



**Universidade de Brasília
Departamento de Estatística**

**Processamento de Linguagem Natural para Aplicação de Técnicas de
Aprendizado de Máquina e Reconhecimento de Entidades Nomeadas em
Portarias Jurídicas**

Davi Esmeraldo da Silva Albuquerque

Trabalho de Conclusão de Curso apresentado para o Departamento de Estatística da Universidade de Brasília como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

**Brasília
2025**

Davi Esmeraldo da Silva Albuquerque

**Processamento de Linguagem Natural para Aplicação de Técnicas de
Aprendizado de Máquina e Reconhecimento de Entidades Nomeadas em
Portarias Jurídicas**

Orientador: Prof. Eduardo Monteiro de Castro Gomes

Trabalho de Conclusão de Curso apresentado para o Departamento de Estatística da Universidade de Brasília como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

**Brasília
2025**

Agradecimentos

Aos meus pais, Marconi e Jocyane, dedico meu mais profundo e sincero agradecimento. Eles sempre foram minha base e fonte de inspiração. Nunca mediram esforços para garantir que eu e meu irmão tivéssemos tudo o que precisávamos. O que somos e o que ainda nos tornaremos é também fruto da dedicação imensurável que sempre tiveram à nossa formação e educação. Espero ainda ter muitas oportunidades de retribuir e de deixá-los orgulhosos.

Sou imensamente grato à oportunidade de ter cursado Estatística na Universidade de Brasília (UnB), uma instituição de excelência, com docentes altamente qualificados que contribuíram de maneira significativa para minha formação acadêmica. Desse modo, gostaria de expressar minha sincera gratidão ao professor Eduardo Monteiro pela orientação deste trabalho. Sua atuação superou minhas expectativas, especialmente nos momentos em que demonstrou interesse genuíno pelos meus objetivos pessoais e se colocou à disposição para me escutar e auxiliar com entusiasmo.

Sou profundamente grato pelas experiências acadêmicas e profissionais que vivi ao longo da graduação, com destaque especial à oportunidade de integrar e liderar a Empresa Júnior de Estatística da UnB, a ESTAT. Essa vivência contribuiu de maneira determinante para o meu desenvolvimento pessoal e profissional. Como disse em meu discurso de despedida da empresa, sou grato por cada desafio enfrentado, pelos aprendizados que marcaram minha trajetória, pelas conquistas celebradas e, principalmente, pelas pessoas incríveis que tive o privilégio de conhecer. Levo comigo não apenas memórias, mas as amizades que construí e que espero preservar por muitos anos. Além disso, minha gratidão se estende também às amizades que carrego desde o ensino fundamental e médio, que continuam sendo fonte de apoio, leveza e pertencimento.

Resumo

No presente trabalho é proposta a aplicação de técnicas de Processamento de Linguagem Natural (PLN) para a análise de portarias publicadas em 2024 pelo Gabinete da Presidência do Tribunal de Justiça do Distrito Federal e dos Territórios (TJDFT). O trabalho contempla três frentes principais: a mensuração da similaridade semântica entre esses documentos, a identificação de agrupamentos temáticos e o reconhecimento de entidades nomeadas. Para tanto, os dados textuais foram extraídos por meio de técnicas de raspagem de dados (*web scraping*), seguidos por procedimentos de pré-processamento, vetorização semântica (*embeddings*), redução de dimensionalidade e, por fim, submetidos a modelos de agrupamento e reconhecimento de entidades nomeadas. Adicionalmente, desenvolveu-se uma aplicação interativa em ambiente web utilizando a biblioteca *Streamlit* do *python*, com o objetivo de disseminar o acesso aos resultados deste trabalho.

Palavras-chave: Processamento de Linguagem Natural, Aprendizado de Máquina, Clusterização, Similaridade Textual, Reconhecimento de Entidades Nomeadas, Portarias Jurídicas.

Abstract

This study proposes the application of Natural Language Processing (NLP) techniques to analyze executive orders published in 2024 by the Office of the Presidency of the Federal District and Territories Tribunal of Justice (TJDFT). The work focuses on three main fronts: measuring the semantic similarity between these documents, identifying thematic clusters, and performing named entity recognition (NER). To this end, textual data were extracted using web scraping techniques, followed by preprocessing procedures, semantic vectorization (embeddings), dimensionality reduction, and finally, submission to clustering and NER models. Additionally, an interactive web application was developed using Python's Streamlit library, with the goal of disseminating access to the results of this study.

Keywords: Natural Language Processing, Machine Learning, Clustering, Textual Similarity, Named Entity Recognition, Legal Orders .

Lista de Tabelas

| | | |
|----|--|----|
| 1 | Matriz de confusão | 18 |
| 2 | Estatísticas descritivas da quantidade de tokens por portaria | 35 |
| 3 | Modelos avaliados, hiperparâmetros ótimos e métricas de coesão | 39 |
| 4 | Top 10 portarias mais semelhantes à portaria de referência segundo Word2Vec e FastText | 40 |
| 5 | Distribuição de portarias por cluster | 46 |
| 6 | Quantidade de entidades anotadas | 50 |
| 7 | Distribuição das entidades anotadas por cluster | 50 |
| 8 | Desempenho médio do modelo - Validação cruzada estratificada | 51 |
| 9 | F1-score médio por entidades - Validação cruzada estratificada | 51 |
| 10 | Desempenho médio do modelo - Validação cruzada | 52 |
| 11 | F1-score médio por entidades - Validação cruzada | 52 |
| 12 | Quantidade de entidades anotadas após sobreamostragem | 53 |
| 13 | Desempenho médio do modelo - Após sobreamostragem - Validação cruzada | 53 |
| 14 | F1-score médio por entidades | 53 |

Lista de Figuras

| | | |
|----|--|----|
| 1 | Fluxograma de Processamento de Linguagem Natural | 14 |
| 2 | Síntese metodológica | 26 |
| 3 | Portaria exemplo | 27 |
| 4 | Descrição da portaria exemplo | 28 |
| 5 | Exemplo anotação | 32 |
| 6 | Distribuição mensal das portarias | 34 |
| 7 | Distribuição semanal das portarias | 35 |
| 8 | Nuvem de palavras dos conteúdos das portarias | 36 |
| 9 | Gráfico de barras dos 20 primeiros verbos mais frequentes | 37 |
| 10 | Gráfico de barras dos 20 primeiros bigramas mais frequentes | 38 |
| 11 | Gráfico de barras dos 20 primeiros trigramas mais frequentes | 38 |
| 12 | PCA - Word2Vec | 41 |
| 13 | PCA - FastText | 42 |
| 14 | Determinação do número ótimo de clusters - Word2Vec | 43 |
| 15 | Determinação do número ótimo de clusters - FastText | 43 |
| 16 | Clusterização das portarias - Word2Vec | 44 |
| 17 | Clusterização das portarias - FastText | 45 |
| 18 | Gráfico de barras dos primeiros verbos mais frequentes - Cluster 0 | 47 |
| 19 | Gráfico de barras dos primeiros verbos mais frequentes - Cluster 1 | 47 |
| 20 | Gráfico de barras dos primeiros verbos mais frequentes - Cluster 2 | 48 |
| 21 | Quantidade de portarias publicadas por mês e por cluster | 49 |
| 22 | Interface do aplicativo - Conteúdos | 55 |
| 23 | Interface do aplicativo - Entidades | 56 |
| 24 | Interface do aplicativo - Similaridade | 56 |
| 25 | Interface do aplicativo - Agrupamentos | 57 |

Lista de Abreviações e Siglas

| | |
|---------------|---|
| IA | Inteligência Artificial |
| PLN | Processamento de Linguagem Natural |
| REN | Reconhecimento de Entidades Nomeadas |
| TJDFT | Tribunal de Justiça do Distrito Federal e dos Territórios |
| WCSS | Within-Cluster Sum of Squares |
| LSTM | Long Short-Term Memory |
| BiLSTM | Bidirectional Long Short-Term Memory |
| CBOW | Continuous Bag of Words |
| PCA | Principal Component Analysis |
| VP | Verdadeiro Positivo |
| VN | Verdadeiro Negativo |
| FP | Falso Positivo |
| FN | Falso Negativo |

