras para o nível de frases. Dessa forma, o modelo torna-se adequado para outros tipos de tarefas, como por exemplo, perguntas e respostas. A Figura 2.17 ilustra esse treinamento.

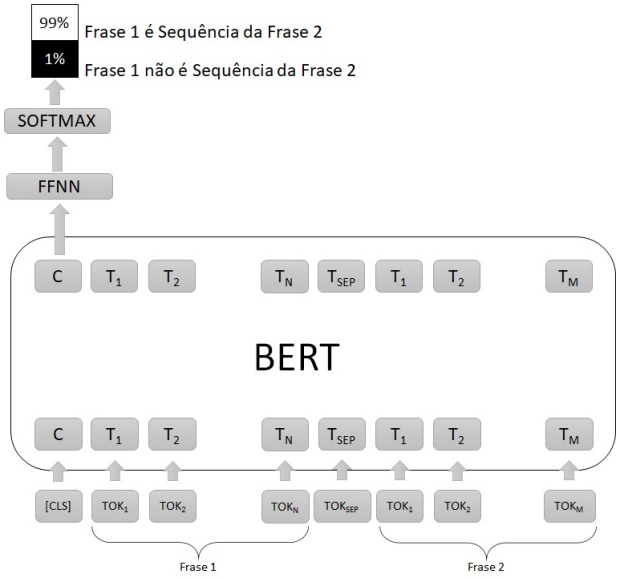

Figura 2.17: Treinamento "Next Sentence Prediction".

2.5.4 *Fine-tuning*

A grande vantagem do BERT é a possibilidade de transferência de aprendizagem. Dessa forma, uma vez que se tenha o modelo pré-treinado, esse pode ser utilizado

em diversas tarefas de PLN, como por exemplo: classificação textual, verificação de semelhança entre sentenças, sistemas de perguntas e respostas.

Para isso, deve-se realizar uma tarefa de *fine-tuning*, a fim de adaptar o modelo original a uma tarefa específica. Nesse sentido, SUN *et al.* [39] investigaram como maximizar o desempenho do BERT para a tarefa de classificação de textos.

O modelo BERT recebe como entrada sequências de 512 *tokens* e fornece como saída vetores de 768 ou 1024 posições[11]. Sendo assim, a tarefa de *fine-tuning* consiste na inclusão de novas camadas ao modelo, a fim de adaptar a estrutura do modelo original ao problema em questão. Durante o *fine-tuning* também é realizado o treinamento com os dados específicos do problema em questão.

SUN *et al.* [39] defendem que, quando se adapta o BERT para uma tarefa de PLN em um domínio específico, deve-se buscar uma estratégia de *fine-tuning* apropriada. Dessa forma, os autores sugerem 3 métodos de *fine-tuning*.

- ***Fine-tuning Strategies*:** quando se ajusta o BERT para uma tarefa específica, existem muitas formas de utilizar o BERT. SUN *et al.* [39] argumentam que as diferentes camadas do BERT capturam diferentes níveis de informações semânticas e sintáticas. Dessa forma, torna-se necessário saber qual camada é a melhor para uma determinada tarefa. Os autores defendem ainda que, além da definição da camada, deve-se também escolher de forma adequada a taxa de aprendizagem e a estratégia de otimização.

- **Pré-treinamento adicional:** o BERT é treinado em textos de domínio geral (genéricos), com distribuições de dados diferentes da distribuição dos dados dos textos da tarefa de domínio específico, que se deseja adaptar. Sendo assim, SUN *et al.* [39] sugerem um pré-treinamento adicional, com os dados referentes ao domínio da tarefa específica.

- ***Multi-task fine-tuning*:** SUN *et al.* [39] defendem que, quando há várias tarefas disponíveis em um domínio específico, uma questão interessante é verificar se há benefícios no *fine-tuning* do BERT em todas as tarefas simultaneamente.

SUN *et al.* [39] citam a limitação do BERT de trabalhar com tamanhos máximos de textos de até 512 tokens e sugerem dois métodos para solucionar essa dificuldade: o método do truncamento e o método hierárquico.

O método do truncamento consiste em truncar o texto de forma que esse passe a ter apenas 510 tokens (reserva-se dois tokens, para indicar o início e o final do

[11] O modelo *BERT BASE* possui 768 neurônios na última camada oculta, e por tal razão a sua saída é um vetor de 768 valores. Já o *BERT LARGE*, apresenta com saída um vetor de 1024 posições, visto que, ele possui 1024 neurônios na última camada oculta.

texto). Para esse truncamento, SUN *et al.* [39] citam 3 estratégias distintas: manter os 510 tokens iniciais, manter os 510 tokens finais ou selecionar os 128 primeiros tokens e os 382 tokens finais.

Nos métodos hierárquicos, SUN *et al.* [39] propõem a divisão do texto de entrada em grupos de 510 tokens. Dessa forma, torna-se possível apresentar ao modelo cada um desses grupos de forma separada. Assim, obtém-se a representação de cada um desses grupos (de 510 tokens) e em seguida combina-se essas representações a fim de se obter uma representação final.

SUN *et al.* [39] também citam o problema do esquecimento catastrófico, apresentado por MCCLOSKEY e COHEN [40]. Esse tipo de problema ocorre em situações de transferência de aprendizado, em que o conhecimento pré-treinado é apagado durante a aprendizagem de novos conhecimentos.

2.6 Outros Modelos

Um dos grandes avanços introduzidos pela arquitetura *Transformers* foi a proposta de um mecanismo de atenção, que pode ser avaliado em paralelo para cada *token* da sequência de entrada, eliminando assim a dependência sequencial que ocorria em redes neurais recorrentes, como a LSTM.

No entanto, a limitação desse mecanismo é o seu alto custo computacional. O fato de todos os *tokens* se relacionarem com todos os demais *tokens* da sequência torna quadrática a complexidade desse mecanismo (complexidade $O(n^2)$) .

Por tal razão, modelos derivados da arquitetura *Transformers* possuem tamanhos máximos de entrada bem restritos. Um exemplo disso é o modelo BERT, visto anteriormente, que só trabalha com textos de até 512 *tokens*.

Sendo assim, passou-se a pesquisar formas de se alcançar os benefícios trazidos pelos mecanismos de atenção, introduzidos pela arquitetura *Transformers*, sem o alto custo computacional.

Nesse sentido, foram propostas algumas soluções que diminuíam a complexidade do mecanismo de atenção. Dentre essas propostas estão as arquiteturas *Longformer* [5] e *Big Bird* [6], que possuem complexidade linear e por isso aceitam textos maiores.

2.6.1 *Longformer: The Long-Document Transformer*

BELTAGY *et al.* [5] definem alguns padrões de atenção que possibilitam a matriz de atenção escalar linearmente com o tamanho da sequência de entrada. Os padrões de atenção propostos foram a janela deslizante, a janela deslizante dilatada e a atenção global, que são apresentados na Figura 2.18.

A janela deslizante é o padrão que considera um tamanho de janela arbitrário w, e

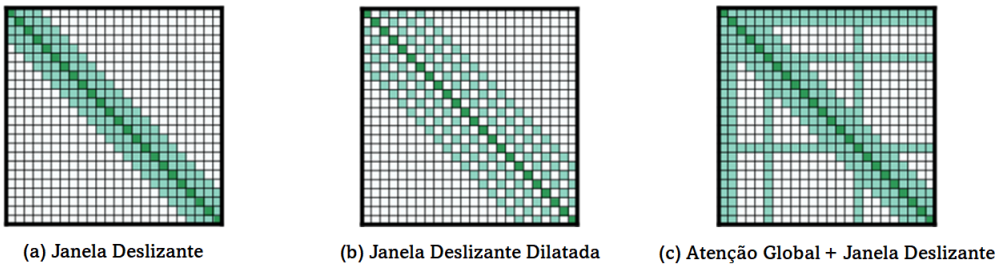

Figura 2.18: Padrões de atenção da arquitetura LongFormer, adaptado de [5]

cada token se relaciona apenas com os w tokens mais próximos dele ($w/2$ à esquerda e $w/2$ à direita). Essa modificação faz com que a complexidade do mecanismo de atenção caia para $O(n)$. Assim, passa-se de uma complexidade quadrática para uma complexidade linear.

Os autores defendem a utilização de diferentes valores de w para camadas distintas, pois isso pode propiciar um equilíbrio entre a eficiência e a capacidade de representação do modelo.

BELTAGY *et al.* [5] também propõem a janela deslizante dilatada, com o objetivo de aumentar o campo receptivo sem aumentar a computação. Essa janela funciona de forma análoga a citada anteriormente, porém, com lacunas de dilatação.

Dessa forma, torna-se possível atingir um número maior de tokens, sem prejudicar a complexidade do algoritmo. Isso ocorre porque as lacunas deixam os campos respectivos mais amplos, mantendo o número de operações matemáticas.

Apesar das soluções anteriores resolverem o problema da complexidade do mecanismo de atenção, elas não têm condições de tratar dependências de longa duração, ou seja, elas não permitem a identificação de relacionamentos entre *tokens* que estejam muito distantes no texto.

Sendo assim, para tratar esse tipo de situação, os autores propõem também a utilização do mecanismo de atenção tradicional. No entanto, a ideia não é calcular a relação entre todos os *tokens*, como acontece em [3]. Nesse caso, apenas alguns *tokens* pré-selecionados tem a sua atenção totalmente calculada.

Os autores ressaltam que essa operação de atenção é simétrica, ou seja, um *token* com uma atenção global atende a todos os *tokens* na sequência e todos os *tokens* na sequência atendem a ele.

Porém, como o número desses *tokens* selecionados não depende do tamanho da sequência (ou seja, a quantidade de *tokens* que recebem a atenção global não varia com o tamanho de n), a complexidade desse mecanismo de atenção continua sendo $O(n)$.

Sendo assim, o *Longformer* trabalha com mecanismos de atenção que combinam janelas deslizantes com a aplicação de atenção global para alguns *tokens* seleci-

onados. Essa combinação consegue resultados semelhantes aos dos *Transformers* originais, porém, com complexidade bem menor.

2.6.2 *Big Bird: Transformers for Longer Sequences*

ZAHEER *et al.* [6] também propõem um mecanismo de atenção esparsa que reduz a complexidade quadrática dos mecanismos originais para uma complexidade linear. Esse modelo pode ser entendido como uma combinação de 3 tipos de mecanismos de atenção: atenção aleatória, janela de atenção e atenção global. A Figura 2.19 ilustra o funcionamento desses mecanismos.

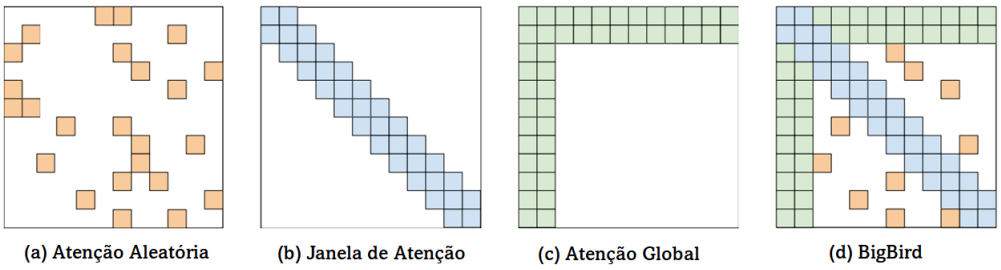

Figura 2.19: Mecanismo de atenção do BigBird, adaptado de [6]

A atenção aleatória é uma proposta em que um determinado *token* só se conecta com alguns *tokens* aleatoriamente selecionados, ao invés de se conectar com todos os demais *tokens* da sequência, como ocorre no mecanismo de atenção original. Sendo assim, supondo que sejam selecionados r *tokens* para se conectarem com um determinado *token* (por exemplo $r = 2$), a complexidade deixa de ser $O(n^2)$ e passa a ser $O(r.n)$. Porém, como r é um valor constante, pode-se dizer que a complexidade é $O(n)$.

Dessa forma, para cada *token* da sequência, seleciona-se um número aleatório de *tokens* com quem ele irá se relacionar, sendo que esse número aleatório é fixo, não dependendo assim do tamanho da sequência.

Já a janela aleatória funciona de maneira análoga à janela deslizante proposta por [5]. Nesse tipo de mecanismo, cada *token* se comunica com um número fixo w de vizinhos mais próximos. Logo, a complexidade passa de $O(n^2)$ para $O(w.n)$. No entanto, como w também é fixo, pode se dizer que a complexidade é $O(n)$.

Por fim, na atenção global, seleciona-se alguns *tokens* que se conectam a todos os demais *tokens*. Portanto, essa estrutura também é similar à empregada em [5].

O Big Bird une essas três estruturas a fim de propor uma aproximação para o mecanismo de atenção original. Portanto, busca-se resultados semelhantes com menos complexidade. Essas alterações permitem a utilização de sequências de textos maiores.

2.7 Conclusão

Esse capítulo apresentou um estudo sobre as principais técnicas de *Deep Learning* para o Processamento de Linguagem Natural. No entanto, esse assunto é muito amplo e constantemente novas técnicas estão sendo propostas. Dessa forma, esse trabalho não tem a pretensão de esgotar esse assunto, sendo apenas uma fundamentação teórica para os estudos desenvolvidos nessa tese.

Capítulo 3

Análise da Arquitetura mais apropriada para a classificação de denúncias

Esse capítulo apresenta os estudos realizados para responder à questão de pesquisa 1: "Qual é a arquitetura de rede mais apropriada para o problema de classificação de aptidão de denúncias?".

A evolução dos algoritmos de *Deep Learnig* tem propiciado grandes avanços na área de Processamento de Linguagem Natural, inclusive no que diz respeito à classificação de textos. No entanto, para cada caso específico, uma solução diferente pode ser considerada como a mais adequada.

Logo, nesse capítulo são propostas e avaliadas quatro arquiteturas de rede a serem utilizadas no problema de classificação de textos de denúncias. Sendo assim, o restante desse Capítulo está dividido da seguinte forma: a Seção 3.1 faz um estudo de trabalhos relacionados. Já as Seções 3.2 e 3.3 apresentam as arquiteturas propostas e os experimentos realizados. A Seção 3.4 analisa os resultados e finalmente a Seção 3.5 enumera as conclusões do capítulo.

3.1 Trabalhos Relacionados

Atualmente, as redes neurais profundas têm sido uma das principais técnicas utilizadas no Processamento de Linguagem Natural. Essa tendência começou com as redes CNNs e RNNs e foi seguida pelos modelos de linguagem pré-treinados como o BERT e os seus derivados.

Nesse sentido, KIM [27] aplicou redes CNNs em diversas tarefas de classificação de sentenças. KIM [27] explica que os núcleos convolucionais inicializados aleatori-

amente se tornam detectores específicos de recursos de *n-grama*[1] úteis para tarefas de classificação.

KALCHBRENNER *et al.* [41] tentaram tratar o problema da dependência de longa duração, presente em textos mais longos, com a utilização de uma rede neural convolucional dinâmica (DCNN) para a modelagem semântica de sentenças. Essa rede usava o Dynamic k-Max Pooling, uma operação de *poolling* global sobre sequências lineares. Isso fazia com que a rede conseguisse lidar com frases de comprimentos variados, tornando-a capaz de capturar tanto relações próximas, quanto distantes em uma determinada sentença.

Os autores testaram a DCNN em quatro experimentos: previsão de sentimentos binários e multiclasse em pequena escala, classificação de perguntas e previsão de sentimentos no Twitter por supervisão distante, obtendo resultados satisfatórios.

WANG *et al.* [42] propuseram o uso da CNN para modelar representações de textos curtos, que sofrem com a falta de contexto disponível e, portanto, exigem esforços extras para se criar representações significativas. O artigo propõe um agrupamento semântico, que introduz unidades semânticas em várias escalas para serem usadas como conhecimento externo para os textos curtos. Nesse caso, a CNN foi usada para combinar essas unidades e formar a representação geral.

PORIA *et al.* [43] propuseram a detecção de sarcasmo em textos do Twitter usando uma rede CNN. Os autores utilizaram redes pré-treinadas, treinadas em conjuntos de dados de emoções, sentimentos e personalidade, e conseguiram obter um desempenho satisfatório.

ZHOU *et al.* [44] combinaram as arquiteturas LSTM e CNN e propuseram um modelo unificado, chamado C-LSTM, para representação de sentenças e classificação de texto. A arquitetura proposta utiliza rede CNN para extrair uma sequência de representações de frase de nível superior e alimenta uma rede LSTM para obter a representação das sentenças. O C-LSTM é capaz de capturar características locais, assim como a semântica global e temporal das frases. Essa proposta foi avaliada em tarefas de classificação de sentimentos e classificação de perguntas, sendo que os resultados superaram os obtidos pela aplicação isolada de redes CNNs e LSTMs.

VU *et al.* [45] analisaram a CNN e a RNN básica para tarefas de classificação. Os autores relatam maior desempenho da CNN em relação a RNN, mas, eles indicam que as CNNs e as RNNs fornecem informações complementares. Segundo os autores, enquanto as RNNs calculam uma combinação ponderada de todas as palavras da sentença, as CNNs extraem os *n-gramas* mais informativos e consideram apenas as ativações resultantes.

Considerando que ambas as arquiteturas apresentam resultados satisfatórios na atividade de processamento de linguagem natural, YIN *et al.* [46] se propõem a fazer

[1] *n-gram* é um termo utilizado para indicar combinações de palavras que ocorrem juntas.

um estudo comparativo entre as redes CNN e RNN para esse tipo de tarefa. O objetivo do estudo era fornecer orientações básicas para a seleção da rede a ser utilizada. Segundo os autores, a arquitetura com melhor desempenho depende da importância de entender semanticamente toda a sequência. YIN *et al.* [46] concluem que as RNNs apresentam bom desempenho em uma ampla gama de tarefas, exceto quando a tarefa é essencialmente uma tarefa de reconhecimento de frase-chave, como em algumas configurações de detecção de sentimentos e correspondência de perguntas e respostas.

Outra conclusão desse artigo foi que a escolha da quantidade de camadas ocultas e o tamanho do "batch" podem ter grande influência no desempenho da rede. Isso sugere que a otimização desses dois parâmetros é crucial para o bom desempenho das CNNs e RNNs.

Já com relação a utilização de modelos de linguagem pré-treinados, ZHENG e YANG [47] propõem uma combinação do modelo BERT com redes CNN (BERT-CNN). Os autores argumentam que a adição de camadas CNN a partes específicas da arquitetura BERT, torna possível a obtenção de fragmentos importantes dos textos.

YU *et al.* [48] argumentam que o modelo BERT tradicional, treinados com textos de domínios cruzados (BookCorpus e Wikipedia), apresentam bons resultados quando se aplica o ajuste fino em tarefas de classificação de textos gerais.

No entanto, os autores argumentam que para alguns casos mais específicos, a carência de conhecimentos específicos da tarefa de classificação em questão e de informações relacionadas ao domínio podem trazer prejuízos. Os autores alegam que para melhorar o desempenho do modelo BERT são necessárias análises de estratégia de ajuste fino mais detalhadas.

Sendo assim, eles propõem a utilização de sentenças auxiliares que permitem a transformação da tarefa de classificação em uma tarefa de pares de sentenças binárias. Com isso, além de se tratar os problemas apresentados, também se resolve a questão de dados de treinamento limitados.

Diante da diversidade de soluções para problemas de classificação textual, alguns trabalhos se propuseram a comparar diferentes propostas em cenários específicos. BANERJEE *et al.* [49] avaliaram as RNNs e as CNNs na classificação de textos em relatórios de radiologia. O trabalho utilizou essas arquiteturas para sintetizar informações sobre embolia pulmonar em mais de 7370 relatórios clínicos de radiologia coletados em quatro grandes centros de saúde dos Estados Unidos da América.

Os autores concluem que os resultados apontam para a viabilidade de CNNs e RNNs na classificação automatizada dos relatórios textuais e sugerem a aplicação dessas técnicas em vários casos de uso, tais como: priorização de pacientes em radiologia, geração de corte para pesquisa clínica, triagem de elegibilidade para ensaios clínicos e avaliação da utilização de imagens.

Seguindo essa linha, QUAN *et al.* [50] e UNDAVIA *et al.* [51] também realizam estudos comparativos entre redes CNNs e LSTMs para tarefas específicas de processamento de linguagem natural. QUAN *et al.* [50] comparam o desempenho dessas arquiteturas em problemas ligados a pesquisas de opinião. UNDAVIA *et al.* [51] avalia a performance dessas redes no contexto de textos jurídicos.

ZHAO *et al.* [52] propõem uma abordagem de análise de sentimento e detecção de entidade chave baseada no modelo BERT. A abordagem proposta é aplicada na mineração de textos financeiros online e na análise de opiniões publicadas em redes sociais. Primeiro, os autores utilizaram o modelo pré-treinado para a análise de sentimento e, em seguida, consideraram a detecção de entidade chave como uma correspondência de sentença ou tarefa de compreensão de leitura de máquina em granularidades distintas. No entanto, é importante ressaltar que os autores focam principalmente em informações de sentimentos negativos.

Como pode ser visto, não existe uma arquitetura ideal para todos os tipos de textos e tarefas, sendo que, a arquitetura mais apropriada varia de acordo com as características dos dados e do problema a ser resolvido. Nesse sentido, não foi identificado na literatura trabalhos que fizessem esse tipo de comparação para textos na língua portuguesa, e mais especificamente para textos de denúncias escritos por cidadãos no intuito de descrever irregularidades que estejam ocorrendo na Administração Pública. Os textos de denúncias diferem dos textos tradicionais pelo fato de necessitarem de informações específicas para qualificarem uma denúncia como apta. Sendo assim, a principal contribuição dessa parte da pesquisa é a identificação da arquitetura mais apropriada para a classificação de textos de denúncias escritos na língua portuguesa.

3.2 Proposta

Nessa seção são apresentadas quatro propostas de arquitetura para o problema de classificação textual de aptidão de denúncias. As propostas utilizam arquiteturas de redes LSTMs, CNNs, modelos pré-treinados BERT e uma combinação de modelos BERTs com LSTM. Essas arquiteturas são detalhadas nas subseções 3.2.2, 3.2.3, 3.2.4 e 3.2.5, respectivamente. No entanto, independente da arquitetura utilizada, os textos das denúncias precisam passar por uma preparação prévia, que é descrita na Subseção 3.2.1.

3.2.1 Preparação dos Textos

Como apresentado no Capítulo 1 (Seção 1.2), cada denúncia é composta por duas partes: o texto original da denúncia e os arquivos anexos que são utilizados para

subsidiá-las. Dessa forma, todo o conteúdo textual desses arquivos anexos também deve ser considerado no processo de classificação de denúncias.

Esses arquivos anexos podem estar em diferentes formatos, sendo que, os processos tradicionais de análise textual não conseguem tratar essa diversidade de formatos.

Sendo assim, o primeiro passo para se automatizar a análise das denúncias é a construção de um processo capaz de tratar os diferentes tipos de arquivos, retirar todo o conteúdo textual deles e colocar em um formato de texto plano, a fim de possibilitar a concatenação desse conteúdo com o texto original da denúncia.

A Figura 3.1 ilustra o processo de preparação dos dados textuais. Essa preparação torna possível o treinamento dos modelos de classificação de denúncias e a posterior inferência com a utilização dos modelos treinados.

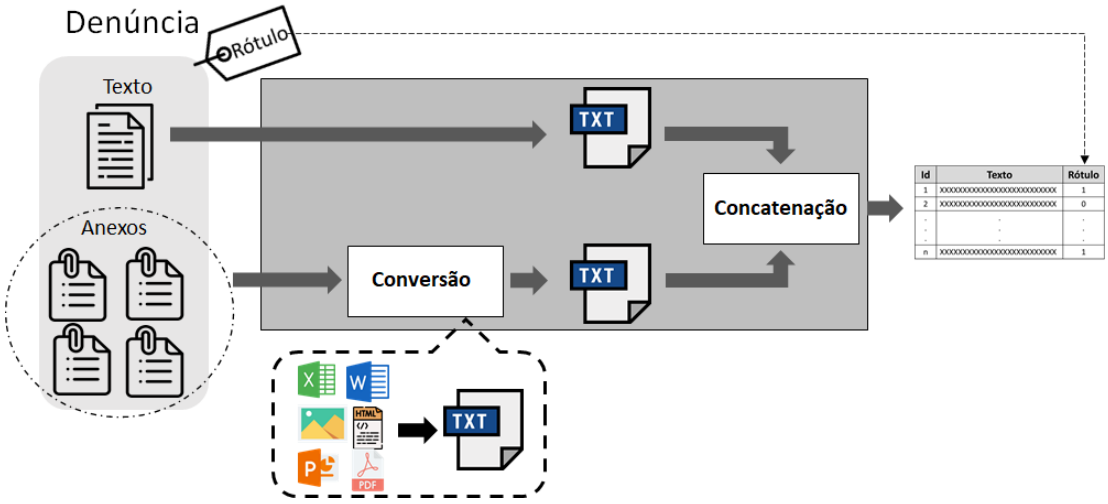

Figura 3.1: Processo de Preparação dos Dados Textuais

Conforme pode ser observado na Figura 3.1, todos os arquivos anexos passam por um processo de conversão. Essa fase de conversão recebe arquivos dos tipos planilhas, apresentações, PDFs, figuras, entre outros. Todos os conteúdos textuais desses arquivos são extraídos e apresentados em formato de texto plano. Essa atividade é executada com o auxílio da ferramenta Apache Tika[2].

Após isso, os conteúdos extraídos de cada um dos anexos são concatenados com o texto original das respectivas denúncias. Posteriormente, monta-se um arquivo com os dados textuais completos das denúncias, e com os seus respectivos rótulos, a fim de utilizá-los no treinamento dos modelos.

Esse mesmo processo também é utilizado para a preparação dos dados para a inferência. No entanto, nessa situação, a tabela final não possui a coluna com os rótulos das denúncias.

[2]Apache Tika é uma ferramenta que detecta e extrai metadados e texto de diferentes tipos de arquivos (https://tika.apache.org/).

3.2.2 Rede LSTM

A primeira estratégia proposta foi a utilização de uma arquitetura de rede LSTM. A rede em questão é composta por uma primeira camada de *embedding*, cujo objetivo é receber as palavras que compõem o texto de entrada e convertê-las em suas representações numéricas de vetores de *embeddings*. Após isso, utiliza-se uma camada LSTM, seguida por uma camada de *Dropout*[3], que utiliza uma taxa de *dropout* de 20%. Por fim, uma camada densa é responsável por fornecer as probabilidades de uma determinada denúncia ser considerada como "Apta" ou "Não Apta". A Figura 3.2 ilustra a arquitetura da rede LSTM.

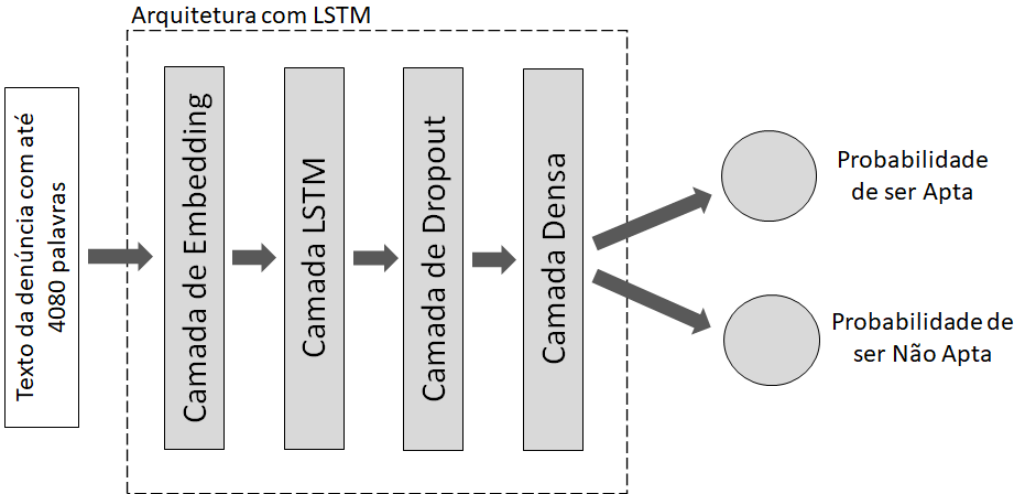

Figura 3.2: Arquitetura da rede LSTM

Conforme pode ser observado na Figura 3.2, essa arquitetura espera receber textos de até 4080 palavras. Sendo assim, caso o texto da denúncia em questão tenha mais do que esse limite de palavras, ele é truncado, de forma que apenas as 4080 palavras iniciais sejam consideradas.

Por outro lado, caso o tamanho da denúncia seja menor do que esse limite, o texto inteiro é considerado e a quantidade de palavras necessárias para completar as 4080 posições são completadas com valores zeros.

A Figura 3.3 ilustra esse processo, que pode ser descrito por uma sequência de dois passos. Primeiro, realiza-se um processo de tokenização. Essa tokenização nada mais é do que uma associação de cada uma das palavras do texto a um índice numérico, que indica a referência da palavra em questão. Após isso, caso o número de palavras do texto de entrada seja menor do que o tamanho de entrada do texto para o qual a rede está preparada (no caso 4080 palavras), completa-se com valores

[3] *Dropout* é uma técnica para resolver o problema de *overfitting*. Essa técnica descarta aleatoriamente unidades (neurônios) da rede neural durante o treinamento, evitando uma super adaptação dessas unidades [53]. A taxa de Dropout define a porcentagem de neurônios que são desconsideradas durante o processo de treinamento.

de índice zero até se alcançar o limite esperado. Logo, todos os textos de entrada ficam padronizados em um vetor de tamanho fixo (nesse caso, 4080 posições).

Dessa forma, a camada de *embedding* lê esses índices[4] e recupera os valores de *embeddings* associados a cada um dos índices passados. As representações *embeddings* utilizadas para se obter os vetores representativos das palavras que compõem os textos das denúncias foram obtidas por uma versão em português do *Word2Vec*, na sua variante CBOW [54]. A versão utilizada representava cada palavra com vetores numéricos de 50 posições.

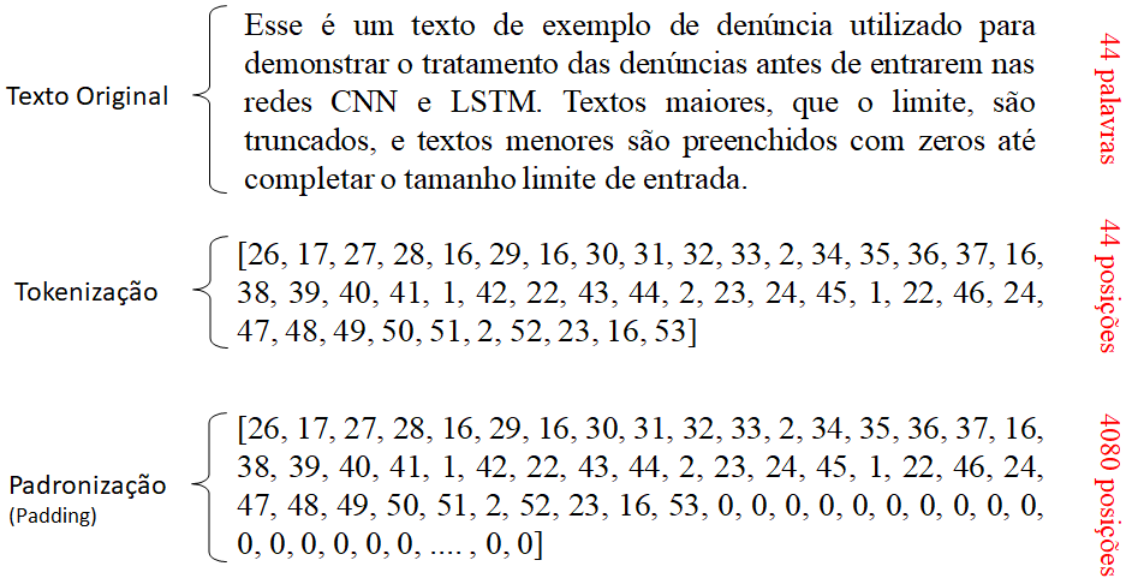

Figura 3.3: Tratamento dos dados textuais para a entrada das redes CNN e LSTM

3.2.3 Rede CNN

A segunda estratégia proposta utiliza uma rede CNN. Essa rede é composta por cinco camadas. A primeira camada é uma camada de *embedding*, cujo objetivo é atualizar os valores da *word embedding* carregada. A segunda camada é uma camada de convolução de 1 dimensão, formada por 8 filtros. Posteriormente, tem-se uma camada de *pooling* e uma camada de *dropout* (com taxa de *dropout* de 20%). Por fim, uma camada densa tem a responsabilidade de fazer a classificação dos dados (probabilidades da denúncia ser considerada Apta ou Não Apta). A estrutura dessa rede é apresentada na Figura 3.4.

Cabe ressaltar que os textos de entrada para a rede CNN passam pelos mesmo processo de preparação descrito para a rede LSTM.

[4]Os índices em questão são utilizados para referenciar as palavras do vocabulário. Sendo assim, cada palavra única (presente no texto) possui um índice associado. Esses índices são utilizados para identificar qual é o vetor de *word embedding* que representa a palavra em questão.

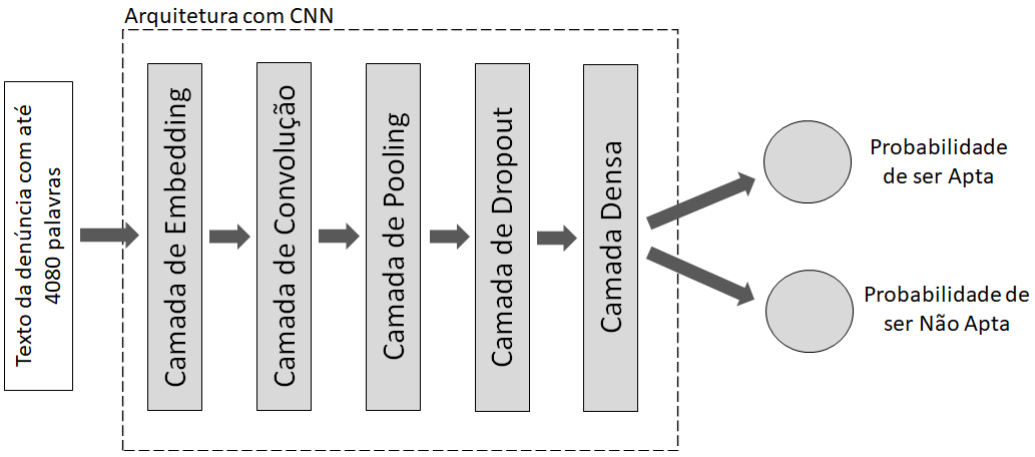

Figura 3.4: Arquitetura da rede CNN

3.2.4 Rede com o modelo BERT

A terceira estratégia proposta apresenta uma arquitetura de rede neural que utiliza uma versão em Português do modelo de linguagem BERT. A rede em questão é composta por uma camada de BERT BASE[5], que traz embutida toda a estrutura de uma rede BERT na sua variante BASE (apresentada no Capítulo 2). Além disso, essa rede possui uma camada de *dropout* com uma taxa de *dropout* de 20% e uma camada densa, responsável por fornecer as probabilidades de uma determinada denúncia ser considerada como "Apta" ou "Não Apta". A Figura 3.5 ilustra a estrutura da rede com o modelo BERT.

Conforme apresentado no Capítulo 2, uma das limitações do modelo BERT é o fato dele só ser capaz de processar textos com até 510 tokens[6]. Logo, os textos de denúncias que ultrapassam esse limite são truncados, de forma que apenas os 510 tokens iniciais sejam considerados.

3.2.5 Rede com o modelo BERT e LSTM

A fim de contornar a limitação da quantidade de 510 tokens imposta pela utilização do modelo BERT puro, a quarta estratégia a ser avaliada faz uma combinação de diversos modelos BERT em paralelo com uma camada LSTM, que recebe as saídas dos modelos BERT, de forma sequencial.

Sendo assim, o texto de entrada é dividido em grupos de 510 tokens e cada um desses grupos é processado por uma instância do modelo BERT (BERT Base). Após isso, uma camada LSTM recebe o vetor de saída de cada um dos modelos de

[5]A versão em português do modelo de linguagem BERT está disponível em https://huggingface.co/neuralmind/bert-base-portuguese-cased.

[6]O BERT trata até 512 tokens, no entanto, como a implementação do modelo utiliza um token especial para marcar o início da sentença e outro para marcar o final, sobram apenas 510 tokens para o texto propriamente dito.

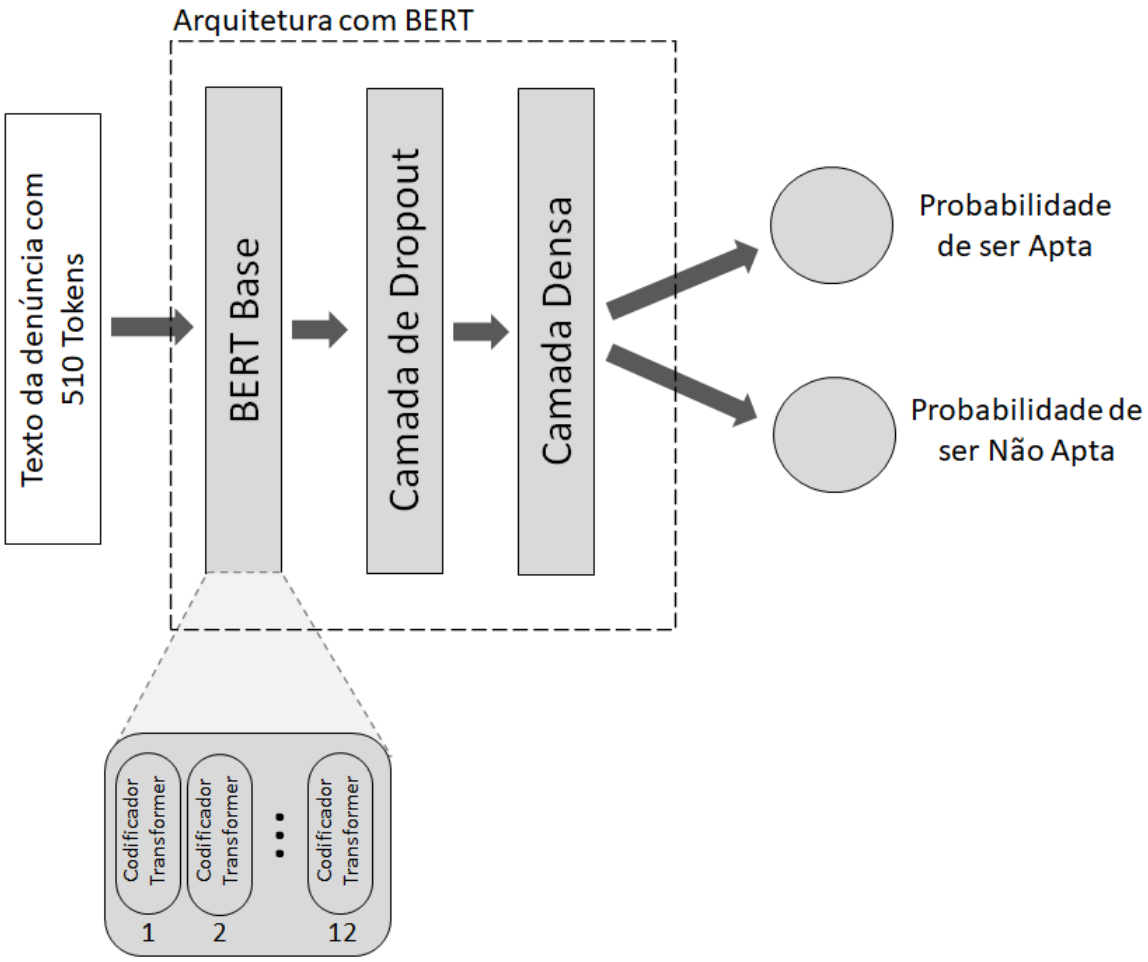

Figura 3.5: Arquitetura da rede com BERT

forma sequencial. Essa camada LSTM produz uma saída que é passada para uma camada densa, que é responsável por fornecer as probabilidades de uma determinada denúncia ser considerada como "Apta" ou "Não Apta". A Figura 3.6 ilustra essa rede.

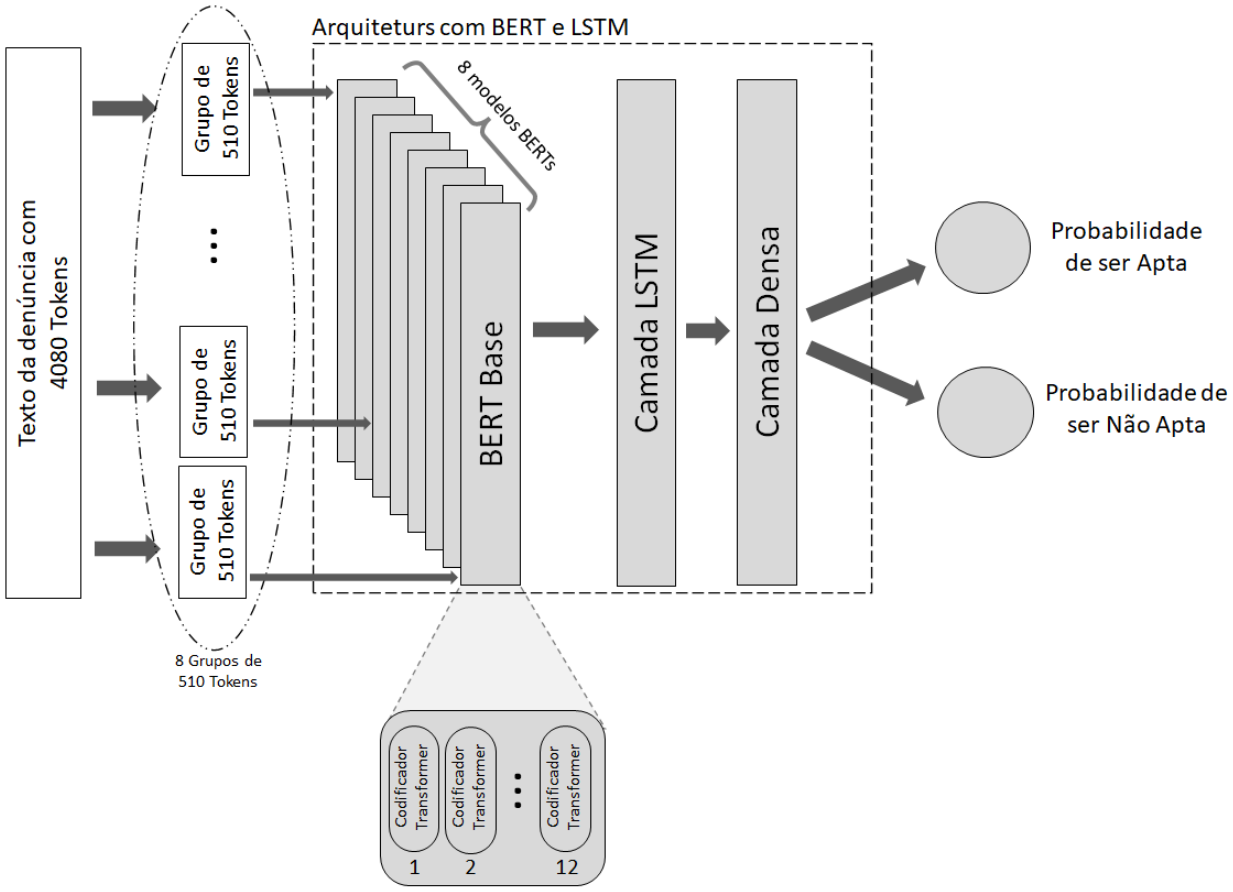

Figura 3.6: Arquitetura da rede com BERT e LSTM

3.3 Experimentos

Uma vez definidas as propostas de arquitetura a serem avaliadas, alguns experimentos foram realizados, a fim de se identificar a estrutura que melhor se adaptaria ao problema de classificação de aptidão de denúncias.