Sumário

| | |
|--|----|
| 1 Introdução | 11 |
| 2 Referencial Teórico | 13 |
| 2.1 Processamento de Linguagem Natural (PLN) | 13 |
| 2.1.1 Pré Processamento | 13 |
| 2.1.2 <i>Word Embeddings</i> | 15 |
| 2.2 Aprendizado de Máquina | 16 |
| 2.3 Aprendizado de Máquina Supervisionado | 16 |
| 2.3.1 Fluxo de Processamento | 17 |
| 2.3.2 Otimização da Escolha de Hiperparâmetros | 19 |
| 2.3.3 Amostragem Estratificada | 20 |
| 2.3.4 Validação Cruzada | 20 |
| 2.3.5 Tratamento de Bases Desbalanceadas | 21 |
| 2.4 Aprendizado de Máquina Não Supervisionado | 22 |
| 2.4.1 <i>K-Means</i> | 22 |
| 2.4.2 Similaridade Cosseno | 23 |
| 2.4.3 Análise de Componentes Principais | 23 |
| 2.5 Reconhecimento de Entidades Nomeadas (REN) | 25 |
| 3 Metodologia | 26 |
| 3.1 Coleta de Dados | 26 |
| 3.2 Pré Processamento e Limpeza dos Dados | 27 |
| 3.3 Representação e Modelagem dos Dados | 28 |
| 3.4 Reconhecimento de Entidades Nomeadas | 31 |
| 3.5 Aplicativo Web | 33 |
| 4 Resultados | 34 |
| 4.1 Análise Exploratória dos Dados | 34 |
| 4.2 Definição do Modelo de Embedding | 39 |
| 4.3 Visualização em Dimensionalidade Reduzida dos Embeddings Gerados | 40 |
| 4.4 Agrupamento das Portarias | 42 |

| | | |
|----------|---|-----------|
| 4.5 | Análise Exploratória dos Dados Agrupados | 46 |
| 4.6 | Reconhecimento de Entidades Nomeadas | 49 |
| 4.6.1 | Características do Conjunto de Portarias Anotadas | 49 |
| 4.6.2 | Avaliação do Modelo com Validação Cruzada Estratificada | 51 |
| 4.6.3 | Avaliação do Modelo com Validação Cruzada Simples | 51 |
| 4.6.4 | Avaliação do Modelo com Validação Cruzada Simples após sobrea- mostragem | 52 |
| 4.7 | Aplicativo Web | 54 |
| 5 | Conclusão | 58 |
| 6 | Referências | 60 |

1 Introdução

O avanço das técnicas de Processamento de Linguagem Natural (PLN) tem transformado significativamente a forma como grandes volumes de dados textuais são analisados em diferentes domínios, como os da saúde, das humanidades digitais, do jurídico, entre outros. Esse progresso se deve à capacidade dessas técnicas de permitir que máquinas compreendam, interpretem e gerem linguagem humana (CASELI; NUNES, 2024).

No contexto jurídico, essas ferramentas têm o potencial, por exemplo, de auxiliar na análise de documentos por meio da automatização da extração de nomes de partes envolvidas, datas, dispositivos legais citados e decisões judiciais. No entanto, a análise e a extração de informações desses textos ainda apresentam desafios consideráveis, em razão das especificidades da linguagem jurídica e da heterogeneidade semântica dos documentos desse domínio (GOCHHAIT, 2024).

Ademais, modelos de aprendizado de máquina e de aprendizado profundo destacam-se como abordagens promissoras, pois têm demonstrado bom desempenho, quando combinados com técnicas de PLN, em tarefas relacionadas à utilização de linguagem natural. Tais métodos possuem o potencial de transformar dados não estruturados em conhecimento estruturado, o que facilita o acesso a informações relevantes e contribui para a maior agilidade em processos administrativos e jurídicos (OLIVEIRA; NASCIMENTO, 2021).

Diante deste cenário, considerando a expressiva quantidade de documentos oficiais publicados por instituições do sistema de justiça, identifica-se uma oportunidade de aplicação das técnicas descritas. Desse modo, o objetivo deste trabalho é aplicar modelos baseados em aprendizado de máquina e aprendizado profundo para realizar as tarefas de mensuração de similaridade semântica, identificação de agrupamentos e Reconhecimento de Entidades Nomeadas (REN) de portarias emitidas em 2024 pelo Gabinete da Presidência do Tribunal de Justiça do Distrito Federal e Territórios (TJDFT). Outrossim, propõe-se o desenvolvimento de um aplicativo web interativo, com o objetivo de proporcionar uma visualização clara e sistemática dos principais resultados obtidos.

Assim, destaca-se que a aplicação de técnicas como o cálculo de similaridade semântica e de agrupamento de textos possibilita a criação de representações mais interpretáveis dos dados, tanto para humanos quanto para algoritmos de aprendizado. Essas representações facilitam a segmentação de documentos com base em características temáticas ou estruturais comuns, contribuindo para a organização da informação e a identificação de padrões (MÜLLER E GUIDO, 2016, apud FREITAS, 2023).

De maneira análoga, modelos de Reconhecimento de Entidades Nomeadas (REN) se destacam por sua capacidade de identificar e categorizar informações específicas. No

setor jurídico, essa técnica tem ampla aplicação prática na extração automatizada de, por exemplo, cláusulas contratuais e jurisprudências, promovendo análises mais rápidas e precisas (GOCHHAIT, 2024).

Diante do exposto, por meio da aplicação das técnicas propostas neste trabalho, espera-se contribuir para a modernização da análise de portarias no setor jurídico, oferecendo ganhos de eficiência, suporte à tomada de decisões administrativas e judiciais e transparência dos processos internos presentes nas portarias contempladas.

2 Referencial Teórico

Nesta seção, são apresentados os métodos estatísticos e computacionais adotados neste estudo. A escolha dessas abordagens foi orientada por sua eficácia em tarefas alinhadas aos objetivos do presente trabalho, como o reconhecimento de entidades nomeadas, o cálculo de similaridade textual e o agrupamento das portarias em análise.

2.1 Processamento de Linguagem Natural (PLN)

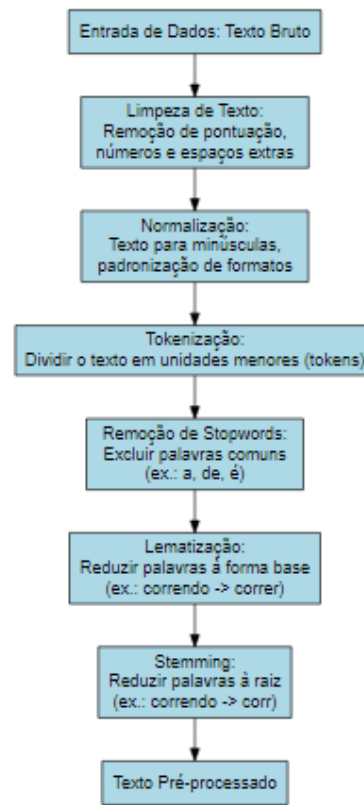
O Processamento de Linguagem Natural (PLN), ou *Natural Language Processing (NLP)*, é um campo da Inteligência Artificial que investiga métodos para o processamento computacional da linguagem humana, em seus diversos possíveis formatos. Adicionalmente, tem-se como foco o desenvolvimento de sistemas capazes de compreender, interpretar e gerar linguagem natural, sendo desafiado, principalmente, pela complexidade inerente à atribuição de significado às expressões linguísticas (CASELI; NUNES, 2024).

2.1.1 Pré Processamento

O pré-processamento de dados textuais é uma etapa essencial na aplicação de técnicas de inteligência artificial, como o Processamento de Linguagem Natural (PLN) (OLIVEIRA; NASCIMENTO, 2021). Seu objetivo principal é transformar dados brutos em um formato limpo e estruturado, eliminando ruídos e facilitando a modelagem computacional ou estatística (COPATTI, 2022). Desse modo, esse processo pode ser dividido em etapas, cada uma contribuindo para a melhoria da qualidade dos textos analisados.

Nesse aspecto, o fluxograma ilustrado na Figura 1 sintetiza as principais etapas de pré-processamento de dados textuais, proporcionando uma visão clara do processo e das operações envolvidas.

Figura 1: Fluxograma de Processamento de Linguagem Natural



Fonte: Elaboração própria.

Conforme representado na Figura 1, a primeira etapa envolve a entrada dos dados em seu formato bruto original. Posteriormente, é aplicada a remoção de caracteres especiais e números que não possuem relevância para a análise. De forma correlata, elementos como pontuação e símbolos são descartados para que o texto fique mais limpo (COPATTI, 2022).

A etapa subsequente é a normalização que visa garantir maior consistência e uniformidade nos dados textuais. Por tal motivo, essa etapa inclui, por exemplo, a remoção de acentos e a transformação das letras das palavras para minúsculas (*lowercasing*), assegurando que variações como “Casa” e “casa” sejam tratadas como equivalentes.

A próxima etapa, conhecida como tokenização, consiste em segmentar o texto em unidades menores denominadas tokens. Com isso, um token corresponde a uma palavra ou termo, sendo a menor unidade significativa do texto (STRAKA; HAJIC; STRAKOVÁ, apud COPATTI, 2022).