Sendo assim, para cada uma das arquiteturas avaliadas, os experimentos foram repetidos cinco vezes, a fim de diminuir os efeitos oriundos de uma escolha de conjuntos de dados (para treino e validação do modelo) que pudesse trazer algum tipo de viés para as análises. Cabe ressaltar que o número de repetições dos experimento ficou limitado a cinco pelo fato desses modelos serem muito complexos, consumindo assim muitos recursos computacionais, e exigindo grandes intervalos de tempo para o treinamento.

A base de dados utilizada para os experimentos era formada por 6939 registros

de denúncias, que representavam denúncias recebidas no período de dezembro de 2019 a setembro de 2022. Para cada uma das 5 execuções das arquiteturas testadas, separou-se aleatoriamente 80% dos registros para o treino e 20% para teste. O treinamento de cada uma das redes utilizou 3 épocas. A Figura 3.7 ilustra a estrutura dos experimentos.

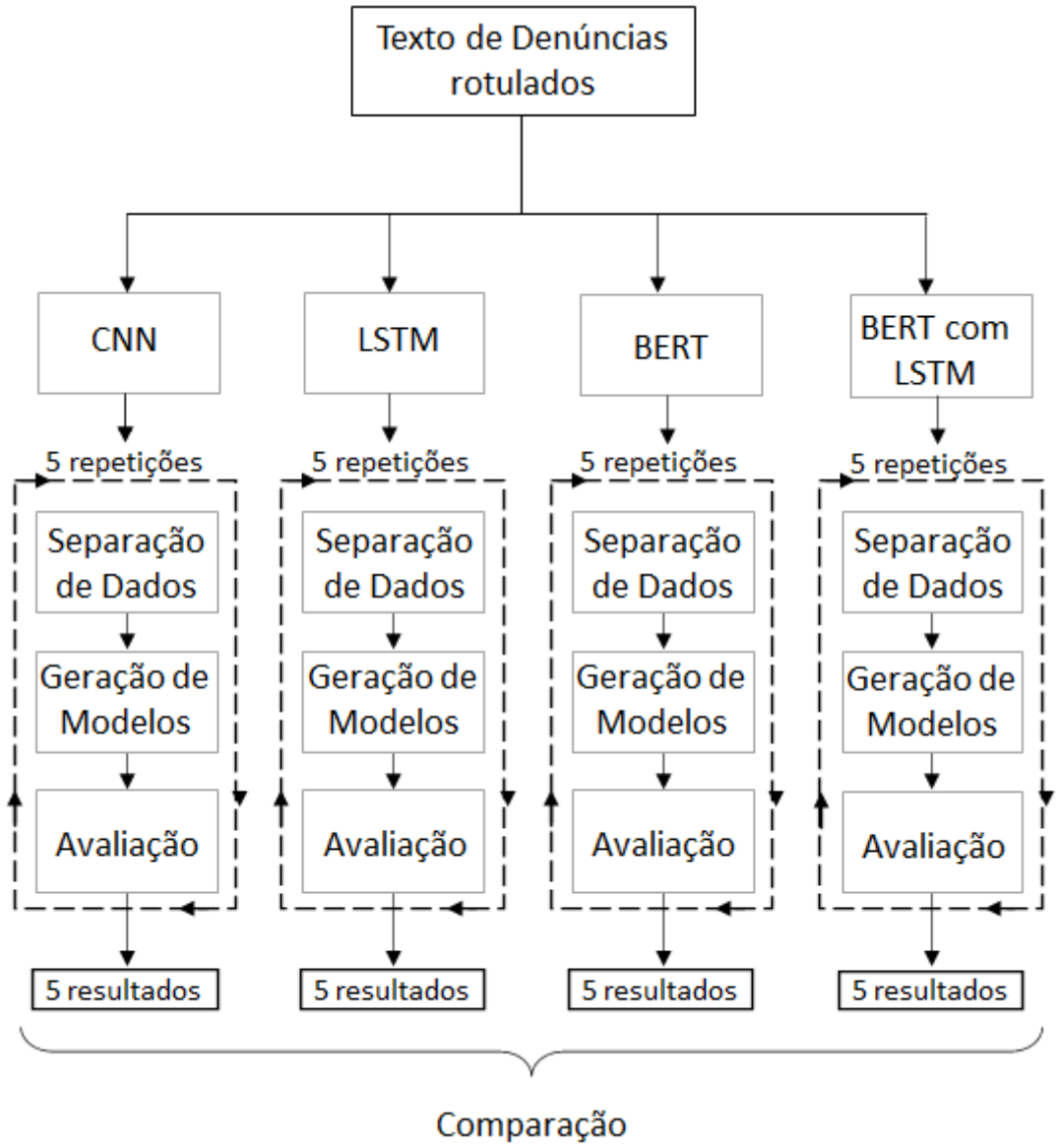

Figura 3.7: Estrutura dos experimentos

Conforme apresentado no Capítulo 1, as métricas que guiaram os experimentos dessa pesquisa foram a área sob a curva ROC (ROC-AUC) e Área sob a curva Precision-Recall (AUPRC). Sendo assim, as Tabelas 3.1, 3.2, 3.3 e 3.4 apresentam os resultados obtidos pelas arquiteturas CNN, LSTM, BERT e BERT com LSTM, respectivamente, para cada uma das 5 vezes em que os experimentos foram repetidos.

Tabela 3.1: Resultados do Experimento com arquitetura CNN

Iteração	**ROC-AUC**	**AUPRC**
1	0,7242	0,4846
2	0,7209	0,4984
3	0,7341	0,5029
4	0,7206	0,5195
5	0,7307	0,5161
Média	**0,7261**	**0,5043**

Tabela 3.2: Resultados do Experimento com arquitetura LSTM

Iteração	**ROC-AUC**	**AUPRC**
1	0,7120	0,4510
2	0,6945	0,4775
3	0,7253	0,5081
4	0,7165	0,4819
5	0,6938	0,4586
Média	**0,7084**	**0,4754**

Tabela 3.3: Resultados do Experimento com arquitetura BERT

Iteração	**ROC-AUC**	**AUPRC**
1	0,8285	0,6466
2	0,7435	0,5373
3	0,8282	0,6510
4	0,8228	0,6370
5	0,8119	0,6105
Média	**0,8070**	**0,6165**

Tabela 3.4: Resultados do Experimento com arquitetura BERT com LSTM

Iteração	**ROC-AUC**	**AUPRC**
1	0,6828	0,4178
2	0,6580	0,4153
3	0,6987	0,4489
4	0,6527	0,4177
5	0,6804	0,4266
Média	**0,6745**	**0,4252**

3.4 Análise dos Resultados

Como pode ser observado nas Tabelas 3.1, 3.2, 3.3 e 3.4, a arquitetura que demonstrou melhor desempenho foi a que utilizou o modelo BERT puro, seguido pelas arquiteturas CNN e LSTM, respectivamente. A arquitetura que combinava os modelos BERT com LSTM foi a que apresentou pior desempenho. A Figura 3.8 apresenta esses resultados de forma visual, através do gráficos do tipo *box-plot* de cada um dos experimentos para as métricas avaliadas.

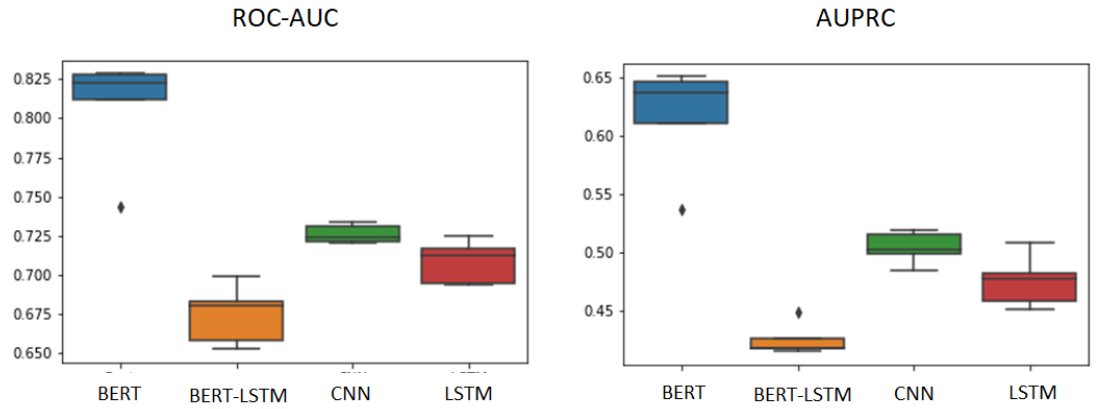

Figura 3.8: Comparação dos Resultados dos Modelos

Uma questão que merece ser observada é o fato de que a arquitetura que apresentou melhores resultados (modelo BERT) foi a que utilizou menor quantidade de textos (número de palavras) durante os processos de treinamento e de inferência, visto que, a arquitetura BERT tradicional só trata textos de até 510 tokens. Esse fato pode indicar que a arquitetura em questão seja a mais eficaz para esse caso específico, mas também pode ser um indicativo de que as partes iniciais dos textos das denúncias sejam as mais relevantes para o processo de classificação em questão, visto que, as partes dos textos que são utilizadas são sempre as iniciais.

Já quanto às redes CNN e LSTM, estas apresentaram resultados intermediários, sendo que, a rede CNN teve melhor desempenho do que a rede LSTM. Esse resultado corrobora com as conclusões apresentadas em [11], que fez um estudo comparativo entre esses dois tipos de rede para o caso de classificação de textos de solicitação de informações. As solicitações de informações, apesar de apresentarem tamanhos de textos menores, possuem características semelhantes às dos textos das denúncias, uma vez que em ambos os casos os textos são escritos por cidadãos em situações de interação com o governo. PAIVA *et al.* [11] apontam que para o tipo de texto em questão, as redes CNN seriam as mais indicadas. No entanto, é importante ressaltar que esse trabalho não avaliou o desempenho de redes que utilizavam o modelo pré-treinado BERT.

Por fim, a arquitetura que apresentou o pior desempenho foi a que combina os modelos BERT com uma camada LSTM. Caso a hipótese de que os textos iniciais das denúncias sejam mais relevantes para o processo de classificação, isso poderia justificar o desempenho pior dessa arquitetura. Nessa situação, a saída do primeiro modelo BERT (que trata os 510 tokens iniciais da denúncia) seria a mais informativa, e as demais saídas dos outros 7 modelos BERT utilizados acabariam por prejudicar o processo de classificação como um todo.

Sendo assim, a fim de testar se essa hipótese é verdadeira, executou-se novamente os experimentos com a rede CNN, que foi a que apresentou o segundo melhor desempenho, só que, no teste em questão, os experimentos foram configurados para somente considerarem as 510 palavras iniciais dos textos das denúncias, a fim de possibilitar a comparação dos resultados com o modelo treinado com textos de até 4080 palavras.

A Tabela 3.5 apresenta os resultados desse experimento específico, enquanto a Figura 3.9 apresenta um gráfico que confronta os resultados da rede CNN processando 4080 palavras com os resultados dessa mesma rede processando apenas as 510 palavras iniciais das denúncias.

Tabela 3.5: Resultados do Experimento com arquitetura CNN com 510 palavras

Iteração	**ROC-AUC**	**AUPRC**
1	0,7245	0,5071
2	0,7203	0,5138
3	0,7266	0,5120
4	0,6931	0,4715
5	0,7073	0,4962
Média	**0,7144**	**0,5001**

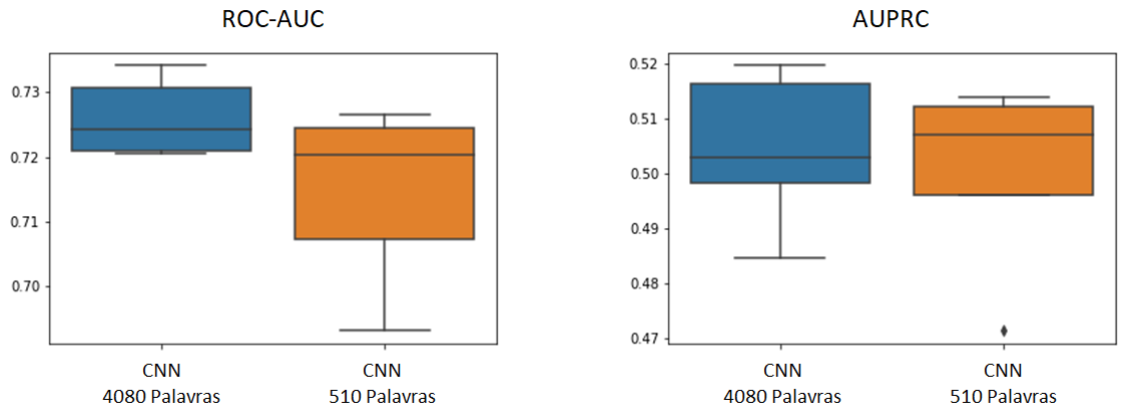

Figura 3.9: Comparação dos Resultados dos Modelos CNN (4080 e 510 palavras)

Como pode ser observado na Tabela 3.5 e na Figura 3.9, o desempenho dos dois experimentos (com 4080 palavras e com 510 palavras) foi bem semelhante, o que

corrobora com a ideia de que os 510 tokens iniciais das denúncias são suficientes para realizar a classificação de aptidão das denúncias.

Dessa forma, uma possível explicação para esse fenômeno pode ser o fato de que o texto original das denúncias sempre vem antes dos textos oriundos dos anexos, e o processo de obtenção dos textos dos anexos, em muitas situações, pode gerar distorções, pois ele não faz distinção de cabeçalhos, rodapés, textos fora de contexto e outras condicionantes que podem prejudicar a coerência do texto como um todo.

No entanto, é importante ressaltar que não se pode desconsiderar os textos dos anexos, pois em muitas situações, essa é a única fonte de informação disponível para o processo de classificação, uma vez que existem denúncias que não possuem textos originais, ou casos em que esses textos são muito genéricos (como por exemplo: "vide arquivos anexos").

Sendo assim, as análises apontam que para o caso específico do problema de classificação textual de aptidão de denúncias, a arquitetura mais adequada a ser utilizada é a que emprega o modelo BERT em sua estrutura original. Assim, mesmo com a limitação de se considerar apenas os 510 tokens iniciais, essa arquitetura demonstra melhores resultados, quando confrontada com as demais arquiteturas avaliadas.

3.5 Conclusão

Esse capítulo fez a análise de quatro arquiteturas de rede que foram propostas para resolver o problema de classificação de aptidão de denúncias. Sendo assim, realizaram-se experimentos em que foram treinadas e avaliadas diferentes arquiteturas e divisões de dados.

Cada experimento foi executado cinco vezes, sendo que a arquitetura que demonstrou o melhor resultado foi a que utilizava o modelo BERT puro, seguido pelas arquiteturas CNN e LSTM. Por fim, a arquitetura que apresentou o pior resultado foi a que utilizava uma combinação de modelos BERT com camada LSTM.

Os estudos também demostraram que as 510 palavras iniciais dos textos das denúncias eram suficientes para realizar a classificação de aptidão de denúncias.

Capítulo 4

Sumarização de Denúncias: Proposta e Avaliação de Métodos de Geração de Resumos

Uma denúncia pode conter informações que não são relevantes para o processo de classificação de aptidão de denúncias. Sendo assim, formulou-se a seguinte hipótese: "se forem aplicadas técnicas de sumarização de denúncias antes de se gerar os modelos de classificação de aptidão de denúncias, então, esses modelos gerarão melhores resultados".

Dessa forma, o objetivo desse capítulo é propor e avaliar duas estratégias de sumarização de denúncias, a serem utilizadas na análise de aptidão de denúncias, a fim de avaliar se a hipótese formulada é verdadeira.

Os estudos desenvolvidos nesse capítulo visam responder a questão de pesquisa 2: "Como extrair as principais informações dos textos das denúncias, a fim de utilizá-las no processo de classificação textual?"

O restante desse capítulo está dividido da seguinte forma: a Seção 4.1 traz alguns trabalhos relacionados e a Seção 4.2 apresenta as metodologias de sumarização propostas. Já as Seções 4.3 e 4.4 relatam os experimentos e a análise dos seus resultados, respectivamente. Finalmente, a Seção 4.5 faz a conclusão do capítulo.

4.1 Trabalhos Relacionados

A tarefa de sumarização pode ser categorizada em dois métodos: extrativo e abstrativo. A sumarização extrativa seleciona frases do documento original para formar um resumo, enquanto a sumarização abstrativa interpreta o documento original e gera o resumo com outras palavras [55]. Como é muito difícil para a máquina produzir um resumo compreensível para os humanos, na prática, as abordagens extrativas

são mais utilizadas [56].

ABDEL-SALAM e RAFEA [55] definem 3 abordagens principais para a sumarização de textos: as abordagens estatísticas, as baseadas em grafos e as baseadas em *Deep Learning*.

As abordagens baseadas em estatísticas são as mais antigas, e começaram a ser desenvolvidas na década de 50. LUHN [57] sugeriu o uso de informações derivadas do cálculo de frequências das palavras e da sua distribuição no texto para calcular uma medida de significância dessas palavras.

Posteriormente, EDMUNDSON [58] parte da ideia de LUHN [57] e propõe a seleção automática de sentenças com maior potencial de transmitir o conteúdo dos textos. EDMUNDSON [58] sugere ainda a identificação de algumas palavras (*"cue words"*) que sinalizam conteúdos importantes.

Outra abordagem baseada em estatística é a apresentada em [59]. Os autores propõem uma estratégia de busca gulosa que classifica as sentenças de acordo com as frequências, a fim de definir os pesos das probabilidades das palavras e minimizar redundâncias.

STEINBERGER *et al.* [60] descrevem um método genérico de sumarização de texto que utiliza a técnica de análise semântica latente para identificar sentenças semanticamente importantes. Os autores aplicam decomposição de valores singulares sobre a matriz do documento para fazer a identificação das sentenças.

Com relação as abordagens baseadas em grafos, MIHALCEA e TARAU [61] sugerem a representação do documento como um grafo de sentenças, sendo que, as sentenças representam os nós do grafo e a similaridade entre as sentenças são representadas pelas arestas.

Outra abordagem baseada em grafos é apresentada em [62]. Esse trabalho utiliza o conceito de autovetores para auxiliar na criação dos grafos representativos dos textos.

KOSMAJAC e KEŠELJ [14] utilizam o algoritmo TextRank [61] para propor uma sumarização extrativa de notícias escritas no idioma sérvio. O processo começa com a concatenação dos textos dos artigos. Depois, divide-se o texto em frases individuais. Posteriormente, encontra-se a representação vetorial dessas sentenças. Então é utilizada a semelhança dos cossenos para gerar uma matriz de similaridade. Essa matriz é posteriormente convertida em um grafo. Por fim, as frases melhor classificadas são selecionadas para formar o resumo.

Nessa mesma linha, DONG *et al.* [63] também propõem uma forma de ranqueamento de frases baseada em grafos, a fim de sumarizar textos científicos. O modelo assume uma representação gráfica hierárquica de dois níveis para o documento e busca indicações de assimetrias posicionais para determinar a importância das sentenças.

Segundo os autores, os resultados dos experimentos sugerem que os padrões na estrutura do discurso são um forte sinal para determinar a importância das sentenças em artigos científicos.

Atualmente, o avanço das arquiteturas das redes neurais, e dos modelos de linguagem derivados da arquitetura *Transformers* [3] estão possibilitando o alcance de excelentes resultados nas tarefas de sumarização textual baseadas em *Deep Learning*.

Nesse sentido, MILLER [64] utiliza o BERT [4], um modelo de linguagem derivado da arquitetura *Transformres*, para encontrar a representação vetorial das sentenças dos textos. Após isso, o autor aplica um processo de clusterização nesses vetores. Por fim, são identificados os centroides de cada um dos *clusters* gerados e as sentenças cujos vetores estão mais próximos desses centroides são as que têm as melhores condições de representar o texto em questão.

GU *et al.* [65] tentam identificar frases de qualidade dentro de corpus de texto. Para isso, os autores utilizam o ROBERTA [33], outro modelo de linguagem derivado da arquitetura *Transformer* para identificar frases de qualidade. Os autores propõem a captura de frases de alta qualidade com base na força do relacionamento existente entre as palavras de uma mesma sentença.

Todos esses trabalhos têm em comum o fato dos textos sumarizados serem bem estruturados e escritos corretamente. Porém, os textos das denúncias nem sempre apresentam uma estrutura organizada e bem escrita e nem uma sequência coerente de ideias. Isso acontece por deficiências na escrita ou por problemas no processo de conversão dos diferentes formatos de arquivo (pdfs, apresentações, figuras) para o formato textual ou ainda pela junção (automática) de diferentes arquivos, que apesar de comporem uma mesma denúncia, muitas vezes não tratam de assuntos complementares ou correlatos.

Sendo assim, as principais contribuições dessa parte da pesquisa são: a identificação da estratégia de sumarização mais adequada para auxiliar na atividade de análise de aptidão de denúncias e a verificação se aplicação de técnicas de sumarização melhora o desempenho do classificador de aptidão de denúncias.

4.2 Propostas de Geração de Resumos

Nessa seção são propostas duas técnicas de sumarização de textos para a geração de resumos de denúncias. Uma das técnicas utiliza uma abordagem estatística e se baseia em frequência de palavras e a outra utiliza uma abordagem de *Deep Learning* e emprega o modelo de linguagem BERT [4]. As próximas subseções detalham as estratégias propostas.

4.2.1 Sumarização pela Frequência de Palavras

O método de sumarização pela frequência de palavras baseia-se em um levantamento estatístico das frases mais relevantes do texto e anexos das denúncias. O método em questão está dividido em 5 passos: pré-processamento dos textos, cálculo da frequência ponderada dos *tokens*, identificação das frases, cálculo de importância das frases e seleção de frases.

A ideia principal dessa metodologia é que palavras que aparecem mais vezes no texto tendem a ser mais importantes. Sendo assim, aquelas frases que possuem tais palavras serão consideradas mais relevantes para o contexto e terão condições de transmitir melhor a ideia que está sendo passado no texto.

A Figura 4.1 ilustra o processo de geração de resumos pela frequência de palavras, sendo que, os números nos círculos pretos representam a ordem em que as atividades ocorrem.