Após a tokenização, realiza-se a remoção de *stopwords*, que elimina palavras de pouco valor semântico, como preposições, artigos e conjunções. Embora comuns, essas palavras não agregam valor significativo à análise. A esse respeito, sua remoção torna o

texto mais focado e relevante, evitando que as etapas posteriores de processamento sejam excessivamente influenciadas por termos muito frequentes (COPATTI, 2022).

Já a lematização desempenha um papel importante ao reduzir as palavras à sua forma base, ou lema. Por exemplo, palavras como “correr”, “corre” e “correndo” são todas transformadas na forma infinitiva “correr”. Sendo assim, essa simplificação é útil para evitar redundâncias no modelo.

A etapa de *stemming* também pode vir a ser útil, complementando a lematização ao reduzir as palavras às suas raízes. Diferente da lematização, o *stemming* pode gerar formas incompletas ou truncadas das palavras, assim como é contemplado na Figura 1, mas igualmente pode contribuir para a redução da dimensionalidade e para a melhoria da performance dos modelos.

2.1.2 *Word Embeddings*

Em Processamento de Linguagem Natural (PLN), representar palavras por meio de vetores numéricos é uma etapa essencial, pois possibilita converter dados textuais em formatos, como vetores ou matrizes, que podem ser interpretados e processados por algoritmos, modelos de aprendizado de máquina e técnicas computacionais (FREITAS, 2023; NASCIMENTO, 2022).

Nesse contexto, uma abordagem que viabiliza a representação de textos em tarefas de Processamento de Linguagem Natural (PLN) é o uso de *Word Embeddings*. Essa abordagem consiste em associar a cada palavra de um vocabulário um vetor numérico, de modo que palavras com significados semelhantes possuam vetores com características similares. Assim, são preservadas relações semânticas e sintáticas entre as palavras (NASCIMENTO, 2022).

Uma técnica de obtenção dos *Word Embeddings* comumente adotada é o *Word2Vec*, que contempla o *Continuous Bag of Words (CBOW)* e o *Skip-gram* como suas arquiteturas para a modelagem de palavras. Frente a isso, o *CBOW* prevê uma palavra com base no contexto de palavras vizinhas e, por outro lado, o *Skip-gram* prevê o contexto de palavras vizinhas dada uma palavra central (MIKOLOV et al., 2013; FREITAS, 2023).

Já o *Doc2Vec (Distributed Memory Version of Paragraph Vector)*, uma extensão do *Word2Vec*, foi proposto com o objetivo de gerar representações vetoriais numéricas não apenas para palavras, mas também para documentos inteiros, preservando informações semânticas de maior escala. Assim como o *Word2Vec*, o *Doc2Vec* também possui duas arquiteturas, o *Distributed Memory (DM)* e o *Distributed Bag of Words (DBOW)* (LE; MIKOLOV, 2014; CASELI; NUNES, 2024).

Dito isso, a arquitetura *DM* funciona de maneira semelhante ao *CBOW*, incor-

porando um vetor de contexto do documento juntamente com as palavras vizinhas para prever a palavra central. Por sua vez, a arquitetura *DBOW* é análoga ao *Skip-gram*, sendo responsável por prever palavras a partir do vetor do documento.

Analogamente, o modelo *FastText*, desenvolvido pelo grupo *Facebook AI Research (FAIR)*, também surge como uma extensão do *Word2Vec* que aprimora a qualidade das representações de palavras. Diferentemente do *Word2Vec*, que trata as palavras como unidades indivisíveis, o *FastText* representa cada palavra como a soma de vetores associados a seus n-gramas de caracteres, incorporando, assim, informações sobre subpalavras e, conseqüentemente, captura informações contextuais mais ricas (JOULIN et al., 2016; CASELI; NUNES, 2024).

Em suma, a escolha da técnica mais adequada para a geração de *embeddings* de palavras exige um processo cuidadoso de investigação, experimentação e comparação de diferentes modelos (OLIVEIRA; NASCIMENTO, 2021).

2.2 Aprendizado de Máquina

A aprendizagem de máquina (*Machine Learning*), ramo da inteligência artificial, concentra-se no desenvolvimento de algoritmos capazes de aprender com experiências anteriores e tomar decisões baseadas em dados, minimizando a necessidade de interferência humana. Posto isso, essa abordagem tem-se mostrado eficaz na solução de problemas complexos, como classificação de dados, reconhecimento de padrões e previsões (FREITAS, 2023).

De acordo com Géron (2019), os métodos de aprendizado de máquina podem ser classificados em distintas categorias, considerando diferentes critérios, tais como a presença ou ausência de supervisão no processo de treinamento (aprendizado supervisionado, não supervisionado, semi-supervisionado ou por reforço), a capacidade de aprendizado incremental em tempo real (aprendizado online ou em lote), a fundamentação na comparação direta de novos dados com instâncias previamente observadas (aprendizado baseado em instâncias) ou na construção de modelos preditivos que capturam padrões subjacentes nos dados (aprendizado baseado em modelos).

2.3 Aprendizado de Máquina Supervisionado

No aprendizado de máquina supervisionado, o modelo é treinado com um conjunto de dados rotulados, ou seja, para cada entrada existe uma saída correta esperada. De maneira geral, o algoritmo precisa ser capaz de gerar respostas para novas entradas sem intervenção humana, baseando-se unicamente na experiência adquirida e nas regras

desenvolvidas a partir de um conjunto inicial de dados de treinamento (FREITAS, 2023).

Além disso, é também reforçado por Freitas (2023) que os problemas supervisionados de aprendizado de máquina podem ser classificados em dois tipos principais: classificação e regressão. Na classificação, o objetivo é prever a categoria de uma observação entre várias opções, enquanto a regressão tem como objetivo a previsão de valores contínuos. Neste trabalho, o foco será direcionado à abordagem de classificação.

2.3.1 Fluxo de Processamento

Um algoritmo de aprendizagem de máquina supervisionado, segundo Lantz, 2013, apud Freitas et al, 2024, segue uma série de etapas estruturadas, começando pela coleta de dados, que envolve a obtenção, organização e preparação das informações necessárias para o treinamento do modelo.

Em seguida, realiza-se a análise exploratória, etapa crucial para avaliar a qualidade dos dados. Nesta, são incluídos o tratamento de dados faltantes, a verificação de correlações entre variáveis e a visualização de padrões iniciais, como agrupamentos e valores atípicos (*outliers*).

Após a análise exploratória, é fundamental garantir que o modelo desenvolvido possua boa capacidade de generalização, ou seja, que mantenha um desempenho consistente quando aplicado a dados não vistos. Para isso, adota-se uma estratégia de particionamento dos dados em subconjuntos para treino, validação e teste. Conforme destacado por Géron (2019), o conjunto de treino é utilizado para o ajuste dos parâmetros internos do modelo, enquanto o conjunto de validação permite a calibração dos hiperparâmetros, como taxa de aprendizado, número de neurônios e intensidade de regularização, etapa essencial para prevenir o sobreajuste (*overfitting*). Por fim, o conjunto de teste é utilizado exclusivamente para a avaliação final, o que proporciona uma estimativa do desempenho do modelo.

Em virtude disso, a avaliação final do modelo ocorre por meio da aplicação de métricas específicas que permitem mensurar seu desempenho. Comumente, os modelos de classificação são comparados utilizando métricas como acurácia (*accuracy*), precisão (*precision*), sensibilidade (*recall*) e *F1-score* (GRUS, 2016).

Desse modo, cabe destacar que essas métricas são obtidas com base nos valores da matriz de confusão, exemplificada para classificação binária na Tabela 1, a qual é composta por quatro elementos resultantes da comparação entre os rótulos reais e os rótulos preditos pelo modelo. A classe real corresponde ao valor verdadeiro observado, enquanto a classe predita refere-se à classificação atribuída pelo modelo (GARCIA, 2021).

Tabela 1: Matriz de confusão

| Classe Real | Classe Predita: Positiva | Classe Predita: Negativa |
|-------------|--------------------------|--------------------------|
| Positiva | VP (Verdadeiro Positivo) | FN (Falso Negativo) |
| Negativa | FP (Falso Positivo) | VN (Verdadeiro Negativo) |

No que tange aos conceitos dos termos que compõem a Tabela 1, o Verdadeiro Positivo (VP) refere-se à situação em que o modelo identifica que a classe é positiva (Classe Predita: Positiva) e, ao verificar o rótulo real, confirma que a classe é, de fato, positiva (Classe Real: Positiva). Em oposição, o Verdadeiro Negativo (VN) ocorre quando o modelo classifica a classe como negativa, e ao comparar com o rótulo real, verifica-se que a classe é, de fato, negativa (FREITAS, 2023).

Analogamente, o Falso Positivo (FP) é identificado quando o modelo classifica a classe como positiva, mas ao verificar a resposta real, constata-se que a classe é negativa. Por fim, o Falso Negativo (FN) ocorre quando o modelo prediz a classe como negativa, mas ao verificar a resposta, descobre-se que a classe era positiva (FREITAS, 2023).

Além disso, vale ressaltar que, em classificações multiclasse, a matriz de confusão torna-se uma matriz quadrada $n \times n$, em que n representa o número de classes. As linhas correspondem às classes reais e as colunas às classes preditas. Os elementos da diagonal indicam os acertos, enquanto os demais representam erros de classificação entre as classes.

Definidos os termos que compõem a Tabela 1, prossegue-se com a apresentação das métricas mencionadas anteriormente. Assim, a acurácia mensura a fração de previsões corretas realizadas pelo modelo (FREITAS, 2023).

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + VN + FN}$$

É relevante salientar que em cenários com forte desbalanceamento entre as classes, algo bastante comum na prática, a acurácia pode apresentar valores elevados mesmo quando as demais métricas indicam baixo desempenho. Por essa razão, para a tarefa de Reconhecimento de Entidades Nomeadas, é comum a adoção preferencial das métricas de precisão, sensibilidade e *F1-score* (GARCIA, 2021).

Em continuidade, a precisão mede o quão precisas são as previsões positivas. Em outras palavras, ela avalia a proporção de acertos entre todas as classificações positivas identificadas pelo modelo (GARCIA, 2021).

$$\text{Precisão} = \frac{VP}{VP + FP}$$

Na sequência, tem-se a sensibilidade (*recall*), métrica que avalia a capacidade do modelo em identificar corretamente as instâncias positivas. Ou seja, mede a proporção de positivos que foram corretamente classificados (GARCIA, 2021).

$$\text{Sensibilidade} = \frac{VP}{VP + FN}$$

Finalmente, o *F1-score* é obtido como a média harmônica entre a precisão e a sensibilidade, fornecendo uma medida balanceada do desempenho do modelo, especialmente útil em contextos com classes desbalanceadas (GARCIA, 2021).

$$\text{F1-score} = 2 \cdot \frac{\text{Precisão} \cdot \text{Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}}$$

Além das métricas já abordadas, Nascimento (2022) destaca também o uso recorrente do *micro F1-score* e do *macro F1-score* em sistemas de aprendizado profundo. Dessa forma, as equações que definem as duas métricas mencionadas são apresentadas a seguir:

$$\text{Macro-F1} = \frac{1}{C} \sum_{n=1}^C \text{F1}_n$$

$$\text{Micro-F1} = \frac{VP}{VP + \frac{1}{2}(FP + FN)}$$

em que C representa o número total de classes e F1_n corresponde ao valor do F1-score calculado individualmente para cada classe n .

Sendo assim, cabe o destaque de que a métrica *Macro F1-score*, por se tratar da média simples dos F1-scores de cada classe, é mais indicada em cenários com dados desbalanceados, enquanto a *Micro F1-score* tende a ser preferida quando os dados estão balanceados (LEUNG, 2022 apud NASCIMENTO, 2022).

Após a conclusão de todas as etapas, coleta de dados, análise exploratória, treinamento e avaliação do modelo, o mesmo torna-se apto para ser aplicado (LANTZ, 2013, apud FREITAS et al., 2024).

2.3.2 Otimização da Escolha de Hiperparâmetros

Após a seleção inicial de modelos promissores, uma etapa fundamental no desenvolvimento de sistemas de aprendizado de máquina é o ajuste fino dos hiperparâmetros

(*fine-tuning*). Essa prática busca encontrar a melhor combinação de valores para parâmetros que não são aprendidos diretamente durante o treinamento, mas que impactam significativamente o desempenho final do modelo (Géron, 2019).

O ajuste manual dos hiperparâmetros, embora possível, é altamente custoso e limitado em termos de abrangência, especialmente diante de múltiplas combinações possíveis. Dado este cenário, para automatizar esse processo e garantir uma busca mais abrangente e sistemática, uma possível abordagem é o *Grid Search* (Géron, 2019).

Nesse sentido, o funcionamento geral do *Grid Search* consiste na definição de diversos valores (grade) para cada um dos hiperparâmetros. Em seguida, todas as combinações entre esses valores são elaboradas e, para cada uma delas, o modelo é treinado e avaliado. Por fim, seleciona-se a combinação que apresenta o melhor desempenho com base em uma métrica previamente definida (Géron, 2019).

2.3.3 Amostragem Estratificada

A amostragem estratificada consiste na divisão de uma população em subgrupos, denominados estratos, definidos com base em uma ou mais características conhecidas da população. A partir desses estratos, são selecionadas amostras de forma independente, respeitando proporções preestabelecidas. Desse modo, essa técnica é especialmente recomendada quando se busca melhorar a precisão das estimativas ou obter informações não apenas para a população como um todo, mas também para suas subpopulações (BOLFARINE; BUSSAB, 2004).

De acordo com Bolfarine e Bussab (2004), um dos aspectos fundamentais nesse processo é a definição do critério de alocação da amostra entre os estratos, pois essa escolha impacta diretamente na eficiência do plano amostral. Dentre os métodos de alocação, destaca-se a alocação proporcional, na qual o tamanho da amostra em cada estrato é definido de forma proporcional ao tamanho do próprio estrato na população. Desse modo, se n é o tamanho total da amostra, N o tamanho da população e N_h o tamanho do estrato h , então o número de unidades selecionadas no estrato h é dado por:

$$n_h = n \cdot \frac{N_h}{N}$$

2.3.4 Validação Cruzada

Conforme destacado em Hastie, Tibshirani e Friedman (2009), em um cenário ideal, seria possível reservar um subconjunto de dados exclusivamente para avaliação, de modo a aferir o desempenho do modelo. Contudo, na prática, essa abordagem frequente-

mente se mostra inviável devido à escassez de dados.

Para contornar essa limitação, emprega-se o procedimento denominado *K-fold Cross-Validation*, no qual os dados são particionados em K subconjuntos aproximadamente do mesmo tamanho e sem sobreposição. Posteriormente, o modelo é ajustado iterativamente utilizando-se $K - 1$ desses subconjuntos como dados de treino, enquanto o subconjunto remanescente é utilizado para teste, sendo nesse avaliado o desempenho por meio das métricas de interesse. Ao final das K iterações, calcula-se a média dessas métricas, obtendo-se uma estimativa final do desempenho do modelo (LOCA, 2023).

Por outro lado, a validação cruzada estratificada (*StratifiedKFold*) é uma extensão da abordagem tradicional, sendo particularmente relevante quando as classes da variável resposta apresentam distribuições desbalanceadas. Segundo Géron (2019), esse método assegura que cada partição (ou *fold*) preserve aproximadamente a mesma proporção de instâncias de cada classe existente no conjunto de dados por meio de uma amostragem estratificada.

2.3.5 Tratamento de Bases Desbalanceadas

Técnicas de balanceamento de dados são amplamente empregadas para mitigar os efeitos do desbalanceamento de classes em tarefas de classificação. Dentre as abordagens mais consolidadas na literatura, destacam-se o *oversampling* (sobreamostragem), o *undersampling* (subamostragem) e a combinação de ambos (CORDEIRO, 2020).

O *oversampling*, também conhecido como *upsample*, busca aumentar a representatividade da classe minoritária por meio da replicação de dados ou da geração de novos exemplos (CORDEIRO, 2020; FREITAS, 2023).

Por outro lado, o *undersampling*, recomendado para quando o tamanho do conjunto de dados é consideravelmente expressivo, visa reduzir a quantidade de exemplos da classe majoritária, promovendo o equilíbrio entre as classes por meio da eliminação de possíveis redundâncias (CORDEIRO, 2020).

Adicionalmente, há a possibilidade de aplicar a combinação de *oversampling* e *undersampling*, aproveitando os benefícios de ambas as abordagens de forma conjunta. Essa estratégia híbrida busca não apenas aumentar a representatividade da classe minoritária, mas também eliminar redundâncias da classe majoritária, promovendo um melhor desempenho do modelo em tarefas de predição (CORDEIRO, 2020).