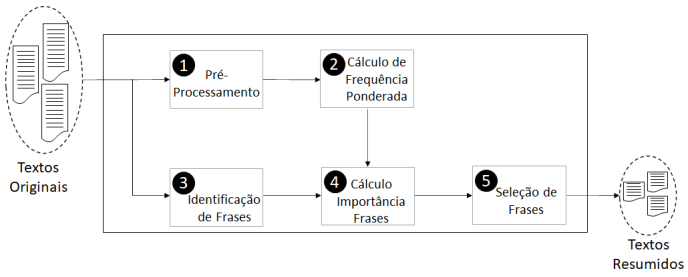

Figura 4.1: Processo de Sumarização pela Frequência de Palavras

Durante o pré-processamento, é realizado um tratamento no texto de forma que todas as letras sejam passadas para o formato de letra minúscula e que todos os sinais de acentuação sejam retirados. Esse tratamento faz-se necessário para que palavras iguais grafadas de formas diferentes não recebam tratamentos distintos. Nessa fase, também são retiradas as *stopwords* e os sinais de pontuação.

O passo seguinte é o cálculo da frequência ponderada de cada uma das palavras que compõem o texto pré-processado. Sendo assim, o texto é *tokenizado* e posteriormente calcula-se a frequência absoluta (quantidade de ocorrências) de cada *token* no texto tratado. Uma vez realizada a contagem dos *tokens*, deve-se identificar o *token* com maior número de ocorrências, sendo que tal *token* receberá o valor 1 e os demais receberão valores de frequências proporcionais. Assim, caso o *token* mais frequente tenha aparecido 10 vezes e um outro *token* tenha aparecido 4 vezes, o *token* mais frequente receberá o valor 1 e o outro o valor de 0,4 (4/10).

Na fase de identificação das frases, os textos originais (sem o pré-processamento) são divididos em sentenças. A fase seguinte faz o cálculo de importância de cada uma das frases dentro do texto. Sendo assim, percorre-se cada um dos *tokens* dessas frases e verifica-se a pontuação de tais *tokens* (de acordo com o valor da frequência

ponderada calculado na segunda fase do processo). A importância de cada frase será dada pela soma dos valores de frequência ponderada dos *tokens* que a compõem. Sendo assim, quanto maior for o valor dessa soma, maior será a importância da frase para o texto em questão.

O passo final consiste na ordenação em forma decrescente de importância das frases e na seleção das n frases mais importantes, sendo n um parâmetro de entrada do algoritmo, que indica o número de frases desejadas para compor o resumo.

Cabe ressaltar que as sentenças que irão compor o texto resumido devem aparecer na mesma ordem em que aparecem no texto original, de forma que a coerência do texto seja preservada.

4.2.2 Sumarização pelo BERT

A segunda técnica de sumarização proposta utiliza o modelo de linguagem pré-treinado BERT [4]. Essa técnica é composta por cinco etapas.

A primeira etapa é responsável por dividir o texto em sentenças. Depois aplica-se o modelo BERT a cada uma dessas sentenças a fim de se obter o vetor de *embedding* correspondente a cada uma delas. Esses vetores de *embeddings* são as representações vetoriais das respectivas sentenças, sendo que essas representações preservam a carga semântica do texto. Assim, sentenças com significados semelhantes tendem a ser representadas por vetores próximos no espaço vetorial em questão. Da mesma forma, sentenças com significados muito diferentes devem ser representadas por vetores distantes nesse mesmo espaço vetorial.

A partir da representação vetorial de todos as sentenças do texto, calcula-se o vetor médio, a fim de se identificar o vetor que representa a ideia central do texto em questão. Por fim, seleciona-se as n sentenças com menores distâncias para esse vetor médio. Essas sentenças selecionadas são as que melhor representam o texto e por isso são escolhidas para compor o resumo a ser criado. A Figura 4.2 ilustra esse processo.

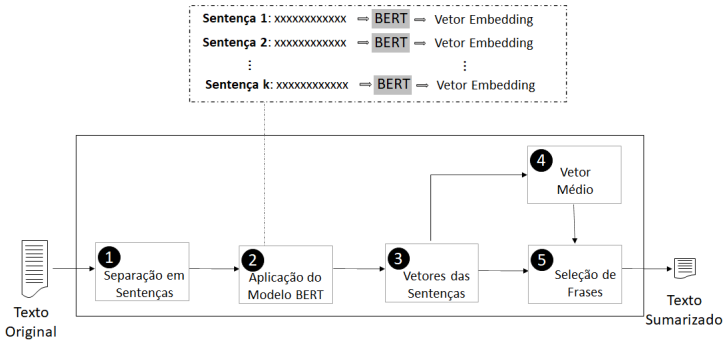

Figura 4.2: Processo de Sumarização pelo BERT

Cabe ressaltar que n é um parâmetro externo do algoritmo, e representa o número de sentenças que irá compor o resumo. Também é importante destacar que as sentenças que compõem o resumo aparecem na mesma ordem em que elas aparecem no texto original, a fim de preservar a coerência do texto de saída.

4.3 Experimentos

Os processos de sumarização foram avaliados através da análise dos resultados de modelos de classificação textual gerados a partir dos textos originais e dos resumos obtidos pelas metodologias de sumarização propostas. Assim, utilizou-se um conjunto de denúncias previamente rotuladas (como aptas ou não aptas), gerou-se resumos dos textos dessas denúncias (pelos dois métodos apesentados) e gerou-se diferentes modelos de classificação a partir dos textos originais e resumidos.

Para a geração dos modelos foram utilizadas duas abordagens distintas: uma que gerava modelos de classificação baseados em *Deep Learning* (que utilizava uma rede neural com uma camada de BERT, uma camada de *dropout* e uma camada de saída com dois neurônios) e outra que gerava modelos de classificação baseados em técnicas tradicionais de Aprendizado de Máquina, que utilizava vetorização TF-IDF[1] e o algoritmo LightGBM.

A fim de evitar conclusões que não reflitam a realidade, os experimentos foram repetidos 30 vezes. Sendo assim, foram gerados 30 modelos distintos, com diferentes divisões de dados de treino e de teste para cada uma das estratégias testadas. Cabe ressaltar que todos os testes foram executados utilizando-se 80% dos dados para o treinamento e 20% para a avaliação. A Figura 4.3 ilustra a arquitetura dos experimentos.

Durante os experimentos não se investiu na otimização de hiperparâmetros nem na arquitetura da rede utilizada, pois o objetivo dos testes era verificar o comportamento das soluções propostas em modelos obtidos a partir das mesmas configurações, mas com textos de entrada distintos: textos originais e textos resumidos (obtidos por ambas as técnicas apresentadas).

Para esse experimento, utilizou-se uma base composta por uma amostra de 580 denúncias aptas e 580 denúncias consideradas não aptas, selecionadas aleatoriamente. Sendo assim, cada registro dessa base se referia a uma denúncia e era composto pelo texto da denúncia (concatenado com os seus anexos) e um rótulo, que servia para identificar se a denúncia tinha sido considerada como apta ou não.

Os experimentos foram executados com três tipos de textos: textos originais, textos resumidos pela metodologia que utilizava o modelo BERT e textos resumidos

[1]TF-IDF (Term Frequency – Inverse Document Frequency) é uma técnica de vetorização que indica a importância de uma palavra em um documento em relação a toda coleção de documentos.

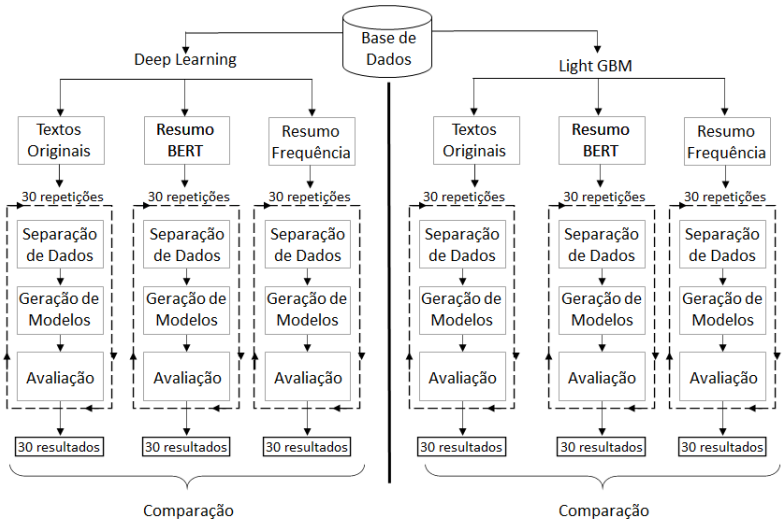

Figura 4.3: Arquitetura dos Experimentos

pela metodologia que utilizava a frequência de palavras. Os resumos foram gerados pela seleção das 10 frases mais relevantes de cada um dos textos originais (de acordo com as metodologias apresentadas). A opção pelo número de 10 frases se deu pelo fato desse parâmetro ser o que gera o número de *tokens* mais próximo do limite de *tokens* admitido como entrada para o modelo BERT.

4.4 Análise dos Resultados

A métrica utilizada para a avaliação dos resultados dos experimentos foi a ROC-AUC. Sendo assim, foram utilizados os resultados obtidos pela métrica ROC-AUC durante as 30 execuções de cada um dos experimentos, para as duas estratégias analisadas: modelos obtidos com *Deep Learning* e modelos obtidos por técnicas tradicionais de Aprendizado de Máquina.

4.4.1 Análise dos resultados dos modelos de *Deep Learning*

A Figura 4.4 apresenta os *Box-plots* com a consolidação dos resultados das 30 execuções de cada um dos experimentos com os modelos de classificação obtidas com algoritmos de *Deep Learning*.

Analisando-se a Figura 4.4, verifica-se-se que o desempenho dos modelos de classificação obtidos a partir dos textos originais foi melhor do que os obtidos a partir dos textos resumidos. Da mesma forma, percebe-se que os resultados dos modelos obtidos com o resumo pelo BERT foram melhores do que os resultados obtidos pelos resumos gerados pela frequência de palavras.

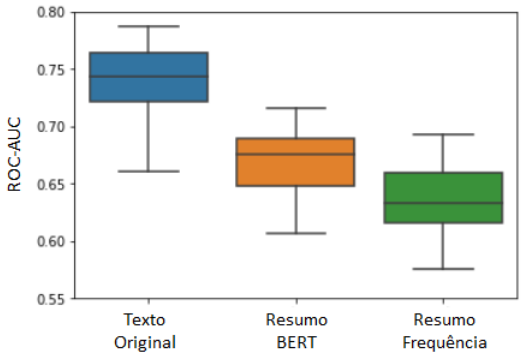

Figura 4.4: Box-plots com os resultados da métrica ROC-AUC para os experimentos com *Deep Learning*

Comparação dos Resultados dos Modelos dos Resumo pelo BERT e do Resumo pela Frequência de palavras

Apesar dos resultados apresentados sugerirem que o modelo obtido com o resumo BERT é melhor do que o modelo obtido pelo resumo pela frequência de palavras, faz-se necessária a realização de testes estatísticos que possam indicar se essas diferenças são estatisticamente relevantes ou se são fruto do acaso.

Sendo assim, optou-se pela realização do teste t para 2 amostras, que tem o objetivo de comparar os valores médios dessas amostras. No entanto, esse tipo de teste parte do pressuposto de que a distribuição dos valores analisados obedece a uma distribuição normal. Sendo assim, a primeira análise a ser feita é a análise de normalidade dos dados estudados.

Para essa análise de normalidade, realizou-se o teste de Shapiro Wilk. Esse teste considera as seguintes hipóteses nula (H_0) e alternativa (H_1)

$$\begin{cases} \boldsymbol{H_0} : \text{a amostra provém de uma população normal} \\ \boldsymbol{H_1} : \text{a amostra não provém de uma população normal} \end{cases}$$

Todos os testes realizados nesse estudo consideraram o nível de significância de 5% ($\alpha = 0,05$). Sendo assim, caso o valor-p[2] obtido pelo teste estatístico seja menor do que esse nível de significância, pode-se dizer que a hipótese nula (H_0) foi rejeitada e consequentemente a hipótese alternativa (H_1) deve ser considerada como verdadeira. Da mesma forma, caso o valor-p seja maior do que o nível de significância, não é possível rejeitar a hipótese nula, e consequentemente, considera-se esta como sendo verdadeira.

Os testes de Shapiro Wilk aplicados aos resultados dos experimentos com os resumos com o BERT e com o resumo pela frequência de palavras obtiveram valor-

[2]O valor-p indica a probabilidade de se observar uma diferença tão grande ou maior do que a que foi observada sob a hipótese nula [66].

p de 0,73 e 0,57, respectivamente. Como esses valores são maiores que o nível de significância considerado, a hipótese nula não deve ser rejeitada. Logo, pode-se dizer que ambas as amostras provêm de uma população normalmente distribuída.

Uma vez comprovada a normalidade dos dados, outra análise necessária para a realização do teste t é a análise das variâncias das amostras a serem comparadas, pois o teste t varia de acordo com a homogeneidade ou não das variâncias dessas amostras. Para a análise da variância, executou-se o teste F de Levene, que apresenta as seguintes hipóteses nula e alternativa:

$$\begin{cases} \boldsymbol{H_0} : \text{as variâncias são iguais} \\ \boldsymbol{H_1} : \text{as variâncias são diferentes} \end{cases}$$

Nesse teste, obteve-se o valor-p de 0,51, que é maior do que o nível de significância considerado. Logo, a hipótese nula não pode ser rejeitada e consequentemente as variâncias das amostras analisadas podem ser consideradas como homogêneas.

Por fim, após a comprovação de atendimento dos pré-requisitos, aplicou-se o teste t unicaudal para duas amostras, com as seguintes hipóteses formuladas, onde μ indica a média dos resultados obtidos:

$$\begin{cases} \boldsymbol{H_0} : \mu_{\text{Resumo BERT}} \leq \mu_{\text{Resumo Frequência}} \\ \boldsymbol{H_1} : \mu_{\text{Resumo BERT}} > \mu_{\text{Resumo Frequência}} \end{cases}$$

O teste em questão retornou um valor-p bem próximo de zero. Sendo assim, como o valor-p foi menor do que o nível de significância ($\alpha = 0,05$), a hipótese nula deve ser rejeitada, e consequentemente, assume-se a hipótese alternativa como verdadeira. Logo, pode-se afirmar, com um grau de confiança de 95% (visto que o nível de significância é de 5%), que as médias dos resultados dos modelos obtidos com o resumo pelo modelo BERT realmente é maior do que a média dos resultados obtidos com o resumo pela frequência de palavras.

Comparação dos Resultados dos Modelos do texto Original com dos Resumo BERT

Para essa análise também foi realizado o teste *t*. Dessa forma, mais uma vez, fez-se necessária a verificação dos pressupostos do referido teste. Quanto a análise de normalidade, só se analisou a normalidade dos resultados obtidos com o texto original, visto que, a normalidade dos resultados do resumo com BERT já havia sido verificada na subseção anterior.

O teste de Shapiro Wilk retornou um valor-p de 0,31, que é maior do que o nível de significância considerado, o que aponta para a normalidade dos dados.

Para o teste F de Levene, que avalia a homogeneidade de variâncias (no caso entre

os resultados obtidos pelos experimentos dos resumos feitos pelo BERT e dos textos originais), obteve-se o valor-p de 0,60, que indica que as variâncias das amostras são homogêneas.

Por fim, aplicou-se o teste t para duas amostras com a seguinte formulação das hipóteses:

$$\begin{cases} \boldsymbol{H_0} : \mu_{\text{Texto Original}} \leq \mu_{\text{Resumo BERT}} \\ \boldsymbol{H_1} : \mu_{\text{Texto Original}} > \mu_{\text{Resumo BERT}} \end{cases}$$

Para esse teste, obteve-se um valor-p bem próximo de zero, o que nos permite rejeitar a hipótese nula e concluir que a média do resultado obtido pelo texto Original realmente é maior do que a média obtida pelo modelo do Resumo BERT (com nível de confiança de 95%).

4.4.2 Análise dos resultados dos Modelos Tradicionais

A Figura 4.5 apresenta os *Box-plots* dos resultados com modelos tradicionais. Conforme pode ser observado, novamente, os resultados indicam um melhor desempenho dos modelos obtidos pelo texto original, seguidos pelos modelos obtidos pelo resumo BERT e pelo resumo pela frequência de palavras. Porém, para sustentar tais conclusões, foram realizados testes estatísticos.

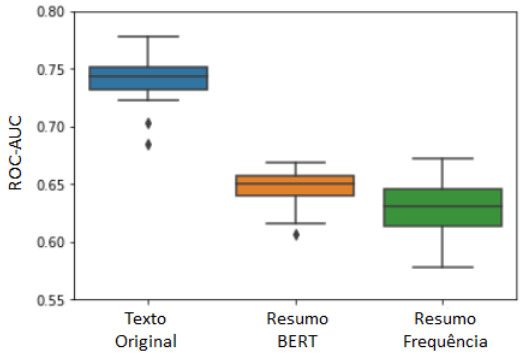

Figura 4.5: Box-plots com os resultados da Métrica ROC-AUC para os experimentos com modelos tradicionais

Comparação dos Resultados dos Modelos de Resumo pelo BERT e de Resumo pela Frequência de Palavras

Para essa comparação também se utilizou o teste t. Desta forma, foram realizados os testes de normalidade e de homogeneidade de variância, citados anteriormente.

Para o teste de Shapiro Wilk, obteve-se o valor-p de 0,14 e 0,92 para as distribuições do resumo pelo BERT e do resumo pela frequência de palavras, respectivamente.

Logo, pode-se concluir que essas distribuições são normais (visto que esses valores são maiores do que o nível de significância de 0,05).

Com relação a análise de homogeneidade de variância das amostras, realizou-se o teste F de Levene e obteve-se o valor-p de 0,04, que é menor do que o nível de significância considerado ($\alpha = 0,05$), fazendo com que a hipótese nula seja rejeitada e que a hipótese alternativa seja considerada como verdadeira. Ou seja, as variâncias das amostras não são homogêneas.

Por fim, aplicou-se o teste t para o caso de amostras com variâncias diferentes, sendo que as hipóteses consideradas foram as seguintes:

$$\begin{cases} \boldsymbol{H_0} : \mu_{\text{Resumo Original}} \leq \mu_{\text{Resumo BERT}} \\ \boldsymbol{H_1} : \mu_{\text{Resumo Original}} > \mu_{\text{Resumo BERT}} \end{cases}$$

O valor-p para o teste em questão foi de 0,0002. Logo, como esse valor é menor do que o nível de significância considerado, pode-se dizer que a hipótese nula foi rejeitada e assume-se a hipótese alternativa como verdadeira. Portanto, a média dos resultados obtidos pelos modelos do resumo pelo BERT é maior do que a média dos resultados obtidos pelos modelos do resumo pela frequência de palavras (com um grau de confiança de 95%).

Comparação dos Resultados dos Modelos do texto Original com dos Resumo BERT

Para aplicar o teste t na comparação entre o resultado dos modelos obtidos pelo texto original com os resultados dos modelos obtidos pelo resumo pelo BERT, fez-se a análise desses resultados, pela aplicação do teste de Shapiro Wilk. Nesse caso, obteve-se o valor-p de 0,02, que é menor do que o nível de significância. Esse resultado indica que a hipótese nula deve ser rejeitada. Logo, os resultados do texto original não proveem de uma distribuição normal e por isso não se pode aplicar o teste t.

Sendo assim, optou-se pela aplicação de um teste não paramétrico, que possui menos pressupostos com relação aos dados testados. Utilizou-se o teste de Wilcoxon, que em vez de comparar as médias das amostras, compara as suas medianas. Para a realização desse teste, formulou-se as seguintes hipóteses:

$$\begin{cases} \boldsymbol{H_0} : Mediana_{\text{Texto Original}} \leq Mediana_{\text{Resumo BERT}} \\ \boldsymbol{H_1} : Mediana_{\text{Texto Original}} > Mediana_{\text{Resumo BERT}} \end{cases}$$

O teste retornou um valor-p de bem próximo de zero. Com isso, a hipótese nula foi rejeitada e consequentemente optou-se pela hipótese alternativa, que diz que a mediana dos resultados dos modelos com o texto original é maior do que a mediana

dos resultados dos modelos obtidos com o resumo pelo BERT.

4.4.3 Conclusão da Análise

Tanto o experimento com *Deep Learning*, quanto o experimento com modelos tradicionais indicaram um desempenho superior para os modelos obtidos com o texto original. Uma possível explicação para essa constatação pode ser o fato de que todo processo de sumarização pressupõe alguma perda de informação em relação ao texto original. Essa perda de informação acaba oferecendo menos recursos para o classificador decidir sobre a aptidão da denúncia.

Isso indica que a hipótese de pesquisa: "se forem aplicadas técnicas de sumarização de denúncias antes de se gerar os modelos de classificação de aptidão de denúncias, então, esses modelos gerarão melhores resultados" é falsa.

No entanto, também ficou comprovado que o resumo obtido pela metodologia do BERT apresentou melhores resultados do que o resumo obtido pela metodologia da frequência de palavras. Isso demonstra que para a sumarização de textos denúncia, o método do resumo pelo BERT é o mais apropriado.

4.5 Conclusão

Esse capítulo apresentou a proposta de dois métodos de sumarização textual a serem aplicados nos textos de denúncias: um baseado na utilização do modelo BERT e outro baseado em frequência de palavras.

Esses métodos foram avaliados pelos resultados dos modelos de classificação textual de aptidão de denúncias gerados a partir de textos originais e resumidos. Os modelos de classificação foram gerados com a utilização de duas abordagens: uma baseada em *Deep Learning*, com a utilização do modelo BERT, e outra baseada em métodos tradicionais de Aprendizado de Máquina, utilizando vetorização TF-IDF e o algoritmo *Light GBM*. Cada experimento foi realizado 30 vezes. Por fim, foram aplicados teste estatísticos aos conjuntos de resultados, a fim de se comparar as performances.

Os testes apontaram, com um nível de confiança de 95%, que o desempenho dos modelos gerados pelos textos originais era melhor do que o desempenho dos modelos gerados pelos resumos, e que o desempenho dos modelos gerados pelo resumo do BERT era melhor do que o gerado pelo resumo pela frequência de palavras.

Capítulo 5

Obtenção de dados estruturados a partir de textos de denúncias

Esse capítulo apresenta os estudos realizados para responder à questão de pesquisa 3: "como considerar informações externas aos textos das denúncias no processo de classificação de aptidão de denúncias?".