2.4 Aprendizado de Máquina Não Supervisionado

O aprendizado não supervisionado consiste em uma abordagem da aprendizagem de máquina na qual os modelos são treinados utilizando dados não rotulados, ou seja, sem informações explícitas sobre os resultados esperados. Diferentemente do aprendizado supervisionado, o objetivo central é identificar padrões ocultos, detectar anomalias, verificar agrupamentos (*clusters*), reduzir dimensionalidade e viabilizar a visualização dos dados ou relações relevantes entre eles (GÉRON, 2019).

Sob essa perspectiva, por sua natureza exploratória, o aprendizado não supervisionado desempenha um papel fundamental na compreensão inicial de bases de dados, especialmente em cenários em que a obtenção de rótulos é inviável devido a custos elevados ou ao tempo demandado no processo de anotação.

2.4.1 *K-Means*

O algoritmo particional *K-Means* é uma técnica empregada em tarefas de aprendizado de máquina não supervisionado, sendo utilizada para agrupar dados quantitativos em k grupos distintos. Desse modo, sua popularidade se deve, principalmente, à combinação entre simplicidade operacional e eficiência na formação de agrupamentos (MAGALHÃES, 2020).

Diante disso, seu funcionamento baseia-se inicialmente na seleção de k pontos do conjunto de dados como centróides iniciais. Em seguida, cada ponto é atribuído ao grupo cujo centróide estiver mais próximo, com base em uma medida de distância. A partir disso, os centróides são atualizados como a média dos pontos alocados a cada grupo. Dessa forma, esse processo é repetido iterativamente até que não haja mais mudanças nas atribuições ou seja atingido um número máximo de iterações (GOLDSCHMIDT; PASSOS; BEZERRA, 2015 apud MAGALHÃES, 2020)

De maneira recorrente, no *K-Means*, utiliza-se a distância euclidiana como medida para calcular o quão distantes dois pontos estão em um espaço multidimensional. No caso do agrupamento de textos, esse espaço é composto pelos seus vetores representativos (*embeddings*), em que cada dimensão representa uma informação derivada das palavras presentes nos textos (MAGALHÃES, 2020).

A distância euclidiana entre dois vetores x e y , com n componentes, representa a menor distância entre os pontos, calculada pela raiz quadrada da soma dos quadrados das diferenças entre suas coordenadas:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Uma abordagem popularmente empregada para determinar o número ideal de grupos (k) no algoritmo *K-Means* é o método do cotovelo. Assim, ele consiste em executar o algoritmo para diferentes valores de k e calcular, para cada um, a soma dos quadrados das distâncias do centróide para os pontos de dentro de cada cluster (WCSS). Ao representar esses valores em um gráfico, observa-se uma curva decrescente, na qual o ponto de inflexão indica o valor de k a partir do qual acréscimos não geram melhorias significativas na compactação dos grupos, sugerindo assim o número ótimo de clusters (L.; OBUNADIKE, 2022).

2.4.2 Similaridade Cosseno

A similaridade cosseno é uma medida que avalia o cosseno do ângulo formado entre dois vetores em um espaço multidimensional. Seu valor varia de 0 a 1, sendo que 1 indica que os vetores possuem a mesma direção, ou seja, são totalmente semelhantes, enquanto 0 representa vetores completamente distintos (FREITAS, 2023).

Dados dois vetores, X e Y , a similaridade cosseno pode ser expressa pelo produto escalar, conforme a equação:

$$\text{similaridade} = \cos(\theta) = \frac{X \cdot Y}{|X| \cdot |Y|}$$

A título de exemplo, em Freitas (2023), a similaridade cosseno é empregada para classificar processos semelhantes por meio da comparação entre representações vetoriais de textos. Especificamente, a abordagem adotada consistiu em identificar, para cada texto não rotulado, o texto etiquetado mais similar com base nessa medida, atribuindo-lhe a etiqueta correspondente. Por sua vez, em Oliveira e Nascimento (2021), a técnica é utilizada com o propósito de mensurar a distância entre agrupamentos.

2.4.3 Análise de Componentes Principais

A Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis*) é uma técnica estatística multivariada utilizada para reduzir a dimensionalidade de dados ao transformar variáveis possivelmente correlacionadas em um novo conjunto de variáveis não correlacionadas, chamadas de componentes principais (MORAIS, 2011).

Assim, os componentes principais são combinações lineares das variáveis originais

e possuem a propriedade de serem ortogonais entre si, o que elimina a redundância de informação. Conforme descrito por MORAIS (2011), seja $\mathbf{X} \in \mathbb{R}^{n \times m}$ uma matriz de dados com n observações e m variáveis, as etapas da PCA podem ser descritas pelo seguinte processo sequencial:

1. Normalização dos dados: Garante que variáveis com diferentes escalas não exerçam influências desproporcionais na análise.

$$z_i = \frac{x_i - \mu}{\sigma},$$

em que z_i representa o valor padronizado, x_i é a i -ésima observação, μ corresponde à média amostral da variável, e σ ao seu desvio padrão amostral. Assim, todas as variáveis passam a apresentar média zero e desvio padrão unitário.

2. Cálculo da matriz de covariância: a matriz de covariância \mathbf{C}_Z das variáveis padronizadas é então calculada como:

$$\mathbf{C}_Z = \mathbf{Z}^\top \mathbf{Z}$$

em que \mathbf{Z}^\top representa a transposta da matriz \mathbf{Z} .

3. Identificação de autovalores e autovetores: Em seguida, realizam-se os cálculos dos autovalores e dos autovetores da matriz de covariância. Dessa maneira, os autovalores indicam a quantidade de variância explicada por cada componente principal, enquanto os autovetores representam a contribuição de cada variável original na definição da direção das componentes principais (SILVA et al., 2005 apud MORAIS, 2011).
4. Ordenação e seleção das componentes principais: Os autovalores são ordenados em ordem decrescente, e os autovetores correspondentes formam as colunas da matriz \mathbf{P} :

$$\mathbf{P} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_k],$$

5. Diagonalização : O último passo consiste na mudança de base da matriz de covariância \mathbf{C}_Z por meio da matriz de autovetores \mathbf{P} , obtendo-se uma matriz diagonal \mathbf{D} contendo os autovalores de \mathbf{C}_Z :

$$\mathbf{D} = \mathbf{P}^{-1} \mathbf{C}_Z \mathbf{P}.$$

2.5 Reconhecimento de Entidades Nomeadas (REN)

O Reconhecimento de Entidades Nomeadas (REN), ou *Named Entity Recognition (NER)*, é uma técnica dentro do campo de Processamento de Linguagem Natural (PLN), voltada para a identificação e classificação de entidades mencionadas em textos. No contexto jurídico, por exemplo, esse método se destaca ao identificar entidades como nomes de juízes, advogados e outros profissionais do direito em documentos legais, atribuindo um nível adicional de semântica aos dados extraídos (CASTRO, 2019).

O processamento de linguagem natural demanda modelos capazes de capturar a dependência contextual entre as palavras em uma sequência. Isso se dá porque o significado de uma palavra, muitas vezes, depende diretamente das palavras que a antecedem ou sucedem. Nesse contexto, modelos baseados em sequências tornam-se essenciais, especialmente em tarefas como o Reconhecimento de Entidades Nomeadas (REN), nas quais a ordem e a relação entre os termos carregam informações relevantes (NASCIMENTO, 2022).

Dessa maneira, as Redes Neurais Recorrentes do tipo *Long Short-Term Memory (LSTM)*, introduzidas por Hochreiter e Schmidhuber (1997), foram desenvolvidas com o propósito de superar limitações das redes recorrentes convencionais, particularmente no que diz respeito à dificuldade de aprender dependências de longo prazo em sequências de dados (NASCIMENTO, 2022). A principal inovação da *LSTM* reside na incorporação de células de memória responsáveis por armazenar informações ao longo do tempo. Além disso, os portões de esquecimento, introduzidos por Gers e Schmidhuber (2000), viabilizam o controle de informações a serem deletadas (NASCIMENTO, 2022).

Adicionalmente, as Redes Neurais Recorrentes Bidirecionais (*Bidirectional Long Short-Term Memory (BiLSTM)*) representam uma extensão das LSTMs tradicionais, desenvolvidas para processar sequências de dados em ambas as direções. Desse modo, essa arquitetura permite capturar informações contextuais tanto anteriores quanto posteriores ao tempo t . Além disso, os rótulos de saída do *BiLSTM* para cada posição t podem ser obtidos pela concatenação dos estados ocultos gerados nas duas direções (COSTA, 2023).

Ambas as técnicas, *LSTM* e *BiLSTM*, apresentam-se como abordagens eficazes no tratamento de sequências em tarefas de Processamento de Linguagem Natural, como o Reconhecimento de Entidades Nomeadas (REN). Além de seus fundamentos teóricos amplamente discutidos na literatura, essas arquiteturas estão implementadas e disponíveis para uso por meio de bibliotecas consolidadas na linguagem *Python*, como *TensorFlow* e *PyTorch*, o que facilita sua aplicação prática (GéRON, 2019).

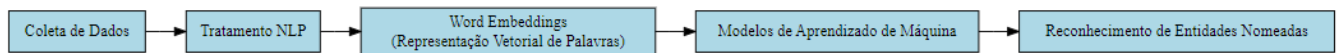
3 Metodologia

Nesta seção, são apresentadas as etapas necessárias para a obtenção dos resultados deste estudo, com base na aplicação das técnicas previamente descritas na Seção 2.

Os processos computacionais desenvolvidos foram implementados utilizando a linguagem de programação *Python* (versão 3.10) em *Google Colab*, plataforma de desenvolvimento em nuvem. Entre as principais bibliotecas utilizadas, destacam-se *Numpy* e *Pandas*, para manipulação e análise de dados, *Sklearn*, para implementação de algoritmos de aprendizado de máquina, *Spacy*, *Nltk* e *Tensorflow*, direcionadas ao pré-processamento, análise linguística e modelagem profunda de textos.

O fluxograma ilustrado na Figura 2 sintetiza a metodologia adotada neste trabalho, ilustrando o processo sequencial de aplicação das técnicas descritas.

Figura 2: Síntese metodológica



Fonte: Elaboração própria.

3.1 Coleta de Dados


O conjunto de dados utilizado neste trabalho compreende portarias emitidas em 2024 pelo Gabinete da Presidência do Tribunal de Justiça do Distrito Federal e dos Territórios (TJDFT). Tendo isso em vista, essas portarias foram coletadas diretamente das publicações oficiais disponíveis no site do Tribunal, utilizando técnicas de raspagem de dados (*web scraping*). Para isso, foram utilizadas as bibliotecas *requests*, *time* e, do pacote *bs4*, a biblioteca *BeautifulSoup*.

A prática de *web scraping* tem sido amplamente adotada devido à sua capacidade de automatizar a coleta de informações de páginas da web, transformando esses dados em formatos estruturados, comumente sem a necessidade de intervenção manual, o que a torna uma ferramenta prática e eficiente para extração de dados (Bhardwaj et al., 2021, apud Oliveira, 2023).

A Figura 3, referente ao conteúdo da Portaria 1963, tem por objetivo ilustrar o formato padrão das portarias analisadas.

Figura 3: Portaria exemplo

Portaria GPR 1963 de 30/12/2024



Poder Judiciário da União
Tribunal de Justiça do Distrito Federal e dos Territórios
Secretaria-Geral do Tribunal de Justiça do Distrito Federal e dos Territórios

PORTARIA GPR 1963 DE 30 DE DEZEMBRO DE 2024

O PRESIDENTE DO TRIBUNAL DE JUSTIÇA DO DISTRITO FEDERAL E DOS TERRITÓRIOS, no uso de suas atribuições legais, nos termos do art. 262-A, § 3º, do [Provimento-Geral da Corregedoria aplicado aos Serviços Notariais e de Registro](#), e em vista do contido no processo SEI 0039527/2024,

RESOLVE:

Art. 1º Exonerar, a pedido, Cláudia Cruz Cerquinho de Oliveira do cargo de juíza de paz 1ª suplente do 3º Ofício de Registro Civil, Títulos e Documentos e Pessoas Jurídicas do Paranoá, a partir de 22 de dezembro de 2024.

Art. 2º Declarar vago o cargo de juiz de paz 1ª suplente do 3º Ofício de Registro Civil, Títulos e Documentos e Pessoas Jurídicas do Paranoá, a ser preenchido mediante remoção entre os juizes de paz suplentes em exercício no Distrito Federal Parágrafo único. Os interessados deverão requerer inscrição no prazo de 10 dias a contar da publicação desta Portaria.

Art. 3º Esta Portaria entra em vigor na data de sua publicação.

Desembargador WALDIR LEÔNIO JÚNIOR
Presidente

Fonte: Site TJDF.

Com o intuito de garantir a extração apenas de conteúdos pertinentes das portarias, empregaram-se expressões regulares (*regex*), que possibilitam a pesquisa por padrões de textos, para identificar e delimitar informações específicas, como número da portaria, data de emissão e conteúdo principal. Essa abordagem assegurou a coleta precisa dos elementos necessários e assegurou que os dados estivessem alinhados com os objetivos do estudo, excluindo partes irrelevantes, como, por exemplo, rodapés e cabeçalhos. A biblioteca utilizada foi a *re*.

3.2 Pré Processamento e Limpeza dos Dados

O tratamento dos dados textuais seguiu o conjunto de procedimentos descritos na Seção 2 para preparar os dados para as análises subsequentes. Primeiramente, os textos foram normalizados para garantir a uniformidade e evitar a duplicidade de palavras com variações de letras maiúsculas e minúsculas.

Em sequência, a remoção de *stopwords* foi realizada utilizando o pacote *NLTK* do *Python*, que fornece uma lista de palavras comuns em português, como "o", "a", "de", "em", entre outras. Além dessas palavras, foram excluídas siglas recorrentes no texto, como "art", "caput", "cc" e caracteres especiais como "§", "º", "ª", "§§", "§º", "ª".

Em seguida, foi realizada a remoção de pontuações, que inclui símbolos como pontos, vírgulas e outros sinais que não contribuem para a análise textual. Além disso, a remoção de números e números romanos (I, II, III, IV, V, etc.) também foi realizada. Por fim, foi conduzida a remoção da assinatura do presidente, que, por estar presente em todas as portarias, foi considerada como uma informação que não agrega potencial discriminativo ou informativo.

Como etapa final, a *tokenização* foi aplicada para dividir os textos em unidades menores, facilitando as posteriores etapas de análise, como a vetorização numérica.

Para a etapa de Reconhecimento de Entidades Nomeadas, em adição ao processo descrito até então, aplicou-se o mesmo protocolo de pré-processamento aos textos descritivos das portarias. A Figura 4 exemplifica o texto descritivo da Portaria 1963, ilustrando para quais dados textuais os tratamentos também foram aplicados.

Figura 4: Descrição da portaria exemplo

Portaria GPR 1963 de 30/12/2024

Exonera, a pedido, Claudia Cruz Cerquinho de Oliveira do cargo de juíza de paz 1ª suplente do 3º Ofício de Registro Civil, Títulos e Documentos e Pessoas Jurídicas do Paranoá.

Fonte: Site TJDFT.

3.3 Representação e Modelagem dos Dados

Com os dados devidamente tratados, conforme detalhado anteriormente na Seção 3.2, procedeu-se à vetorização textual numérica com o uso dos modelos *Word2Vec*, *Doc2Vec* e *FastText*. A implementação foi realizada por meio das bibliotecas *gensim* e *fasttext*.

A definição dos hiperparâmetros dos modelos foi conduzida com o auxílio da técnica de *grid search*. A faixa de variação dos hiperparâmetros foi definida com base em alterações próximas aos valores padrão das bibliotecas utilizadas e variações pertinentes às características dos dados em questão, em especial no que diz respeito aos seus tamanhos reduzidos, por se tratarem de portarias.

Além disso, vale ressaltar que as variações dos hiperparâmetros, de modo geral, foram definidas de forma alinhada com as particularidades de cada modelo, com o objetivo de assegurar uma comparação válida entre os resultados obtidos. Desse modo, para os mo-

delos *Word2Vec* e *Doc2Vec*, foram adotados hiperparâmetros comuns, como `min_count=2` e `workers=4`, que controlam, respectivamente, o número mínimo de ocorrências para que uma palavra seja considerada e o número de núcleos de processamento utilizados durante o treinamento. Por outro lado, o modelo *FastText*, implementado por meio da biblioteca *fasttext*, não possui exatamente esses mesmos hiperparâmetros ou equivalentes diretos, sendo, portanto, excluído dessas configurações compartilhadas.

A seguir, apresentam-se as variações implementadas para cada hiperparâmetro, acompanhadas de uma breve explicação de sua funcionalidade. Entre parênteses, indica-se o hiperparâmetro correspondente no *FastText*.

- `vector_size` (dim para *fasttext*) = 100 e 200: define a dimensão dos vetores de palavras.
- `window` (ws para *fasttext*)= 5 e 10: especifica o tamanho da janela de contexto simétrica, ou seja, determina quantas palavras antes e depois da palavra-alvo o modelo considera como contexto.
- `epochs` (epoch para *fasttext*) = 5, 10, 15, 20, 40 e 60: define o número de vezes que o modelo percorre todos os dados durante o treinamento.