Para responder essa questão de pesquisa, o capítulo propõe uma metodologia que faz a extração de variáveis de textos de denúncias e posteriormente correlaciona essas variáveis com outras bases de dados, a fim de obter novas informações, que são utilizadas para a classificação das denúncias.

Como apresentado no Capítulo 1, durante a análise de aptidão de denúncias, valida-se as informações narradas nos textos e verifica-se outros fatos comprobatórios em fontes de dados externas. Assim, deve-se identificar informações específicas nos textos e correlacioná-las com dados externos.

Logo, um processo automatizado deve identificar e extrair certas informações do texto das denúncias. FELDMAN *et al.* [67] apresentam quatro tipos básicos de elementos que podem ser extraídos de textos: entidades, atributos, fatos e eventos. Dessa forma, essa parte da pesquisa utiliza a ideia introduzida em [67] para buscar por esses elementos nos textos das denúncias e depois tentar correlacioná-los com outras informações em bases de dados externas.

Sendo assim, o objetivo desse capítulo é propor uma metodologia para a extração e o enriquecimento de informações de textos de denúncias, a fim de utilizá-las juntamente com os textos originais para realizar a classificação automatizada de aptidão de denúncias.

Para isso, formulou-se a seguinte hipótese: "se forem empregadas técnicas de extração e enriquecimento de variáveis em textos de denúncias, então, os resultados da classificação textual dessas denúncias podem melhorar".

O restante desse capítulo está dividido da seguinte forma: a Seção 5.1 apresenta uma revisão da literatura. Já as Seções 5.2 e 5.3 descrevem a proposta e a aplicação

dessa proposta, respectivamente. As Seção 5.4 e 5.5 apresentam os experimentos realizados para validar a proposta e a análise dos resultados desses experimentos, respectivamente. Finalmente, a Seção 5.6 traz a conclusões do capítulo.

5.1 Trabalhos Relacionados

Em muitas situações, a classificação textual carece de informações que não estão narradas diretamente nos textos ou dependem de formas alternativas para a representação desses textos.

Nesse sentido, alguns trabalhos tentam endereçar essa necessidade de informações externas para a classificação textual. WU *et al.* [68] propõem a criação de um dicionário de gírias. Esse dicionário é utilizado durante a atividade de classificação de sentimentos de mensagens postadas em mídias sociais. Dessa forma, ao analisar um determinado texto, pode-se consultar o dicionário e encontrar a palavra correspondente a uma determinada gíria citada na mensagem. Esse procedimento facilita a tarefa de classificação de sentimentos.

KARTHIKEYAN *et al.* [69] propõem a classificação de textos da Internet utilizando apenas partes do conteúdo desses textos. Os autores extraem os documentos da Web e recuperam os conteúdos relatados com base em *queries*, agregações e transformações de dados, a fim de obter uma representação estruturada do dado anteriormente desestruturado.

Após isso, os autores utilizam uma técnica de eliminação recursiva de variáveis com o objetivo de selecionar o melhor subconjunto de variáveis candidatas. Por fim, o modelo de classificação é gerado com algoritmos da aprendizagem de máquina tradicionais para categorizar as páginas da web.

LI [70] apresenta um estudo para a área jurídica cujo objetivo é encontrar correspondência entre uma previsão legal e a descrição de um evento escrito na língua inglesa. Para isso, o autor utiliza um método de classificação de textos legais baseado em palavras características.

As palavras características dos documentos são calculadas pelo TF-IDF, porém, o autor propõe a introdução de fatores de correção ao TF-IDF baseados na estatística chi-quadrada e na posição das palavras.

Como as previsões legais podem ser extraídas dos documentos de julgamento com precisão, torna-se possível estabelecer uma relação entre a previsão legal e as palavras características.

Nessa mesma linha de utilizar modificações do TF-IDF, LIU *et al.* [71] sugerem a realização da classificação textual utilizando uma representação vetorial de palavras que combina o word2vec com o TF-IDF. O artigo propõe que a representação da palavra seja dada pela multiplicação do peso da palavra, obtido com o TF-IDF,

pelo vetor de representação word2vec dessa mesma palavra. Sendo assim, cada texto passa a ser representado pelo acúmulo de todos os vetores de palavras e tal representação pode ser utilizada na classificação dos textos.

COUSSEMENT e VAN DEN POEL [72] sugerem um modelo de classificação que utiliza como variáveis informações sobre o estilo de escrita dos textos. Sendo assim, são utilizadas como variáveis a quantidade de verbos, pronomes, palavras únicas, palavras de negação, palavras afirmativas, artigos, preposições, entre outras informações derivadas do texto. Os autores alegam que a utilização destes elementos de estilo linguístico, juntamente com vetores de palavras, pode aumentar o desempenho do classificador.

Todos esses trabalhos propõem formas alternativas para a representação dos textos a serem classificados. No entanto, essa representação sempre fica limitada a conteúdos extraídos dos próprios textos, ou de dicionários, sem buscar informações derivadas ou correlacionadas.

Dessa forma, a principal contribuição dessa parte da pesquisa é a proposta de uma metodologia que possibilita a representação dos textos com elementos do próprio texto e com outros elementos correlacionados, extraídos de fontes de dados externas. Sendo assim, essa metodologia propicia a obtenção de um conjunto de dados estruturados que pode ser utilizado para melhorar o desempenho da classificação textual.

5.2 Proposta

Para o entendimento completo dos textos, muitas vezes é necessário o conhecimento de outras informações que não estão presentes nos textos. Da mesma forma, em algumas situações, o Processamento de Linguagem Natural carece de informações que não estão nos textos analisados. Tal situação ocorre na classificação de textos de denúncias.

Essa seção apresenta uma metodologia que faz a extração de variáveis dos textos, e complementa com outras variáveis que trazem informações oriundas de diversas bases de dados.

O método proposto extrai os 4 tipos de elementos citados por FELDMAN *et al.* [67]: entidades, atributos, fatos e eventos. No entanto, ao invés de realizar essa extração utilizando apenas os textos originais, propomos a utilização de fontes externas (57 bancos de dados), a fim de identificarmos novos elementos relacionados aos elementos extraídos. Sendo assim, tem-se dois tipos de elementos: os de 1º nível (extraídos diretamente dos textos) e os de 2º nível (oriundos das fontes de dados externas). A metodologia proposta é composta de 5 fases e é ilustrada na Figura 5.1. As próximas subseções descrevem o papel de cada uma dessas fases.

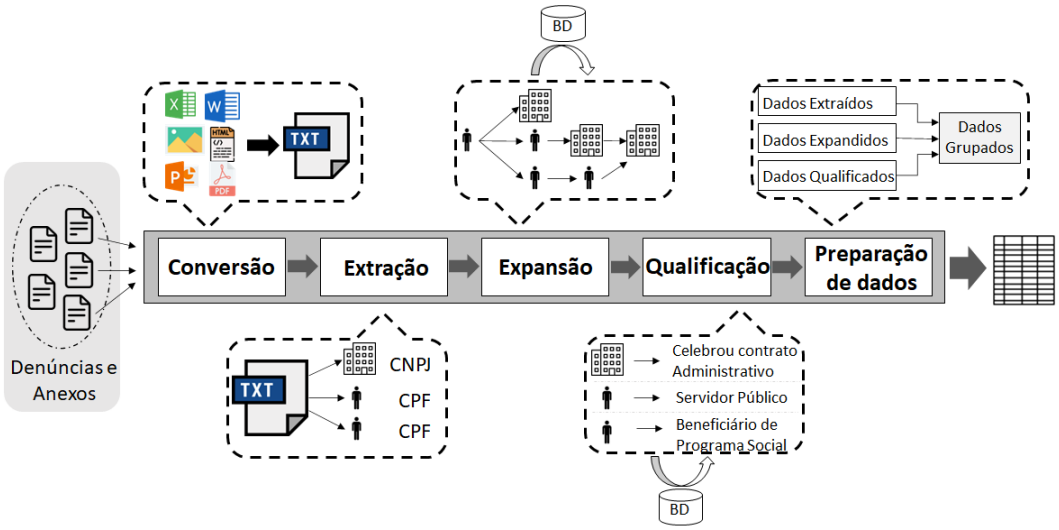

Figura 5.1: Processo de extração e enriquecimento de variáveis

5.2.1 Conversão

A primeira fase do processo é a Conversão. A principal função dessa fase é ler os arquivos anexos (das denúncias) e extrair as informações desses arquivos.

Os arquivos anexos podem vir em diferentes formatos (fotos, planilhas, arquivos pdf scaneados como figuras, apresentações). Dessa forma, os anexos são transformados em um formato textual, para que possam ser utilizados nas fases posteriores do processamento. Essa transformação é feita pelo Apache Tika.

Após essa atividade, os textos extraídos dos anexos são concatenados aos textos originais das denúncias, a fim de que esses sejam tratados de forma única.

5.2.2 Extração

Nessa fase é realizada a extração de informações dos textos das denúncias. Essa metodologia faz a identificação e extração de um conjunto de elementos considerados relevantes para a atividade de tratamento de denúncias. Sendo assim, alguns exemplos de elementos extraídos são:

- Nomes de pessoas
- CPFs
- CNPJs
- Números de NIS[1]
- Números de contratos

[1]O NIS é o Número de Identificação Social atribuído pela Caixa Econômica Federal para identificar pessoas cadastradas em programas sociais do governo.

- Números de convênios
- Valores monetários
- Palavras fortes

O reconhecimento de nomes é feito conforme a proposta apresentada em [73]. Nesse trabalho, os autores fazem o reconhecimento de entidades nomeadas em Português empregando o modelo BERT [4], sendo que, nesse caso, utiliza-se uma versão em Português do modelo BERT [74]. Já o reconhecimento de CPFs, CNPJs, valores monetários, números de NIS, contratos e convênios é realizado pela utilização de expressões regulares. A opção pelo uso de expressões regulares para esse tipo de identificação se deu pelo fato desses elementos apresentarem formatos numéricos bem característicos, de fácil identificação nos textos, tornando o desempenho da identificação melhor do que o obtido por Aprendizado de Máquina.

Os elementos considerados "Palavras fortes" são palavras, ou expressões, consideradas relevantes no contexto de análise de denúncias (por exemplo "fraude", "corrupção", "superfaturamento"). A busca desses tipos de elementos é feita diretamente nos textos, sendo que, a definição dessas palavras e expressões é feita por especialistas da área de negócio.

Uma vez identificados, todos esses elementos são armazenados em um banco de dados para serem utilizados nas outras fases do processamento.

5.2.3 Expansão

A Expansão utiliza as entidades identificadas na fase anterior, e tenta encontrar novas informações a respeito delas em outras bases de dados. Essa busca tem o objetivo de validar a existência das entidades identificadas, bem como descobrir novos elementos que tenham vínculos com as entidades identificadas anteriormente.

Sendo assim, para um determinado CNPJ, identificado no texto da denúncia, a expansão realiza duas atividades. Primeiro, ela verifica, em bases de dados institucionais, se esse CNPJ realmente é um CNPJ válido. Posteriormente, são buscados outros elementos derivados desse CNPJ. Por exemplo, identifica-se todas as pessoas que constam como sócias desse CNPJ. Da mesma forma, para um determinado número de contrato, verifica-se se esse contrato é valido. Posteriormente, são levantadas as partes envolvidas no contrato (contratante e contratado), o objeto do contrato, prazo de vigência do contrato.

Esse procedimento é executado para todas as entidades identificadas nos textos. Dessa forma, passa-se a ter 2 tipos de elementos: os de primeiro nível (identificados diretamente nos textos) e os de segundo nível (derivados dos anteriores).

5.2.4 Qualificação

A fase de qualificação tem o objetivo de fazer a qualificação das entidades identificadas nas fases anteriores. Sendo assim, para um determinado CPF, é verificado se ele pertence a um servidor público, se é beneficiário de algum programa social e outros qualificadores de pessoa física. Para um determinado CNPJ, é verificado se esse é parte de algum contrato ou convênio, se está cadastrado no CEPIM[2], dentre outros qualificadores.

Dessa forma, todas as entidades identificadas passam por esse processo de qualificação, sendo que, cada tipo de entidade possui um conjunto específico de qualificadores que são verificados.

5.2.5 Preparação dos Dados

Durante a preparação dos dados, agrega-se todas as informações obtidas nas fases anteriores, a fim de se criar um conjunto de dados estruturados que possa ser utilizado no treinamento do modelo.

Dessa forma, cada tipo de informação levantada nas fases anteriores passa a ser representada como uma coluna e cada denúncia como uma linha de uma tabela. Sendo assim, o valor referente a cada um dos atributos para cada denúncia é dado pela aplicação de alguma função de agregação, que varia de acordo com o tipo de atributo (por exemplo: contagem, soma, média).

Logo, cada denúncia passa a ser representada por um conjunto de dados estruturados, que foram obtidos nas fases anteriores do processamento.

5.3 Aplicação da Proposta

O processo de extração de variáveis gerou 122 variáveis. Alguns exemplos dessas variáveis obtidas a partir dos textos e das bases de dados externas são: quantidade de empresa com CNPJ no cadastro de empresas inadimplentes, quantidade de servidores públicos citados, quantidade de contratos vigentes, soma dos valores contratados, quantidade de pessoas beneficiadas por programas sociais citadas.

Sendo assim, o primeiro teste a ser realizado é se essas variáveis têm algum poder de predição quanto à análise de admissibilidade das denúncias. Logo, o objetivo inicial foi a criação de um modelo de classificação de denúncias a partir desses dados estruturados.

[2]Cadastro de Entidades Privadas Sem Fins Lucrativos Impedidas (CEPIM): relação de entidades privadas sem fins lucrativos que estão impedidas de celebrar convênios, contratos de repasse ou termos de parceria com a Administração Pública Federal.

Sendo assim, dividiu-se o conjunto de dados em duas partes: 80% dos dados para o treinamento do modelo e 20% para o teste. Dessa forma, gerou-se um modelo de classificação, a partir desses dados estruturados, utilizando-se o algoritmo Random Forest [75].

A avaliação desse modelo atingiu o valor de 0,7508 para a métrica ROC-AUC. Portanto, conclui-se que as variáveis obtidas têm capacidades preditivas para a realização da análise de aptidão de denúncias. No entanto, é bem provável que nem todas as variáveis criadas tenham a mesma importância para a geração do modelo. Sendo assim, o passo seguinte foi a identificação do conjunto de variáveis que realmente eram relevantes para o processo de classificação das denúncias.

Dessa forma, utilizou-se o próprio algoritmo Random Forest [75] para se identificar a importância de cada uma das variáveis no modelo gerado. Após a identificação da importância de cada uma das variáveis, colocou-se essas variáveis em ordem decrescente de importância. Depois disso, gerou-se 122 modelos com diferentes números de variáveis, sendo que o modelo com menos variáveis possuía apenas uma (apenas a variável considerada mais relevante) e o com mais variáveis possuía 122 (número total de variáveis). Assim, cada modelo sempre era gerado com base nas variáveis assumidas como mais relevantes.

Essa estratégia foi utilizada para identificar quantas variáveis seriam necessárias para a geração do modelo. A Figura 5.2 apresenta o gráfico da métrica ROC-AUC para os modelos gerados com as diferentes quantidades de variáveis.

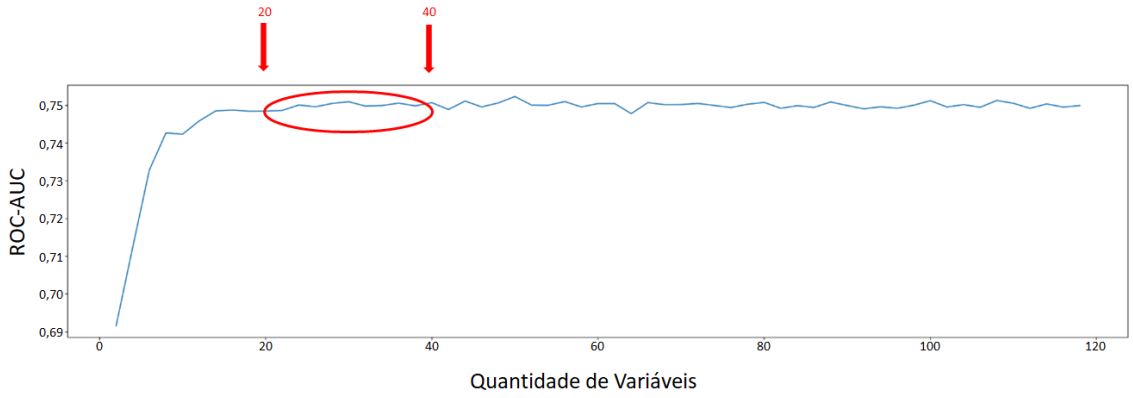

Figura 5.2: Desempenho dos modelos por Número de Variáveis

No gráfico, o eixo horizontal representa o número de variáveis utilizadas pelos modelos. Sendo assim, os valores variam de 1 (número mínimo de variáveis utilizada por um modelo) a 122 (número máximo de variáveis utilizadas por um modelo – total de variáveis geradas). Já o eixo vertical representa o valor da métrica ROC-AUC alcançada pelo modelo em questão.

Como pode ser observado, no gráfico da Figura 5.2, o desempenho dos modelos com menos variáveis são piores. Esse desempenho vai melhorando com a inserção

de novas variáveis. No entanto, entre 20 e 40 variáveis, o desempenho se estabiliza. Desse modo, a partir desse momento a adição de novas variáveis não interfere no desempenho do modelo.

Logo, conclui-se que 40 variáveis são suficientes para o processo de classificação. Dessa forma, a fim de deixar o modelo mais simples, selecionou-se apenas as 40 variáveis mais relevantes para o modelo final. O modelo gerado com as 40 variáveis, utilizando o algoritmo Random Forest [75], obteve um valor de 0,7507 para a métrica ROC-AUC.

Cabe ressaltar que apesar do modelo utilizar apenas 40 variáveis, optou-se por não realizar nenhuma modificação no processo de geração de variáveis, pois as demais variáveis geradas, apesar de não serem utilizadas pelo modelo, podem ser utilizadas pela área de negócio para outros tipos de análises que devem ser feitas com as denúncias.

Uma vez definidas as variáveis a serem utilizadas, optou-se pela utilização de um outro algoritmo para a geração do modelo que utilizava as variáveis estruturadas. Nesse caso, utilizou-se o algoritmo *LightGBM* [76], que é uma versão simplificada do *XGBoost* [77]. Esses algoritmos também utilizam um comitê de árvores de decisão, no entanto, costumam obter melhores resultados do que o Random Forest, pelo fato de apresentar algumas melhorias.

O modelo gerado com o algoritmo LightGBM, utilizando as 40 variáveis selecionadas, obteve um valor de 0,7784 para a métrica ROC-AUC.

5.4 Experimentos

O objetivo do processo de obtenção de variáveis é melhorar o desempenho da classificação de denúncias. Os estudos apresentados no Capítulo 3 apontaram que o classificador de denúncias que obteve melhor desempenho foi o que utilizava a arquitetura com o modelo de linguagem pré-treinado BERT. Esse modelo alcançou um valor de ROC-AUC médio de 0,8070.

Sendo assim, os experimentos realizados nessa seção foram estruturados de forma que fosse possível a comparação entre os resultados obtidos pelo modelo textual puro e os resultados obtidos por um modelo combinado, que resulta da composição do modelo textual com o modelo estruturado, sugerido na Seção 5.3, conforme ilustrado na Figura 5.3.

Esse modelo combinado é obtido por uma média ponderada entre as predições realizadas pelo modelo textual e as predições realizadas pelo modelo estruturado. Sendo assim, realizou-se algumas combinações de pesos (para o modelo textual e para o modelo estruturado), a fim de se identificar aquela que apresentaria melhores resultados.

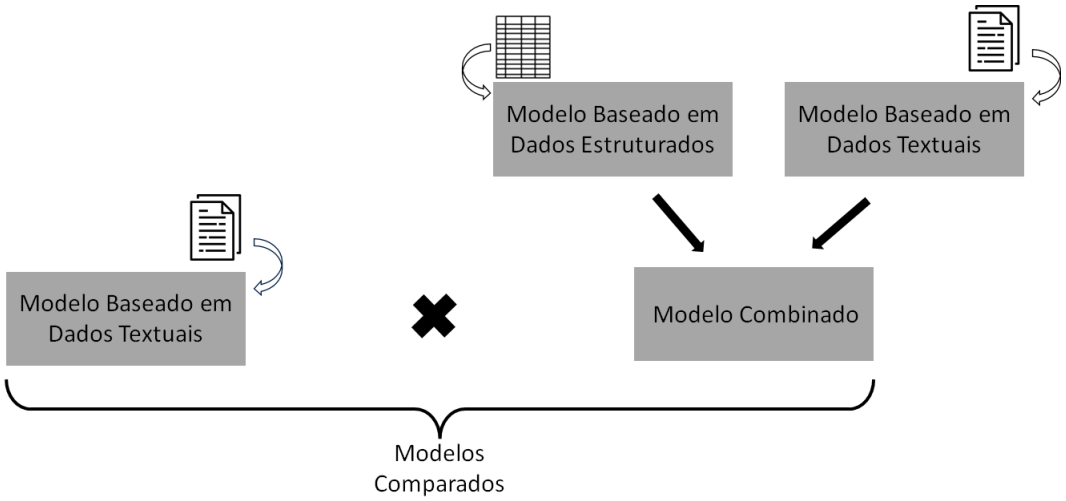

Figura 5.3: Modelos Avaliados

A Tabela 5.1 mostra os resultados médios da métrica ROC-AUC e AUPRC para um conjunto de 30 repetições de testes executados para cada uma das opções avaliadas.

Tabela 5.1: Resumo dos valores obtidos

Identificação do Modelo	Peso dos modelos		AUPRC	ROC-AUC
	Estruturado	Textual		
Estruturado (original)	1	0	0,5917	0,7675
Textual (original)	0	1	0,6087	0,8059
Combinado 1	1	1	0,6358	0,8106
Combinado 2	2	1	0,6372	0,8039
Combinado 3	**1**	**2**	**0,6372**	**0,8132**
Combinado 4	3	2	0,6326	0,8072
Combinado 5	2	3	0,6373	0,8126

Como pode ser observado, a combinação que apresentou melhores resultados foi a combinação 3, que utilizava peso 2 para o modelo textual e 1 para o modelo estruturado. Sendo assim, essa foi a combinação adotada para o prosseguimento dos experimentos.