Além desses, também foram incluídos no processo de *grid search* os seguintes hiperparâmetros que definem o tipo de arquitetura utilizada.

- `sg` = 1 ou 0 no *Word2Vec*.
O valor 1 é referente ao *Skip-gram* enquanto o valor 0 implementa o *Continuous Bag of Words (CBOW)*.
- `dm` = 1 ou 0 no *Doc2Vec*.
O hiperparâmetro `dm=1` corresponde ao modelo *Distributed Memory (DM)*. Já `dm=0` refere-se ao *Distributed Bag of Words (DBOW)*.
- `model` = "skipgram" ou "cbow" no *FastText*.
As possíveis arquiteturas deste modelo são análogas às do *Word2Vec*, sendo `model` = "skipgram" correspondente a implementação do *Skip-gram* e `model` = "cbow" do *Continuous Bag of Words (CBOW)*.

Para avaliar a capacidade dos *embeddings*, gerados pelos modelos correspondentes a cada uma das combinações possíveis de hiperparâmetros obtidas no processo de *grid search*, de capturar relações semânticas e sintáticas, adotou-se a avaliação intrínseca como estratégia principal. Posteriormente, com o intuito de reforçar a verificação da coerência

dos *embeddings* produzidos pelo modelo final, configurado com os hiperparâmetros definidos, foi realizada a avaliação extrínseca.

Nesse sentido, na avaliação intrínseca são analisadas diretamente a capacidade dos *embeddings* em capturar relações sintáticas ou semânticas entre termos textuais, avaliando sua coerência e representatividade. Por outro lado, na avaliação extrínseca os *embeddings* são utilizados como características de entrada em tarefas externas e o desempenho do modelo funciona como um indicador da qualidade dos *embeddings* (SCHNABEL et al., 2015).

Portanto, no presente trabalho, a validação intrínseca foi conduzida por meio da verificação empírica da coerência semântica e sintática das similaridades, mensuradas pela similaridade cosseno entre a portaria 1963 e as dez portarias mais semelhantes a ela, com base na leitura dos respectivos conteúdos destes documentos. Diante disso, vale destacar que a portaria 1963 foi escolhida devido ao conhecimento prévio sobre seu conteúdo, o que facilitou a comparação observacional dos resultados de similaridade.

Já a validação extrínseca baseou-se na verificação da consistência dos resultados obtidos nos processos de agrupamento e reconhecimento de entidades nomeadas, utilizando os *embeddings* gerados. A ideia central é de que, caso as técnicas de agrupamento consigam reunir portarias semanticamente semelhantes e, de forma análoga, o modelo de Reconhecimento de Entidades Nomeadas (REN), com o auxílio dos *embeddings*, apresente bom desempenho na classificação das entidades, tem-se um indicativo da qualidade das representações geradas. Vale destacar que a própria utilização do aplicativo web (Seção 4.7) proposta neste trabalho contribuiu para os processos de avaliação, tanto intrínseca quanto extrínseca.

De maneira complementar, para medir a coesão dos *embeddings*, utilizou-se a média das similaridades cosseno entre todos os pares possíveis de portarias, excluindo as autossimilaridades, com a finalidade de se ter uma medida geral da consistência semântica. Além disso, foi avaliada também a média das similaridades entre cada portaria e suas 10 portarias mais similares, com o fito de identificar quão bem o modelo captura a proximidade entre vizinhos semânticos. Desse modo, vale o destaque de que na Seção 4 essas médias são referenciadas como coesão global e coesão local.

No que tange à visualização e interpretação dos padrões de proximidade entre os *embeddings* gerados, foi empregada a análise de componentes principais para transformar o espaço vetorial original em um novo espaço bidimensional, representado pelas duas primeiras componentes principais que mais explicam a variância dos dados. Essa redução de dimensionalidade viabilizou a observação de relações semânticas entre as portarias e agrupamentos. Desse modo, sua implementação, bem como a dos procedimentos subsequentes descritos nesta seção, foram implementados com o uso da biblioteca *scikit-learn*.

Finalmente, após a definição do modelo de vetorização ideal e a consequente obtenção dos vetores numéricos representativos de cada portaria, aplicou-se a normalização dos *embeddings*, com o objetivo de garantir a comparabilidade entre os três métodos de geração, permitir o uso consistente da distância euclidiana no *K-Means* em relação à similaridade cosseno e assegurar uma redução de dimensionalidade coerente. Desse modo, com o número ótimo de clusters determinado pelo método do cotovelo, procedeu-se à aplicação do algoritmo de agrupamento *K-Means*.

3.4 Reconhecimento de Entidades Nomeadas

Para estabelecer as categorias de entidades a serem identificadas, utilizou-se como referência a metodologia *5W2H*, tradicionalmente aplicada em processos de diagnóstico e planejamento estratégico, por sua capacidade de estruturar informações com base em definições fundamentais, sendo elas, o que (*what*), por que (*why*), onde (*where*), quando (*when*), quem (*who*), como (*how*) e quanto (*how much*).

A adaptação da abordagem mencionada às definições cabíveis ao contexto do presente estudo resultou na definição de quatro categorias de entidades de interesse. A primeira delas, ACAA, corresponde a verbos ou locuções verbais e seu objeto direto que expressam atos presentes nas portarias. Complementarmente, a classe SUJEITO refere-se a nomes próprios de pessoas ou designações institucionais, englobando os agentes responsáveis pelos atos ou aqueles diretamente impactados. A entidade DATA abrange todas as expressões que fazem referência a momentos temporais específicos. E, por fim, a entidade LOCAL corresponde a menções geográficas, organizacionais ou unidades administrativas que são mencionadas nas portarias.

Assim, após a definição das entidades de interesse em conjunto aos agrupamentos obtidos por meio do algoritmo *k-Means*, procedeu-se à realização de uma amostragem estratificada proporcional ao número de portarias presentes em cada cluster. Essa estratégia teve como objetivo assegurar que a diversidade temática dos dados estivesse adequadamente representada durante o processo de anotação manual das entidades.

A anotação das entidades foi feita com os textos de descrição das portarias e foi estruturada em formato de dicionário *python*, no qual cada chave representa o número identificador da portaria e os valores associados contêm a respectiva descrição, as entidades anotadas e suas respectivas categorias.

A seguir na Figura 5, apresenta-se um exemplo ilustrativo da estrutura adotada para a anotação e armazenamento:

Figura 5: Exemplo anotação

```
"1418": {  
  "text": "Estabelece a escala de plantão judicial do Conselho da  
Magistratura do Tribunal de Justiça do Distrito Federal e Territórios, nos  
dias 22 e 23 de junho de 2024.",  
  "labels": [  
    {"text": "Estabelece a escala de plantão judicial", "label": "ACAO"},  
    {"text": "Conselho da Magistratura do Tribunal de Justiça do Distrito  
Federal e Territórios", "label": "LOCAL"},  
    {"text": "22 e 23 de junho de 2024", "label": "DATA"}  
  ]  
},
```

Fonte: Elaboração própria.

Em razão do elevado custo de tempo e esforço envolvidos no processo de anotação, optou-se por selecionar aproximadamente 43% das portarias, utilizando amostragem estratificada a fim de manter essa mesma proporção em cada um dos clusters identificados. Como resultado, foram anotadas manualmente 735 portarias no total.

Após o processo de anotação, os dados foram tokenizados com o uso da biblioteca *spaCy* e alinhados aos seus respectivos rótulos de entidades. Em seguida, os textos foram convertidos em vetores numéricos por meio do modelo de *embedding* previamente definido como ideal, possibilitando sua utilização no modelo de aprendizado supervisionado para o Reconhecimento de Entidades Nomeadas (REN). Nesse modelo, cada token foi classificado de acordo com a entidade à qual pertence ou como não pertencente a nenhuma entidade. O modelo adotado utiliza duas camadas LSTM bidirecionais (BiLSTM) e incorpora a técnica de *dropout*, que consiste em desativar aleatoriamente uma fração dos neurônios durante o treinamento, com o objetivo de reduzir o risco de sobreajuste (*overfitting*). Por fim, a função *softmax* foi aplicada na camada de saída para prever o rótulo correspondente a cada token. Essas implementações foram realizadas por meio da biblioteca *TensorFlow*.

De maneira aditiva, para verificar a robustez do modelo frente à variação dos dados, foram conduzidas validações cruzadas simples e estratificadas com 5 particionamentos (*folds*). No que se refere ao processo de treinamento, aplicou-se a técnica de *early stopping* também com o objetivo de evitar o sobreajuste (*overfitting*) aos dados de treinamento. Quanto à avaliação, foram calculadas a média e o desvio padrão das métricas *macro F1-score* e *micro F1-score*, bem como o *F1-score* médio por entidade, permitindo uma análise mais completa e robusta do desempenho do modelo.