Dessa forma, realizou-se uma série de experimentos a fim de verificar se as diferenças entre os resultados do modelo textual e do modelo combinado eram estatisticamente relevantes, ou se eram fruto do acaso.

Sendo assim, os experimentos foram repetidos 30 vezes, de forma que fossem gerados 30 modelos textuais e 30 modelos combinados, com diferentes divisões de conjuntos de dados de treino e de teste. Esses 30 modelos criados, para cada uma das estratégias avaliadas (modelo textual e modelo combinado), geraram 30 resultados diferentes para a métrica ROC-AUC.

A Figura 5.4 ilustra a estrutura dos experimentos realizados.

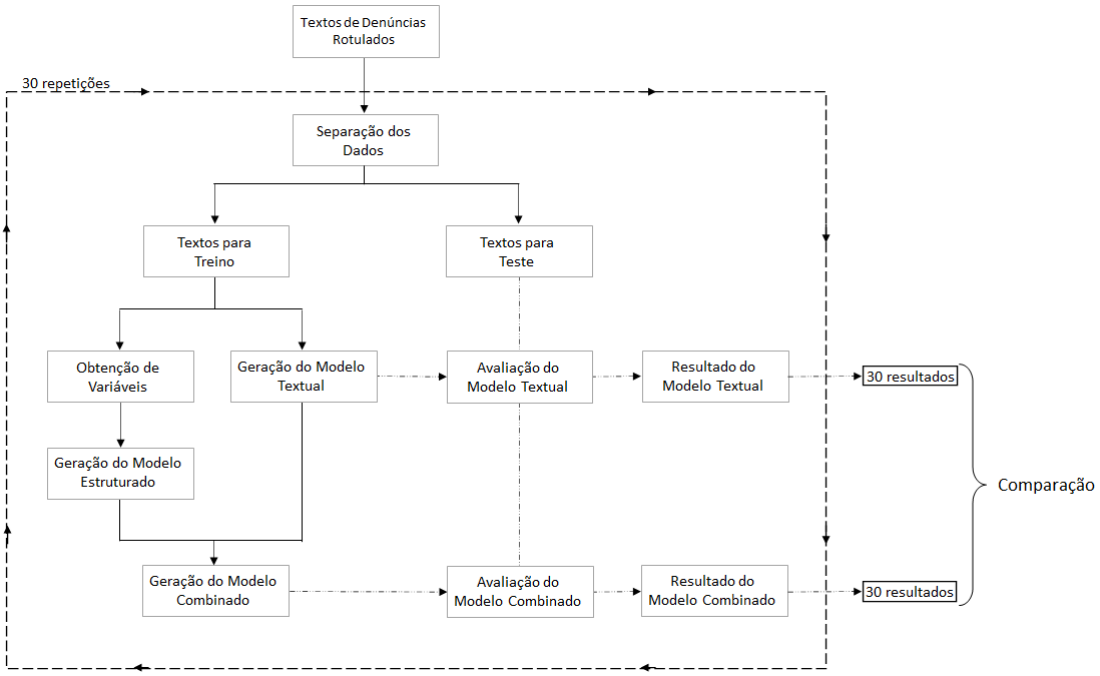

Figura 5.4: Estrutura dos Experimentos

5.5 Análise dos Resultados

A Figura 5.5 apresenta os *Box-plots* com a consolidação dos resultados da métrica ROC-AUC das 30 execuções de cada um dos experimentos realizados: 30 resultados do modelo textual e 30 resultados do modelo combinado.

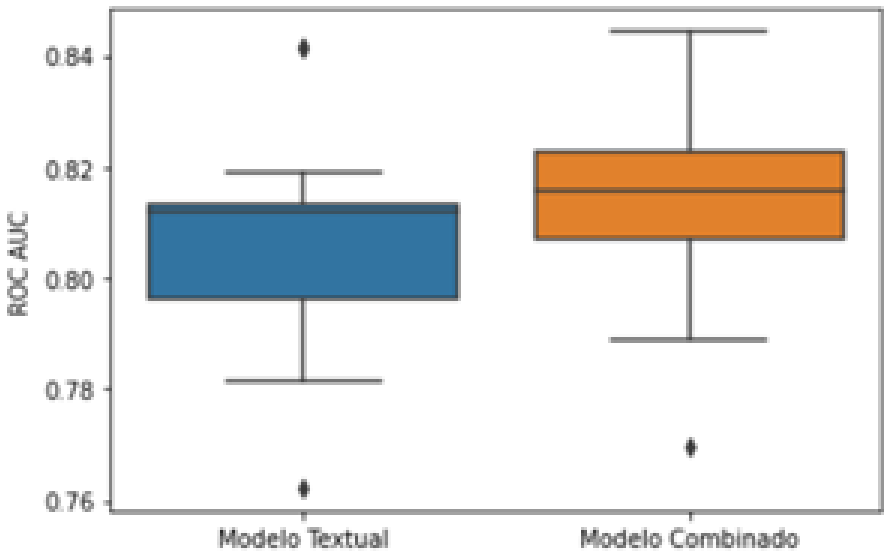

Figura 5.5: Box-plots com os resultados da Métrica ROC-AUC para os experimentos com o modelo Textual e com o modelo Combinado

Os resultados obtidos pelo Modelo Combinado geraram uma ROC-AUC média de 0,8132, enquanto a ROC-AUC média do modelo textual puro foi de 0,8059, o

que sugere que o processo de geração de variáveis trouxe ganhos para o processo de classificação. Essa mesma percepção também pode ser obtida pela análise visual dos gráfico *Box-plots* da Figura 5.5. No entanto, para a confirmação dessa percepção, faz-se necessária a realização de testes estatísticos que indiquem se tal diferença é estatisticamente significativa.

Para isso, optou-se pela realização do teste t para duas amostras, que faz a comparação dos valores médios dessas amostras. Porém, conforme apresentado no Capítulo 4 (Seção 4.4), esse tipo de teste exige que as amostras a serem comparadas, apresentem as características de uma distribuição normal.

Sendo assim, o primeiro passo foi a análise da normalidade das distribuições formadas pelos conjuntos de resultados obtidos nas 30 repetições dos experimentos (para cada estratégia avaliada). Essa análise foi feita com a utilização do Teste de Shapiro Wilk, que apresenta as seguintes hipóteses nula (H_0) e alternativa (H_1).

$$\begin{cases} \boldsymbol{H_0} : \text{a amostra provém de uma população normal} \\ \boldsymbol{H_1} : \text{a amostra não provém de uma população normal} \end{cases}$$

O nível de significância considerado foi de 5% ($\alpha = 0,05$). Dessa forma, se o valor-p, obtido no teste, for menor do que 5%, então a hipótese nula (H_0) é rejeitada e consequentemente a hipótese alternativa (H_1) é considerada como verdadeira. Por outro lado, se o valor-p for maior do que 5%, não é possível rejeitar a hipótese nula, e consequentemente, considera-se que a amostra obedece a uma distribuição normal.

Ao se executar o teste de Shapiro Wilk para os resultados dos experimentos do modelo textual e do modelo combinado, chegou-se, respectivamente, aos seguintes valores-p: 0,05145 e 0,04539. Portanto, levando-se em consideração os parâmetros estabelecidos para o teste, pode-se dizer que o conjunto dos resultados do modelo textual segue uma distribuição normal (valor-p $> \alpha$, considera-se a hipótese nula), mas o conjuntos dos resultados do modelo combinado não (valor-p $< \alpha$, considera-se a hipótese alternativa).

Logo, como os dados não atenderam a um dos requisitos do teste t, optou-se pela realização de um teste não paramétrico para fazer a comparação entre os resultados obtidos pelos dois modelos avaliados.

Dessa forma, utilizou-se o teste de Wilcoxon. Esse teste também tem o objetivo de comparar duas amostras. Porém, nesse caso, ao invés de comparar a média dessas amostras, ele compara as medianas das amostras avaliadas. Sendo assim, formulou-se as seguintes hipóteses:

$$\begin{cases} \boldsymbol{H_0} : Mediana_{\text{Modelo Combinado}} \leq Mediana_{\text{Modelo Textual}} \\ \boldsymbol{H_1} : Mediana_{\text{Modelo Combinado}} > Mediana_{\text{Modelo Textual}} \end{cases}$$

O teste retornou um valor-p muito próximo de zero, ou seja, muito menor do que o nível de significância de 5% considerado. Com isso, rejeita-se a hipótese nula e considera-se a hipótese alternativa. Sendo assim, pode-se afirmar, com 95% de confiabilidade, que a mediana dos resultados do modelo combinado é maior do que a mediana dos resultados do modelo textual puro.

Dessa forma, conclui-se que o processo de extração de variáveis e de obtenção de variáveis externas trouxe ganho para a classificação de aptidão de denúncias. Sendo assim, pode-se dizer que a hipótese formulada: "se forem empregadas técnicas de extração e enriquecimento de variáveis em textos de denúncias, então, os resultados da classificação textual dessas denúncias podem melhorar" é verdadeira.

5.6 Conclusão

Este capítulo apresentou uma metodologia para a extração e o enriquecimento de variáveis em textos de denúncias. A metodologia propõe a identificação e extração de certos elementos dos textos das denúncias. Posteriormente, é feita a validação desses elementos e busca-se por outros elementos, derivados dos primeiros, utilizando para isso bases de dados externas. Por fim, a metodologia fornece um conjunto de dados estruturados.

O objetivo da metodologia proposta é a geração de um conjunto de dados estruturados capaz de representar as denúncias que precisam ser classificados. Dessa forma, torna-se possível a criação de um modelo combinado, que utiliza tanto os textos originais das denúncias quanto as variáveis extraídas, para se obter um classificador que supera o desempenho do classificador baseado apenas nos textos originais das denúncias.

Para validar a metodologia, realizou-se um experimento que gerava dois tipos de modelos: um textual, gerado pelos textos originas das denúncias, e um modelo combinado, obtido pela combinação do modelo textual com um outro modelo, gerado com dados estruturados obtidos pela aplicação da metodologia proposta. O experimento foi repetido 30 vezes com diferentes divisões de dados de teste e de treino.

As análises concluíram, com um grau de confiança de 95%, que a metodologia proposta trouxe ganhos para o processo de classificação de aptidão das denúncias.

Capítulo 6

Conclusão

Este capítulo faz a conclusão da tese e tem o objetivo de apresentar como o modelo foi colocado em produção, as principais contribuições da pesquisa, as possibilidades de trabalhos futuros e algumas considerações finais.

6.1 Modelo em Produção

Os estudos apresentados até o Capítulo 5 definiram uma proposta de modelo de classificação de denúncias. No entanto, a ideia desse modelo não é fornecer uma classificação binária, enquadrando a denúncia em uma das duas classes possíveis (Apta ou Não Apta). Cada denúncia recebe um valor de probabilidade, sendo que, quanto maior for esse valor, maior é a probabilidade da denúncia ser classificada como Apta; e quanto menor for esse valor, maior é a probabilidade dela ser classificada como Não Apta.

A Figura 6.1 ilustra o processo de classificação de novas denúncias pelo modelo gerado. Dessa forma, essa figura retrata uma situação em que o modelo recebe como entrada 3 denúncia com os seus respectivos anexos (denúncias D1, D2 e D3). Nessa situação, o modelo realiza as classificações e apresenta como saída valores de probabilidades, cada um referente a uma das denúncias de entrada.

Esse comportamento do modelo torna possível a definição de regiões de valores para os quais a aptidão ou não aptidão de uma denúncia possa ser definida com graus de confiabilidade aceitáveis pela ouvidoria. Valores altos de probabilidade indicam que o modelo tem alto grau de confiança para classificar a denúncia como apta. Da mesma forma, valores baixos de probabilidade indicam que o modelo tem alto grau de confiança para considerar a denúncia como não apta. No entanto, valores intermediários de probabilidade, próximos de 50%, indicam que o modelo está com dificuldade para definir a aptidão da denúncia.

Dessa forma, o modelo pode ser configurado para classificar de forma automática as denúncias em que ele obtiver maior grau de confiabilidade e as demais denúncias

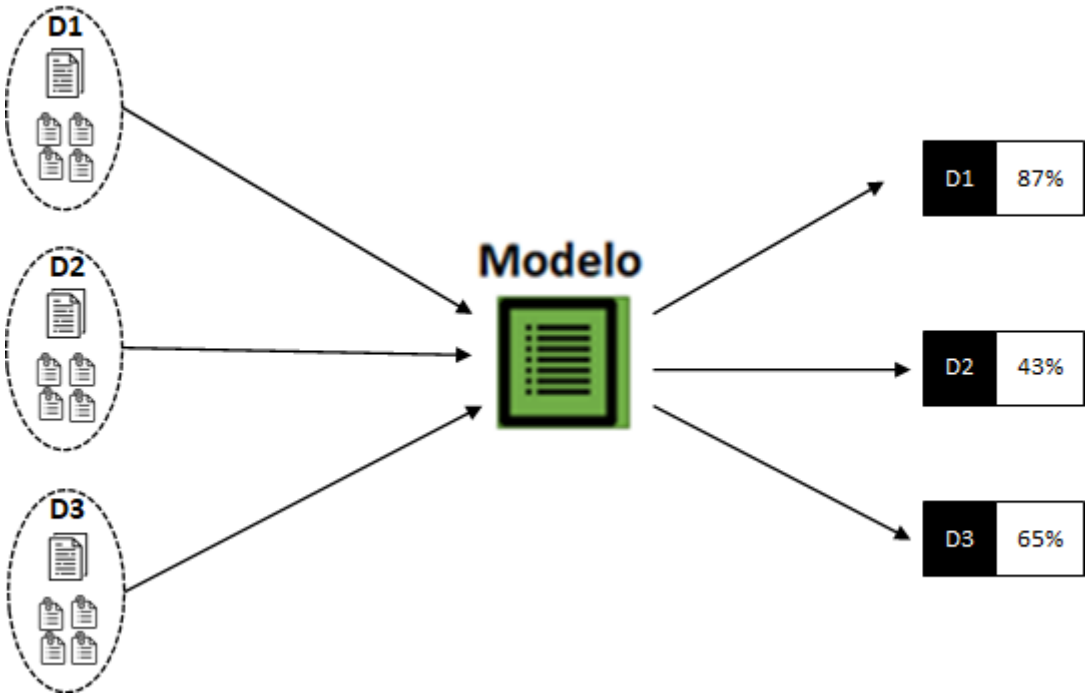

Figura 6.1: Processo de classificação de novas denúncias

cuja confiabilidade na classificação é baixa continuam sendo classificadas de forma manual. Tal procedimento possibilita a diminuição da carga de trabalho dos servidores da ouvidoria, sem o comprometimento da qualidade das classificações realizadas. A Figura 6.2 ilustra esse processo.

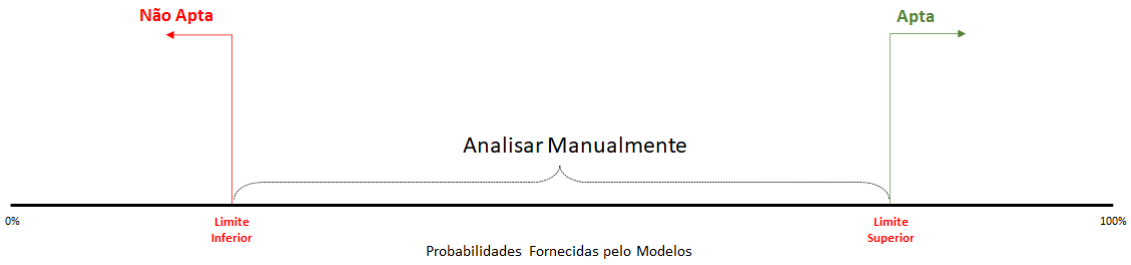

Figura 6.2: Proposta de utilização do modelo

Como pode ser observado na Figura 6.2, todas as denúncias cujo modelo fornece uma probabilidade menor do que o valor de limite inferior são automaticamente consideradas como não aptas. Da mesma forma, todas as denúncias cujo modelo fornece uma probabilidade maior do que o limite superior, são automaticamente consideradas como aptas. Já as denúncias cuja probabilidade fornecida pelo modelo ficam entre os limites inferior e superior continuam sendo classificadas de forma manual.

Sendo assim, a definição desses limites está diretamente relacionada com a confiabilidade desejada. No entanto, valores altos de confiabilidade implicam em menos

denúncias classificadas automaticamente. Por outro lado, maiores quantidades de denúncias classificadas de forma automática implicam em menor confiabilidade dos resultados.

Dessa forma, realizou-se um estudo para subsidiar a escolha desses valores limites de forma mais embasada.

6.1.1 Estudo sobre os Valores Limites

O estudo realizado fez um comparativo utilizando diferentes limites (de aptidão e de não aptidão). Dessa forma, analisou-se a quantidade de denúncias que se enquadravam nesses diferentes critérios e o percentual de acerto obtido pelo modelo para cada um desses limites.

O estudo foi dividido em duas partes: uma referente aos limites para considerar uma denúncia como não apta (limite inferior) e outra referente aos limites para se considerar uma denúncia como apta (limite superior).

Não Aptidão

Para o estudo dos limites de não aptidão, foram testados cinco valores limites: 10, 20, 30, 40 e 50. Dessa forma, para o valor limite de 10 considerou-se como não aptas todas as denúncias cuja probabilidade fornecida pelo modelo fosse menor do que 10%. Da mesma forma, para o limite de 20, considerou-se como não aptas todas as denúncias cuja probabilidade fornecida pelo modelo fosse menor do que 20%. Esse procedimento foi repetido sucessivamente até o limite de 50, situação em que foram consideradas como não aptas todas as denúncias cuja probabilidade fornecida pelo modelo fosse menor do que 50%.

A Figura 6.3 apresenta os resultados obtidos para cada um dos limites estudados, sendo que, as barras verticais indicam a percentagem do total de denúncias que se enquadram no referido limite. Já a linha na parte superior do gráfico indica as percentagens de acerto obtidas para cada um dos limites testados, ou seja, das denúncias que foram consideradas como não aptas, para o limite em questão, quantas realmente eram não aptas.

Pela análise do gráfico apresentado na Figura 6.3, percebe-se que para o limite de 10, consegue-se 100% de acerto para as denúncias consideradas como não aptas pelo modelo. Porém, esse limite só abrange cerca de 3,14% do total de denúncias recebidas.

Da mesma forma, caso se considere o limite de 20, obtém-se uma percentagem de 91,25% de acerto para os casos em que o modelo considerou a denúncia como não apta, sendo que, isso corresponde a 28,95% do total de denúncias.

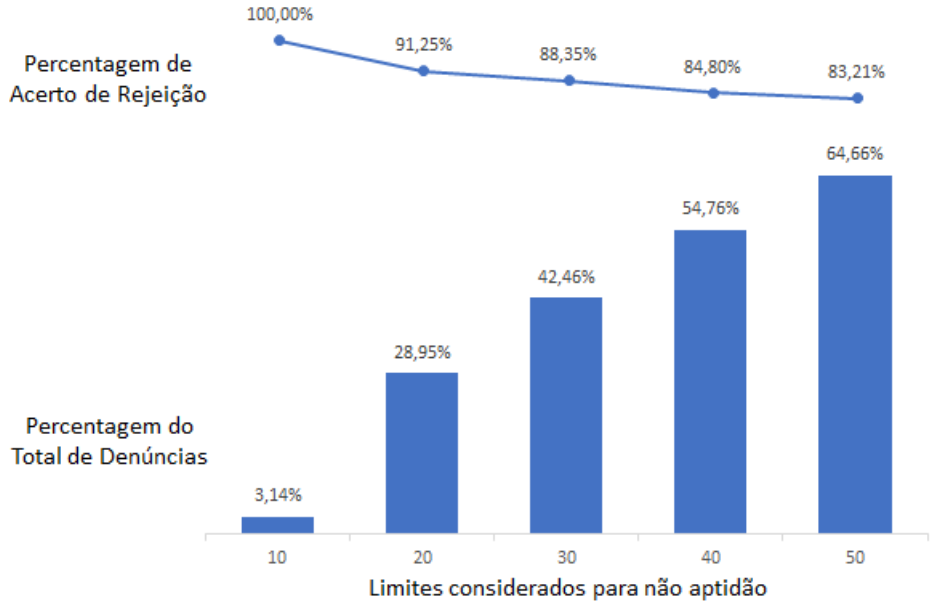

Figura 6.3: Resultado do Estudo dos limites de Não Aptidão de denúncias.

Esse mesmo tipo de análise pode ser feito nas demais partes do gráfico da Figura 6.3 (limites de 30, 40 e 50).

Aptidão

Para o estudo dos limites de aptidão, também foram testados 5 valores limites: 50, 60, 70, 80 e 90. Sendo assim, para o valor limite de 50, considerou-se como aptas todas as denúncias cuja probabilidade fornecida pelo modelo fosse maior que 50%. Da mesma forma, para o limite de 60, considerou-se como aptas todas as denúncias cuja probabilidade fornecida pelo modelo fosse maior do que 60%. Esse procedimento foi sucessivamente repetido até o limite de 90, situação em que foram consideradas como aptas todas as denúncias cuja probabilidade fornecida pelo modelo fosse maior que 90%. A Figura 6.4 apresenta os resultados obtidos no estudo de aptidão.

Pela análise do gráfico da Figura 6.4, percebe-se que para o limite de 90, obteve-se uma percentagem de acerto de 100%, para as denúncias consideradas aptas pelo modelo, porém, esse limite só enquadra 1,21% do total de denúncias. Já para o limite de 80, obtém-se uma percentagem de acerto de 83,33%, sendo que, esse limite abrange cerca de 10,13% do total de denúncias. Esse mesmo tipo de análise também pode ser feito para os demais valores limites apresentados no gráfico da Figura 6.4 (limites de 70, 60 e 50).

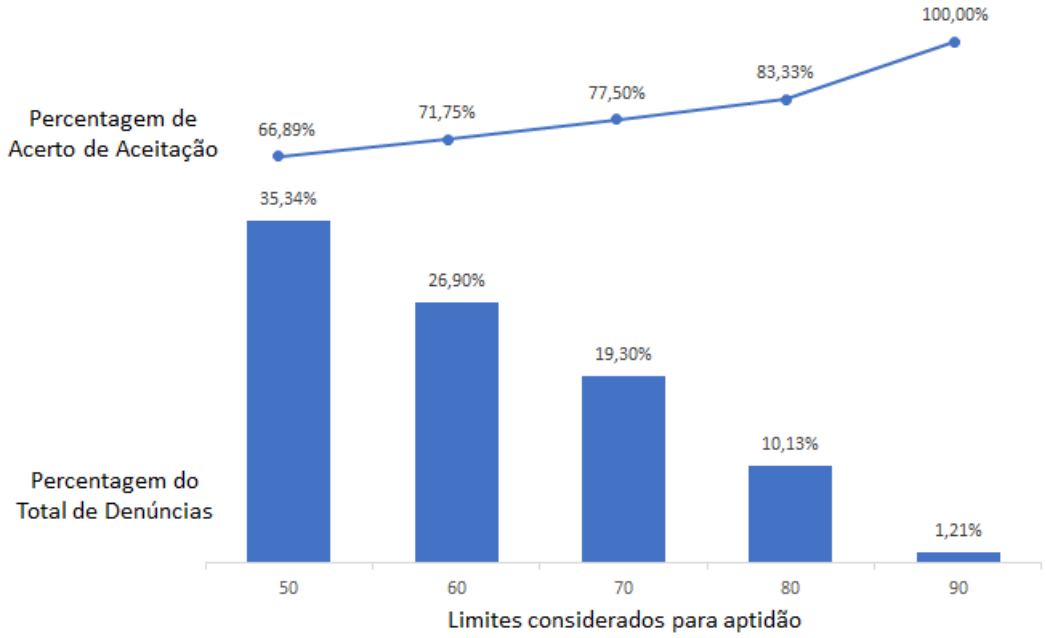

Figura 6.4: Resultado do Estudo dos limites de Aptidão de denúncias.

Conclusão do Estudo

Os resultados obtidos foram apresentados para a área de negócio (Ouvidoria Geral da União), a fim de que esses limites fossem definidos de acordo com as necessidades de confiabilidade dos resultados e a capacidade operacional do pessoal.

Cabe ressaltar que a decisão desses limites de corte foi atribuída para a área de ouvidoria, pelo fato dessa ser uma decisão exclusiva do negócio. Sendo assim, essa pesquisa apenas propôs a metodologia para garantir o grau de confiabilidade exigido pelo negócio e forneceu as informações necessárias para subsidiar a decisão.

Após as análises, a área de negócio decidiu que os limites mais apropriados seriam de 30% para as denúncias não aptas e de 80% para as denúncias aptas. Sendo assim, as denúncias cujo modelo fornecesse probabilidade menor que 30% seriam consideradas automaticamente como não aptas, sendo que, isso corresponde a cerca de 40% do total de denúncias e representa uma percentagem de acerto de 88,35%. Por outro lado, as denúncias cujo modelo fornecesse probabilidades maiores que 80% seriam automaticamente classificadas como aptas, sendo que, isso corresponde a cerca de 10% das denúncias e representa uma percentagem de acerto de 83,33% para as denúncias consideradas aptas. As denúncias que não se enquadrassem nesses limites continuariam a ser classificadas de forma manual. A Figura 6.5 ilustra essa configuração.

Dessa forma, a área finalística (ouvidoria) definiu os limites de acordo com os riscos que ela considerava convenientes e com base nesses limites o modelo foi confi-

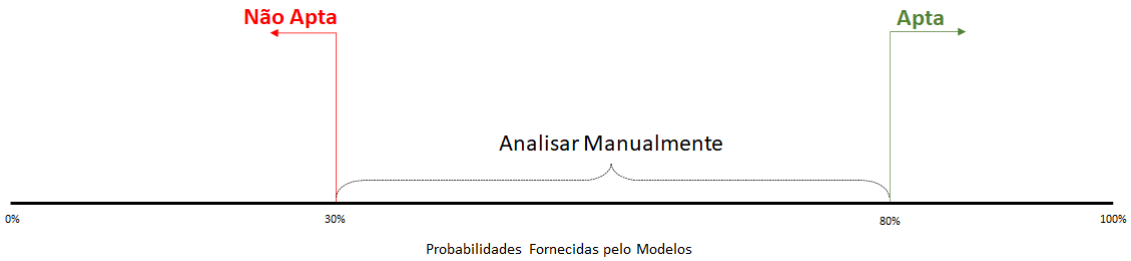

Figura 6.5: Limites para considerar uma denúncia como Apta ou Não Apta.

gurado. De acordo com essa configuração, cerca de 50% das denúncias passaram a ser classificadas de forma automática.

6.1.2 Dados Estruturados

Conforme apresentado no Capítulo 5, para a criação do modelo estruturado, várias informações são extraídas dos textos das denúncias, e de seus respectivos anexos. Essas informações podem ser nomes, CPFs, CNPJs, números de contratos, números de Convênio, números de NIS. A partir desses dados, pode-se obter outras informações derivadas, com base em consultas a banco de dados externos.

Todas essas informações, assim como as demais variáveis geradas pela metodologia proposta, podem ser de grande valia para a análise e apuração das denúncias. Tais informações são representações estruturadas de dados que antes estavam em um formato textual, sendo que, elas ainda são enriquecidas com dados que não estavam presentes no corpo da denúncia.

Sendo assim, todos os dados gerados durante o processamento também são repassados para a área de negócio, a fim de auxiliar nas análises e tratamentos posteriores das denúncias. Dessa forma, até as denúncias que não são classificadas automaticamente também são beneficiadas pela aplicação da metodologia proposta, visto que, os dados estruturados identificados pela metodologia trazem maior celeridade para a análise das denúncias que porventura tenham que continuar sendo analisadas de forma manual.

6.2 Contribuições

Essa pesquisa trouxe contribuições sociais e acadêmicas, além de ter gerado alguns artigos científicos.

As próximas subseções apresentam os detalhes dessas contribuições.

6.2.1 Contribuições Sociais

A principal contribuição social foi a entrega de uma versão do modelo de classificação de denúncias que já está sendo utilizada pela Ouvidoria Geral da União. Essa entrega estava prevista em uma das ações do Plano Anticorrupção do Governo Federal.

O Plano Anticorrupção é uma iniciativa do Governo Federal. Esse plano foi instituído para o período de 2020 a 2025 e tem como objetivo estruturar e executar ações para aprimorar, no âmbito do Poder Executivo Federal, os mecanismos de prevenção, detecção e responsabilização por atos de corrupção, avançando no cumprimento e no aperfeiçoamento da legislação anticorrupção e no atendimento de recomendações internacionais.

Nesse plano foram propostas 142 ações de curto e médio prazo a serem executadas pelo Poder Executivo Federal até o ano de 2025, sendo que, cada uma dessas ações possui uma data de entrega específica.

Dentre as ações previstas para a Controladoria Geral da União, está a Ação CGU 35, que prevê a criação da Ferramenta de Análise de Risco em Ouvidoria (FARO). Essa ação tem como objetivo o desenvolvimento de uma ferramenta para auxiliar na triagem e análise automatizada de denúncias na Plataforma de Ouvidoria e Acesso à Informação (Fala.BR), empregando técnicas de Processamento de Linguagem Natural e Aprendizado de Máquina.

Sendo assim, essa pesquisa teve um papel fundamental para a entrega da ação em questão. Os estudos e experimentos realizados nessa pesquisa foram utilizados para o desenvolvimento do modelo entregue. Dessa forma, uma versão do modelo de classificação de denúncias já está em produção, trazendo assim benefícios para o processo de tratamento das denúncias recebidas pela Ouvidoria Geral da União, e por conseguinte, para a sociedade como um todo.

Outro trabalho que também foi fruto dessa pesquisa e apresenta uma contribuição social foi a monografia intitulada "Identificação de Produtos em Descrições Textuais de Compras: uma proposta para portais de transparência pública". Essa monografia ficou em primeiro lugar no Prêmio Ministro Gama Filho 2019, cujo tema era tecnologias emergentes e o controle do estado.

O Prêmio Ministro Gama Filho é uma distinção feita anualmente a trabalhos acadêmicos que versam sobre temáticas de interesse da administração pública. Essa iniciativa tem por objetivo estimular a realização de estudos e pesquisas a serem apresentados sob a forma de monografias.

O objetivo dessa monografia era fazer a identificação dos produtos mais comprados pela Administração Pública, por meio da análise das descrições textuais de compras, apresentadas nos portais de transparência. Sendo assim, o problema tratado consistia na identificação de forma automatizada de produtos a partir das es-

pecificações textuais que eram usadas para caracterizá-los nas descrições dos gastos que são apresentados nos portais de transparência pública.

O trabalho em questão apresentou uma metodologia capaz de identificar os gastos descritos nos portais de transparência, permitindo assim a comparação dos gastos de forma mais direta. O emprego da metodologia proposta permitiu a identificação de disparidades nas compras feitas, facilitando assim o controle dos gastos públicos.

6.2.2 Contribuições Acadêmicas

A principal contribuição dessa pesquisa foi o desenvolvimento de um modelo de classificação de denúncias escritas na Língua Portuguesa. No entanto, outras contribuições intermediárias foram geradas durante o desenvolvimento dos estudos.

Dentre essas contribuições, pode-se citar o desenvolvimento de uma metodologia capaz de tratar diferentes tipos de arquivos (arquivos anexos) a fim de utilizar o conteúdo textual desses arquivos no processo de classificação textual.

Outra contribuição dessa pesquisa foi a identificação da arquitetura de rede neural profunda mais apropriada para a realização da classificação de aptidão de denúncias escritas em Português. Os estudos apontaram que dentre as redes CNN, LSTM, BERT simples e BERT com LSTM, a arquitetura que utilizava o modelo BERT simples, com algumas camadas adicionais para a realização do *fine tuning*, foi a que apresentou melhores resultados.

Essa pesquisa também propôs dois métodos de sumarização textual, que foram utilizados para sumarizar os textos das denúncias. O primeiro método de sumarização proposto é baseado em frequência de palavras, sendo que, as palavras mais frequentes dos textos recebem uma pontuação maior e as menos frequentes recebem uma pontuação menor. Dessa forma, torna-se possível ranquear a importância das sentenças do texto pela soma das pontuações das palavras que compõem cada uma das sentenças. Por fim, seleciona-se as k sentenças mais importantes para comporem o texto sumarizado, sendo que, esse valor k é uma parâmetro passado como entrada para o algoritmo e indica o número de sentenças que se deseja que o texto sumarizado tenha. Cabe ressaltar que o sumário final é composto pelas sentenças selecionadas, porém, a ordem em que essas sentenças aparecem no resumo respeita a ordem em que elas aparecem no textos original, a fim de se manter a coerência do texto sumarizado.

O outro método de sumarização proposto utiliza uma rede BERT. Nessa proposta, o texto original é dividido em sentenças e cada sentença é submetida ao modelo BERT. Sendo assim, para cada uma das sentenças, o modelo BERT oferece como saída uma representação vetorial que possui uma carga semântica. Ou seja, sentenças com significados semelhantes tendem a ser representadas por vetores pró-

ximos no espaço vetorial em questão. Do mesmo jeito, sentenças com significados muito diferentes ocupam posições distantes nesse mesmo espaço vetorial. Após isso, utiliza-se todos os vetores originados pelas sentenças de entrada a fim de se calcular o vetor médio. Dessa forma, o sumário será formado pelas k sentenças cujos vetores representativos estão mais próximos desse vetor médio. Nesse algoritmo, o valor k também representa um parâmetro de entrada que indica o número de sentenças que devem compor o sumário final. Nesse método, as sentenças que foram selecionadas para compor o sumário, também aparecem na mesma sequência em que elas aparecem no texto original.

Os estudos também apontaram que a utilização dos textos sumarizados para a criação de modelos de classificação de denúncias faz com que os resultados da classificação sejam piores do que a classificação realizada por modelos gerados com os textos originais.

No entanto, caso haja alguma limitação quanto ao processamento das denúncias, a pesquisa também identificou que a sumarização utilizando o método do BERT é mais apropriada do que a que utiliza o método da frequência, para a geração de modelos de classificação de denúncias.

Outra contribuição dessa pesquisa foi a proposta de uma metodologia de extração de variáveis dos textos das denúncias e de seus respectivos anexos e a criação de novas variáveis, utilizando fontes de dados externas, a partir das variáveis identificadas nos textos. Essas variáveis geradas podem ser utilizadas como uma forma estruturada de representação das denúncias.

Por fim, a pesquisa também propôs uma combinação de um modelo de classificação baseado nos textos com um modelo de classificação baseado nos dados estruturados (variáveis extraídas) cujo desempenho superou o desempenho dos modelos individuais.

6.2.3 Publicações

Durante os estudos, também foram geradas algumas publicações acadêmicas.

O artigo *"Convolutional Neural Networks and Long Short-Term Memory Networks for Textual Classification of Information Access Requests"* [11] cria classificadores para textos de solicitações de acesso à informação. Esse artigo foi publicado na revista *IEEE Latin America Transactions*.

O trabalho analisa arquiteturas de redes CNN e LSTM, assim como uma combinação dessas duas arquiteturas, para identificar aquela que é mais apropriada para o problema em questão. O artigo também avalia o comportamento das soluções propostas em ambientes de processamento em paralelo com GPU e em ambiente apenas com CPU.

Já no contexto do tratamento de denúncias, o artigo *"Extraction and enrichment of features to improve complaint text classification performance"* [78] descreve a proposta de extração de variáveis que foi exposta no Capítulo 5 e foi apresentado no XVIII Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2021).

O artigo "Ferramenta para Classificação de Denúncias: Uma abordagem Baseada em Textos e Dados Estruturados" [79] foi apresentado no Workshop de Teses e Dissertações em Sistemas de Informação (WTDSI 2022). Esse Workshop seleciona propostas de teses e dissertações na área de sistemas de informação. Durante a apresentação das propostas, uma banca, formada por professores dessa área, analisa e faz sugestões de melhoria para as pesquisas que estão sendo desenvolvidas. Dessa forma, algumas partes dessa tese são fruto de sugestões recebidas nesse workshop.

O artigo "Sumarização de Denúncias: Proposta e Avaliação de Métodos de Geração de Resumos" [80] foi apresentado no Workshop de Computação Aplicada em Governo Eletrônico (WCGE 2022) e traz os estudos que foram descritos no Capítulo 4.

Já o capítulo "Deep Learning para Processamento de Linguagem Natural" [81] foi publicado no livro "Tópicos Especiais em Sistemas de Informação: Minicursos SBSI 2022" [82]. Esse capítulo se refere a um curso de Deep Learning para Processamento de Linguagem Natural que foi ministrado no XVIII Simpósio Brasileiro de Sistemas de Informação (SBSI 2022). Esse curso foi fruto dos estudos que foram realizados durante o desenvolvimento dessa pesquisa.

6.3 Trabalhos Futuros

A pesquisa em questão abre o horizonte para uma série de trabalhos futuros. As próximas subseções apresentam algumas possibilidades de trabalhos a serem desenvolvidos a partir dessa pesquisa.

6.3.1 Classificador de Manifestações

As denúncias são recebidas por meio de um sistema denominado Fala.BR. No entanto, esse sistema também é utilizado para outros tipos de interações entre o cidadão e o governo.

O sistema Fala.Br pode receber cinco tipos de manifestações: elogios, reclamações, comunicações, pedidos de acesso à informação e denúncias. Porém, a escolha do tipo de manifestação é feita pelo próprio usuário. Muitas vezes, o usuário não sabe em qual categoria ele deve enquadrar a sua manifestação, e por conseguinte, acaba fazendo esse enquadramento de forma errada. Esses erros causam contratempos para as equipes responsáveis pelo tratamento das manifestações, visto que, cada

tipo de manifestação é tratado por uma equipe diferente.

Sendo assim, uma possibilidade de trabalho futuro seria o desenvolvimento de um classificador que identificasse o tipo da manifestação a partir do seu texto.

6.3.2 Indicação se uma manifestação é denúncia

Outro desdobramento para essa pesquisa seria o desenvolvimento de um classificador binário que diz, a partir do texto de uma manifestação, se essa manifestação é uma denúncia ou não.

Esse classificador binário poderia ser utilizado em conjunto com o classificador desenvolvido nessa pesquisa, a fim de evitar que manifestações erradamente classificadas como denúncia pelo usuário, sejam erroneamente tratadas como denúncias e consequentemente automaticamente respondidas de forma equivocada.

6.3.3 Classificador de competência

A Administração Pública é dividida em diferentes níveis: municípios, estados e União. Sendo assim, a União não tem competência para agir em assuntos afetos aos municípios e estados. Logo, caso chegue à Ouvidoria Geral da União uma denúncia que trata de algum assunto relacionado a um município ou estado, a OGU não tem competência para apurar essa denúncia.

Sendo assim, faz-se necessário o desenvolvimento de um classificador de competência que seja capaz de definir se uma determinada denúncia é de competência da União ou não a partir do texto dessa denúncia.

6.3.4 Classificador por área de apuração

Uma vez verificada a aptidão de uma determinada denúncia, ela deve ser encaminhada para a área (ou para as áreas) responsável pela apuração do referido tipo da denúncia. Assim, caso a denúncia trate de assunto relacionado à área de saúde, ela deve ser encaminhada para o setor responsável por auditar questões relacionadas à saúde. Da mesma forma, uma denúncia pode se referir a assunto da área de educação, infraestrutura, dentre outras.

Sendo assim, para a automatização de todo o ciclo de tratamento de uma denúncia, faz-se necessário o desenvolvimento de um classificador de denúncia capaz de identificar o assunto a que essa denúncia se refere, a partir dos textos dessa denúncia. Isso permitiria a automação do envio da denúncia para a área responsável pela apuração.

6.3.5 Agrupamento de denúncias semelhantes

Muitas vezes, recebe-se várias denúncias que tratam de assuntos semelhantes. Dessa forma, o ideal é que essas denúncias recebam o mesmo tipo de encaminhamento. Sendo assim, o desenvolvimento de um modelo capaz de identificar denúncias cujos conteúdos são semelhantes seria de grande valia para o processo de tratamento de denúncias.

6.3.6 Ranqueamento de denúncias por dano potencial

A quantidade de denúncias recebidas é muito grande, e em algumas situações, devido a limitações de capacidade operacional, faz-se necessária a seleção de um conjunto de denúncias que deve ser tratado com maior prioridade.

Dessa forma, o desenvolvimento de um modelo capaz de prever o dano (valor monetário envolvido) potencial dos fatos narrados nos textos das denúncias facilitaria o processo de priorização das denúncias que devem ser apuradas primeiro.

6.3.7 Reconhecimento de novas entidades nomeadas nos textos das denúncias

Uma parte dessa pesquisa consistiu na identificação de alguns elementos nos textos das denúncias, como por exemplo nomes de pessoas, CPFs e CNPJs. No entanto, existem outros tipos de entidades cuja identificação seria relevante para o tratamento das denúncias.

Sendo assim, uma possibilidade de trabalho futuro seria a identificação de possíveis entidades relevantes para o contexto em questão, e o desenvolvimento de metodologias de identificação dessa unidades nos textos das denúncias.

6.3.8 Verificação se um anexo é válido para a apuração da denúncia

Como apresentado na definição do problema, uma denúncia pode vir acompanhada de vários arquivos anexos (algumas denúncias chegam a ter 50 arquivos anexos). No entanto, nem sempre um arquivo anexo é útil para o processo de análise da denúncia, sendo que, nesses casos, o arquivo anexo acaba prejudicando o processo como um todo.

Sendo assim, uma pesquisa que auxiliaria no problema do tratamento de denúncias seria o desenvolvimento de um classificador que fosse capaz de analisar o conteúdo dos arquivos anexos, e baseado nesse conteúdo, determinar se o arquivo é relevante ou não para o contexto da denúncia em questão.

6.3.9 Tratamento de arquivos anexos de áudio e vídeo

Em algumas situações as denúncias vêm acompanhadas de arquivos de áudio e vídeo, que atualmente não são tratados pelo processo de classificação automática da denúncias.

Sendo assim, pesquisas que estudem formas de considerarem os conteúdos de áudio e vídeo de arquivos anexos ajudaria bastante no processo de tratamento das denúncias.

6.3.10 Formas de visualização dos dados estruturados

Como foi citado na Subseção 6.1.2, todos os dados estruturados extraídos e criados durante o processamento das denúncias são repassados para a área de ouvidoria, a fim de auxiliar no tratamento dessas denúncias.

Porém, o volume de dados é muito grande e mutas vezes fica difícil de pessoas sem experiência com manipulação e análise de dados conseguirem consumir esse tipo de informação.

Sendo assim, estudos que busquem novas formas de representação desses dados, facilitando a análise e a descoberta de novos conhecimentos, ajudariam na atividade de tratamento de denúncias.

6.4 Considerações Finais

Essa pesquisa teve o objetivo de desenvolver um modelo de classificação textual capaz de prever se uma denúncia deve ser classificada como apta ou não.

Para atingir esse objetivo, formulou-se três questões de pesquisa. A primeira questão era: "Qual é a arquitetura de rede mais apropriada para o problema de classificação de aptidão de denúncias?".

Os estudos demonstraram que a rede que utilizava o modelo pré-treinado BERT era a que apresentava melhores resultados.

Já a segunda questão de pesquisa a ser respondida era: "Como extrair as principais informações dos textos das denúncias a fim de utilizá-las no processo de classificação textual?". Para responder essa questão, foram desenvolvidas duas estratégias de sumarização de denúncias, uma baseada em frequência de palavras e outra que utilizava o modelo BERT. No entanto, os estudos demonstraram que os modelos gerados com textos sumarizados tinham desempenhos piores do que os modelos gerados com os textos originais.

Por fim, a última questão de pesquisa dizia: "como considerar informações externas aos textos das denúncias no processo de classificação de aptidão de denúncias?".

Essa questão foi respondida com a proposta de uma metodologia que extrai algumas informações dos textos das denúncias, e com base nessas informações, consulta bases de dados externas, a fim de trazer novas informações para o processo de classificação de denúncias.

No entanto, a principal motivação dessa pesquisa foi fornecer meios para que o Estado pudesse responder de forma mais célere e eficaz as denúncias formuladas pelos cidadãos. Essa melhora no tempo de resposta das denúncias é importante porque além de possibilitar uma resposta mais tempestiva a possíveis irregularidades que estejam acontecendo na Administração Pública, também estimula que outros cidadãos façam novas denúncias.

A formulação de denúncias por parte dos cidadãos reflete a preocupação que eles têm com o funcionamento da Administração Pública e os envolve nas atividades do governo. Esse envolvimento crescente dos cidadãos nos atos governamentais faz com que a democracia se fortaleça cada vez mais.

Referências Bibliográficas

[1] RONG, X. "word2vec parameter learning explained", *arXiv preprint arXiv:1411.2738*, 2014.

[2] ZHU, W., LAN, C., XING, J., et al. "Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks", *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 3697–3703, mar 2016. Disponível em: <https://arxiv.org/abs/1603.07772v1http://arxiv.org/abs/1603.07772>.

[3] VASWANI, A., SHAZEER, N., PARMAR, N., et al. "Attention Is All You Need", *Advances in Neural Information Processing Systems*, v. 2017-Decem, n. Nips, pp. 5999–6009, jun 2017. ISSN: 10495258. Disponível em: <http://arxiv.org/abs/1706.03762>.

[4] DEVLIN, J., CHANG, M. W., LEE, K., et al. "BERT: Pre-training of deep bidirectional transformers for language understanding", *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, v. 1, n. Mlm, pp. 4171–4186, 2019.

[5] BELTAGY, I., PETERS, M. E., COHAN, A. "Longformer: The Long-Document Transformer", *ArXiv*, v. abs/2004.05150, 2020.

[6] ZAHEER, M., GURUGANESH, G., DUBEY, K. A., et al. "Big Bird: Transformers for Longer Sequences". In: *Advances in Neural Information Processing Systems*, v. 33, pp. 17283–17297, 2020.

[7] PENNINGTON, J., SOCHER, R., MANNING, C. "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics. ISBN: 9781937284961. doi: 10.3115/v1/D14-1162. Disponível em: <https://aclanthology.org/D14-1162http://aclweb.org/anthology/D14-1162>.

[8] DA UNIÃO, C. G. "Manual de Ouvidoria Pública rumo ao sistema participativo", *OGU. Disponível em:< http://www. ouvidorias. gov. br/central-de-conteudos/biblioteca/arquivos/manual-deouvidoria-publica. pdf> Acesso em*, v. 15, 2016.

[9] ALEXANDRINO, MARCELO E PAULO, V. *Direito administrativo*. Niterói-RJ, Impetus, 2006.

[10] CGU. "Controladoria-Geral da União - Ouvidoria". 2020. Disponível em: <https://www.gov.br/cgu/pt-br/assuntos/ouvidoria/ouvidoria>.

[11] PAIVA, E., PAIM, A., EBECKEN, N. "Convolutional Neural Networks and Long Short-Term Memory Networks for Textual Classification of Information Access Requests", *IEEE Latin America Transactions*, v. 19, n. 5, pp. 826–833, Jun. 2021. Disponível em: <https://latamt.ieeer9.org/index.php/transactions/article/view/4094>.

[12] YADAV, A., VISHWAKARMA, D. K. "Sentiment analysis using deep learning architectures: a review", *Artificial Intelligence Review*, v. 53, n. 6, pp. 4335–4385, 2020.

[13] HASSAN, H., AUE, A., CHEN, C., et al. "Achieving human parity on automatic chinese to english news translation", *arXiv preprint arXiv:1803.05567*, 2018.

[14] KOSMAJAC, D., KEŠELJ, V. "Automatic Text Summarization of News Articles in Serbian Language". In: *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pp. 1–6. IEEE, 2019.

[15] MIKOLOV, T., SUTSKEVER, I., CHEN, K., et al. "Distributed Representations of Words and Phrases and their Compositionality", *Advances in Neural Information Processing Systems*, oct 2013. ISSN: 10495258. Disponível em: <https://arxiv.org/abs/1310.4546v1http://arxiv.org/abs/1310.4546>.

[16] MIKOLOV, T., CHEN, K., CORRADO, G., et al. "Efficient Estimation of Word Representations in Vector Space", *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, jan 2013. Disponível em: <https://arxiv.org/abs/1301.3781v3http://arxiv.org/abs/1301.3781>.

[17] LECUN, Y., BOTTOU, L., BENGIO, Y., et al. "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, v. 86, n. 11, pp. 2278–2324, 1998. doi: 10.1109/5.726791.

[18] ELMAN, J. L. "Finding structure in time", *Cognitive Science*, v. 14, n. 2, pp. 179–211, 1990. ISSN: 0364-0213. doi: https://doi.org/10.1016/0364-0213(90)90002-E. Disponível em: <https://www.sciencedirect.com/science/article/pii/036402139090002E>.

[19] CHOLLET, F. *Deep learning with Python*. 1 ed. Shelter Island, NY 11964, Simon and Schuster, 2017.

[20] GOYAL, P., PANDEY, S., JAIN, K. *Deep Learning for Natural Language Processing*. Berkeley, CA, Apress, 2018. ISBN: 978-1-4842-3684-0. doi: 10.1007/978-1-4842-3685-7. Disponível em: <https://link.springer.com/content/pdf/10.1007/978-1-4842-3685-7.pdfhttp://link.springer.com/10.1007/978-1-4842-3685-7>.

[21] BENGIO, Y., DUCHARME, R., VINCENT, P. "A neural probabilistic language model", *Advances in Neural Information Processing Systems*, v. 13, 2000.

[22] HARRIS, Z. S. "Distributional Structure", *WORD*, v. 10, n. 2-3, pp. 146–162, aug 1954. ISSN: 0043-7956. doi: 10.1080/00437956.1954.11659520. Disponível em: <https://www.tandfonline.com/action/journalInformation?journalCode=rwrd20http://www.tandfonline.com/doi/full/10.1080/00437956.1954.11659520>.

[23] FIRTH, J. R. "A synopsis of linguistic theory, 1930-1955", *Studies in linguistic analysis*, 1957.

[24] KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E. "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, v. 60, n. 6, pp. 84–90, may 2017. ISSN: 0001-0782. doi: 10.1145/3065386. Disponível em: <http://code.google.com/p/cuda-convnet/https://dl.acm.org/doi/10.1145/3065386>.

[25] HAN, D., LIU, Q., FAN, W. "A new image classification method using CNN transfer learning and web data augmentation", *Expert Systems with Applications*, v. 95, pp. 43–56, apr 2018. ISSN: 09574174. doi: 10.1016/j.eswa.2017.11.028. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0957417417307844>.

[26] DÍAZ-GUERRA, L., DAHER ADEGAS, F., STOUSTRUP, J., et al. "Adaptive control algorithm for improving power capture of wind turbines in turbulent winds". In: *2012 American Control Conference (ACC)*, pp. 5807–5812, 2012. doi: 10.1109/ACC.2012.6315452.

[27] KIM, Y. "Convolutional Neural Networks for Sentence Classification". In: Moschitti, A., Pang, B., Daelemans, W. (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1746–1751. ACL, 2014. doi: 10.3115/v1/d14-1181. Disponível em: <https://doi.org/10.3115/v1/d14-1181>.

[28] ZHANG, X., ZHAO, J. J., LECUN, Y. "Character-level Convolutional Networks for Text Classification". In: Cortes, C., Lawrence, N. D., Lee, D. D., et al. (Eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 649–657, 2015. Disponível em: <https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html>.

[29] JOHNSON, R., ZHANG, T. "Effective Use of Word Order for Text Categorization with Convolutional Neural Networks". In: Mihalcea, R., Chai, J. Y., Sarkar, A. (Eds.), *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pp. 103–112. The Association for Computational Linguistics, 2015. doi: 10.3115/v1/n15-1011. Disponível em: <https://doi.org/10.3115/v1/n15-1011>.

[30] PASCANU, R., MIKOLOV, T., BENGIO, Y. "On the difficulty of training recurrent neural networks". In: *30th International Conference on Machine Learning, ICML 2013*, n. PART 3, pp. 2347–2355. PMLR, may 2013. Disponível em: <http://proceedings.mlr.press/v28/pascanu13.html>.

[31] HOCHREITER, S., SCHMIDHUBER, J. "Long Short-Term Memory", *Neural Computation*, v. 9, n. 8, pp. 1735–1780, nov 1997. ISSN: 0899-7667. doi: 10.1162/neco.1997.9.8.1735. Disponível em: <http://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdfhttps://direct.mit.edu/neco/article/9/8/1735-1780/6109>.

[32] CHO, K., VAN MERRIENBOER, B., GULCEHRE, C., et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Stroudsburg, PA, USA, jun 2014. Association for Computational Linguistics. ISBN: 9781937284961. doi: 10.3115/v1/D14-1179. Dispo-

nível em: <https://arxiv.org/abs/1406.1078v3http://aclweb.org/anthology/D14-1179>.

[33] LIU, Y., OTT, M., GOYAL, N., et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach", *ArXiv*, v. abs/1907.11692, 2019.

[34] LAN, Z., CHEN, M., GOODMAN, S., et al. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations", *ArXiv*, v. abs/1909.11942, 2020.

[35] SANH, V., DEBUT, L., CHAUMOND, J., et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter", *ArXiv*, v. abs/1910.01108, 2019.

[36] PETERS, M. E., NEUMANN, M., IYYER, M., et al. "Deep contextualized word representations". In: *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, v. 1, pp. 2227–2237, Stroudsburg, PA, USA, 2018. Association for Computational Linguistics. ISBN: 9781948087278. doi: 10.18653/v1/n18-1202. Disponível em: <http://allennlp.org/elmohttp://aclweb.org/anthology/N18-1202>.

[37] RADFORT, A., NARASIMHAN, K., SALIMANS, T., et al. "(OpenAI Transformer): Improving Language Understanding by Generative Pre-Training", *OpenAI*, pp. 1–10, 2018. Disponível em: <https://gluebenchmark.com/leaderboard>.

[38] VINCENT, P., LAROCHELLE, H., BENGIO, Y., et al. "Extracting and composing robust features with denoising autoencoders". In: *Proceedings of the 25th international conference on Machine learning - ICML '08*, pp. 1096–1103, New York, New York, USA, 2008. ACM Press. ISBN: 9781605582054. doi: 10.1145/1390156.1390294. Disponível em: <http://portal.acm.org/citation.cfm?doid=1390156.1390294>.

[39] SUN, C., QIU, X., XU, Y., et al. "How to Fine-Tune BERT for Text Classification?" *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 11856 LNAI, n. 2, pp. 194–206, may 2019. ISSN: 16113349. doi: 10.1007/978-3-030-32381-3_16. Disponível em: <http://link.springer.com/10.1007/978-3-030-32381-3_16http://arxiv.org/abs/1905.05583>.

[40] MCCLOSKEY, M., COHEN, N. J. "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem", *Psychology of Learning and Motivation - Advances in Research and Theory*, v. 24, n. C, pp. 109–165, jan 1989. ISSN: 0079-7421. doi: 10.1016/S0079-7421(08)60536-8.

[41] KALCHBRENNER, N., GREFENSTETTE, E., BLUNSOM, P. "A Convolutional Neural Network for Modelling Sentences". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pp. 655–665. The Association for Computer Linguistics, 2014. doi: 10.3115/v1/p14-1062. Disponível em: <https://doi.org/10.3115/v1/p14-1062>.

[42] WANG, P., XU, J., XU, B., et al. "Semantic Clustering and Convolutional Neural Network for Short Text Categorization". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pp. 352–357. The Association for Computer Linguistics, 2015. doi: 10.3115/v1/p15-2058. Disponível em: <https://doi.org/10.3115/v1/p15-2058>.

[43] PORIA, S., CAMBRIA, E., HAZARIKA, D., et al. "A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks". In: Calzolari, N., Matsumoto, Y., Prasad, R. (Eds.), *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pp. 1601–1612. ACL, 2016. Disponível em: <https://aclanthology.org/C16-1151/>.

[44] ZHOU, C., SUN, C., LIU, Z., et al. "A C-LSTM Neural Network for Text Classification", *CoRR*, v. abs/1511.08630, 2015. Disponível em: <http://arxiv.org/abs/1511.08630>.

[45] VU, N. T., ADEL, H., GUPTA, P., et al. "Combining Recurrent and Convolutional Neural Networks for Relation Classification". In: Knight, K., Nenkova, A., Rambow, O. (Eds.), *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 534–539. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/n16-1065. Disponível em: <https://doi.org/10.18653/v1/n16-1065>.

[46] YIN, W., KANN, K., YU, M., et al. "Comparative Study of CNN and RNN for Natural Language Processing", *CoRR*, v. abs/1702.01923, 2017. Disponível em: <http://arxiv.org/abs/1702.01923>.

[47] ZHENG, S., YANG, M. "A New Method of Improving BERT for Text Classification". In: Cui, Z., Pan, J., Zhang, S., et al. (Eds.), *Intelligence Science and Big Data Engineering. Big Data and Machine Learning - 9th International Conference, IScIDE 2019, Nanjing, China, October 17-20, 2019, Proceedings, Part II*, v. 11936, *Lecture Notes in Computer Science*, pp. 442–452. Springer, 2019. doi: 10.1007/978-3-030-36204-1_37. Disponível em: <https://doi.org/10.1007/978-3-030-36204-1_37>.

[48] YU, S., SU, J., LUO, D. "Improving BERT-Based Text Classification With Auxiliary Sentence and Domain Knowledge", *IEEE Access*, v. 7, pp. 176600–176612, 2019. doi: 10.1109/ACCESS.2019.2953990. Disponível em: <https://doi.org/10.1109/ACCESS.2019.2953990>.

[49] BANERJEE, I., LING, Y., CHEN, M. C., et al. "Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification", *Artif. Intell. Medicine*, v. 97, pp. 79–88, 2019. doi: 10.1016/j.artmed.2018.11.004. Disponível em: <https://doi.org/10.1016/j.artmed.2018.11.004>.

[50] QUAN, W., CHEN, Z., GAO, J., et al. "Comparative Study of CNN and LSTM based Attention Neural Networks for Aspect-Level Opinion Mining". In: Abe, N., Liu, H., Pu, C., et al. (Eds.), *IEEE International Conference on Big Data (IEEE BigData 2018), Seattle, WA, USA, December 10-13, 2018*, pp. 2141–2150. IEEE, 2018. doi: 10.1109/BigData.2018.8622150. Disponível em: <https://doi.org/10.1109/BigData.2018.8622150>.

[51] UNDAVIA, S., MEYERS, A., ORTEGA, J. "A Comparative Study of Classifying Legal Documents with Neural Networks". In: Ganzha, M., Maciaszek, L. A., Paprzycki, M. (Eds.), *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, FedCSIS 2018, Poznań, Poland, September 9-12, 2018*, v. 15, *Annals of Computer Science and Information Systems*, pp. 515–522, 2018. doi: 10.15439/2018F227. Disponível em: <https://doi.org/10.15439/2018F227>.

[52] ZHAO, L., LI, L., ZHENG, X., et al. "A BERT based Sentiment Analysis and Key Entity Detection Approach for Online Financial Texts". In: Shen, W., Barthès, J. A., Luo, J., et al. (Eds.), *24th IEEE International Conference on Computer Supported Cooperative Work in De-*

sign, CSCWD 2021, Dalian, China, May 5-7, 2021, pp. 1233–1238. IEEE, 2021. doi: 10.1109/CSCWD49262.2021.9437616. Disponível em: <https://doi.org/10.1109/CSCWD49262.2021.9437616>.

[53] SRIVASTAVA, N., HINTON, G. E., KRIZHEVSKY, A., et al. "Dropout: a simple way to prevent neural networks from overfitting", *J. Mach. Learn. Res.*, v. 15, n. 1, pp. 1929–1958, 2014. doi: 10.5555/2627435.2670313. Disponível em: <https://dl.acm.org/doi/10.5555/2627435.2670313>.

[54] HARTMANN, N. S., FONSECA, E. R., SHULBY, C. D., et al. "Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks". In: *Anais do XI Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pp. 122–131, Porto Alegre, RS, Brasil, 2017. SBC. Disponível em: <https://sol.sbc.org.br/index.php/stil/article/view/4008>.

[55] ABDEL-SALAM, S., RAFEA, A. "Performance Study on Extractive Text Summarization Using BERT Models", *Information*, v. 13, n. 2, pp. 67, 2022.

[56] GHODRATNAMA, S., BEHESHTI, A., ZAKERSHAHRAK, M., et al. "Extractive document summarization based on dynamic feature space mapping", *IEEE Access*, v. 8, pp. 139084–139095, 2020.

[57] LUHN, H. P. "The automatic creation of literature abstracts", *IBM Journal of research and development*, v. 2, n. 2, pp. 159–165, 1958.

[58] EDMUNDSON, H. P. "New methods in automatic extracting", *Journal of the ACM (JACM)*, v. 16, n. 2, pp. 264–285, 1969.

[59] NENKOVA, A., VANDERWENDE, L. "The impact of frequency on summarization", *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, v. 101, 2005.

[60] STEINBERGER, J., JEZEK, K., OTHERS. "Using latent semantic analysis in text summarization and summary evaluation", *Proc. ISIM*, v. 4, n. 93-100, pp. 8, 2004.

[61] MIHALCEA, R., TARAU, P. "Textrank: Bringing order into text". In: *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411, 2004.

[62] ERKAN, G., RADEV, D. R. "Lexrank: Graph-based lexical centrality as salience in text summarization", *Journal of artificial intelligence research*, v. 22, pp. 457–479, 2004.

[63] DONG, Y., MIRCEA, A., CHEUNG, J. C. "Discourse-aware unsupervised summarization of long scientific documents", *arXiv preprint arXiv:2005.00513*, 2020.

[64] MILLER, D. "Leveraging BERT for extractive text summarization on lectures", *arXiv preprint arXiv:1906.04165*, 2019.

[65] GU, X., WANG, Z., BI, Z., et al. "UCPhrase: Unsupervised Context-aware Quality Phrase Tagging", *arXiv preprint arXiv:2105.14078*, 2021.

[66] FERREIRA, J. C., PATINO, C. M. "What does the p value really mean?" *Jornal Brasileiro de Pneumologia*, v. 41, n. 5, pp. 485, 2015.

[67] FELDMAN, R., SANGER, J., OTHERS. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press, 2007.

[68] WU, L., MORSTATTER, F., LIU, H. "SlangSD: building, expanding and using a sentiment dictionary of slang words for short-text sentiment classification", *Lang. Resour. Evaluation*, v. 52, n. 3, pp. 839–852, 2018. doi: 10.1007/s10579-018-9416-0. Disponível em: <`https://doi.org/10.1007/s10579-018-9416-0`>.

[69] KARTHIKEYAN, T., SEKARAN, K., D., R., et al. "Personalized Content Extraction and Text Classification Using Effective Web Scraping Techniques", *Int. J. Web Portals*, v. 11, n. 2, pp. 41–52, 2019. doi: 10.4018/IJWP.2019070103. Disponível em: <`https://doi.org/10.4018/IJWP.2019070103`>.

[70] LI, Z. "A Classification Retrieval Approach for English Legal Texts". In: *2019 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)*, pp. 220–223, 2019. doi: 10.1109/ICITBS.2019.00059.

[71] LIU, C.-Z., SHENG, Y.-X., WEI, Z.-Q., et al. "Research of Text Classification Based on Improved TF-IDF Algorithm". In: *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, pp. 218–222, 2018. doi: 10.1109/IRCE.2018.8492945.

[72] COUSSEMENT, K., VAN DEN POEL, D. "Improving customer complaint management by automatic email classification using linguistic style features as predictors", *Decision Support Systems*, v. 44, n. 4, pp. 870–882, 2008. ISSN: 0167-9236. doi: https://doi.org/10.1016/j.dss.2007.10.010. Disponível em: <`https://www.sciencedirect.com/science/article/pii/S0167923607001820`>.

[73] SOUZA, F., NOGUEIRA, R., LOTUFO, R. "Portuguese Named Entity Recognition using BERT-CRF", *arXiv preprint arXiv:1909.10649*, 2019. Disponível em: <http://arxiv.org/abs/1909.10649>.

[74] SOUZA, F., NOGUEIRA, R., LOTUFO, R. "BERTimbau: pretrained BERT models for Brazilian Portuguese". In: *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*, 2020.

[75] BREIMAN, L. "Random forests", *Machine learning*, v. 45, n. 1, pp. 5–32, 2001.

[76] KE, G., MENG, Q., FINLEY, T., et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: Guyon, I., Luxburg, U. V., Bengio, S., et al. (Eds.), *Advances in Neural Information Processing Systems*, v. 30. Curran Associates, Inc., 2017. Disponível em: <https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf>.

[77] CHEN, T., GUESTRIN, C. "XGBoost". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, aug 2016. doi: 10.1145/2939672.2939785. Disponível em: <https://doi.org/10.1145%2F2939672.2939785>.

[78] PAIVA, E., PEREIRA, F. "Extraction and enrichment of features to improve complaint text classification performance". In: *Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional*, pp. 338–349, Porto Alegre, RS, Brasil, 2021. SBC. doi: 10.5753/eniac.2021.18265. Disponível em: <https://sol.sbc.org.br/index.php/eniac/article/view/18265>.

[79] DE PAIVA, E., EBECKEN, N. "Ferramenta para Classificaçao de Denuncias: Uma abordagem Baseada em Textos e Dados Estruturados". In: *Anais Estendidos do XVIII Simpósio Brasileiro de Sistemas de Informação*, pp. 83–90. SBC, 2022.

[80] PAIVA, E., PEREIRA, F., EBECKEN, N. "Sumarização de Denúncias: Proposta e Avaliação de Métodos de Geração de Resumos". In: *Anais do X Workshop de Computação Aplicada em Governo Eletrônico*, pp. 121–132, Porto Alegre, RS, Brasil, 2022. SBC. doi: 10.5753/wcge.2022.223020. Disponível em: <https://sol.sbc.org.br/index.php/wcge/article/view/20716>.

[81] DE PAIVA, E. S., PEREIRA, F. S. "Deep Learning para Processamento de Linguagem Natural", *Sociedade Brasileira de Computação*, 2022.

[82] ARAÚJO, R. D., SETTI, M., BERARDI, R. C. G., et al. "Tópicos Especiais em Sistemas de Informação: Minicursos SBSI 2022", *Sociedade Brasileira de Computação*, 2022.