Por fim, diante do desbalanceamento entre as quantidades de entidades anotadas, foi aplicada a técnica de aumento de dados por reamostragem (*oversampling*), duplicando-se as portarias que continham entidades menos representadas no conjunto de portarias anotadas. Desse modo, para mitigar possíveis problemas de sobreajuste aos dados de trei-

namento (*overfitting*) ou inserção descontrolada de vieses, decorrentes dessa abordagem, optou-se por realizar apenas a duplicação desses exemplos.

3.5 Aplicativo Web

Com o intuito de viabilizar a exploração interativa e a disseminação facilitada dos resultados obtidos neste estudo, foi desenvolvido um ambiente web utilizando a biblioteca *Streamlit*. O site serve como uma interface acessível para consulta e análise dos principais achados deste trabalho, consolidando as etapas de similaridade semântica, agrupamentos e reconhecimento de entidades nomeadas nas portarias analisadas.

4 Resultados

Nesta seção, são apresentados os resultados da implementação dos processos descritos na Seção 3 e suas respectivas análises.

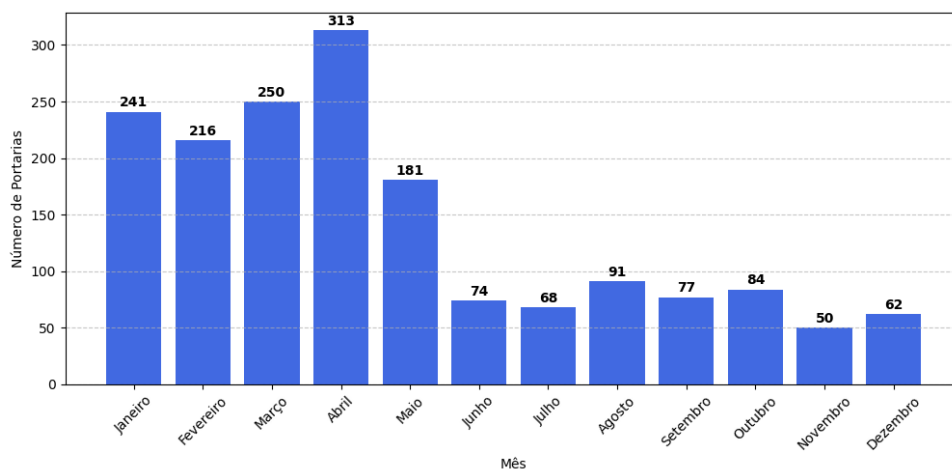
4.1 Análise Exploratória dos Dados

O banco de dados gerado a partir do processo de raspagem de dados (*web scraping*) contém o total de 1.707 portarias publicadas ao longo do ano de 2024. A primeira portaria foi registrada na data de 2 de janeiro, enquanto a última foi publicada em 30 de dezembro.

A última portaria registrada foi identificada com o número 1963, o que evidencia a existência de numerações ausentes ao longo do período analisado. Nesse sentido, vale ressaltar que esse fato pode estar relacionado a diferentes fatores, como portarias anuladas ou atualizadas por portarias posteriores.

No que se refere à distribuição temporal, o gráfico apresentado na Figura 6 ilustra as variações no volume de portarias emitidas ao longo dos meses, permitindo uma visualização clara das oscilações mensais.

Figura 6: Distribuição mensal das portarias



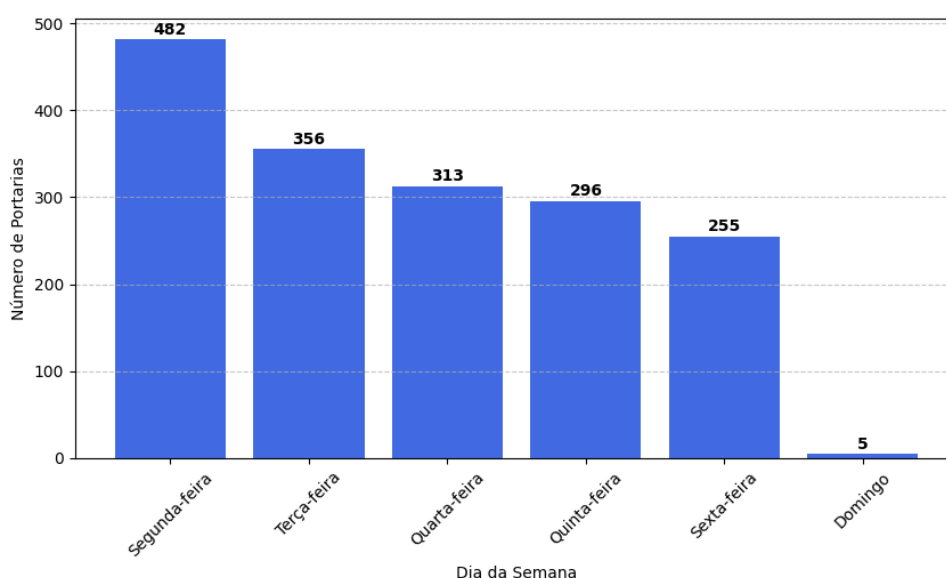
Fonte: Elaboração própria.

Nota-se que abril concentrou o maior número de publicações, totalizando 313 (18,33%) portarias, enquanto novembro registrou a menor quantidade, com apenas 50 (2,92%) publicações. Além disso, observa-se a distribuição não homogênea na publicação das portarias ao longo dos meses, com notável concentração nos primeiros cinco meses do

ano.

Já a Figura 7 reflete a quantidade de portarias publicadas por dias da semana, fornecendo informações sobre o fluxo de trabalho e a intensidade das publicações ao longo dos dias.

Figura 7: Distribuição semanal das portarias



Fonte: Elaboração própria.

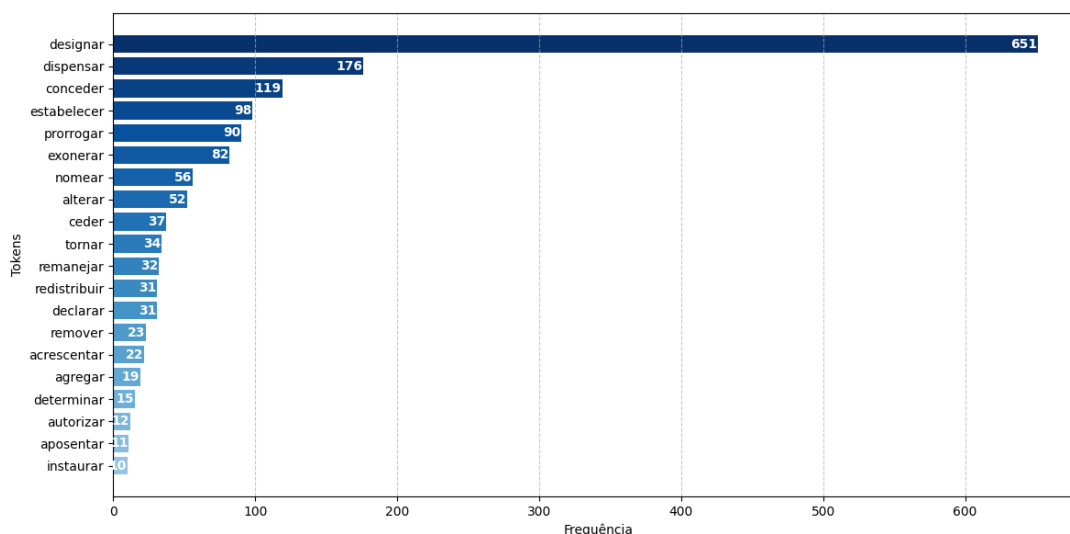
A análise da distribuição semanal das portarias evidencia um pico significativo de publicações às segundas-feiras (28,23%), seguido por uma queda gradual ao longo da semana. A ausência de publicações aos sábados e a quantidade mínima registrada aos domingos reforçam o caráter institucional, cujo trabalho se concentra nos dias úteis.

Outrossim, a fim de caracterizar os dados textuais resultantes das etapas de pré-processamento e limpeza descritas na Seção 3, apresenta-se a seguir a Tabela 2, que resume estatisticamente a distribuição da quantidade de tokens por portaria.

Tabela 2: Estatísticas descritivas da quantidade de tokens por portaria

| Estatística | Valor |
|---------------|--------|
| Média | 119,71 |
| Mediana | 66 |
| Mínimo | 30 |
| Máximo | 4161 |
| Desvio padrão | 216,13 |

Figura 9: Gráfico de barras dos 20 primeiros verbos mais frequentes

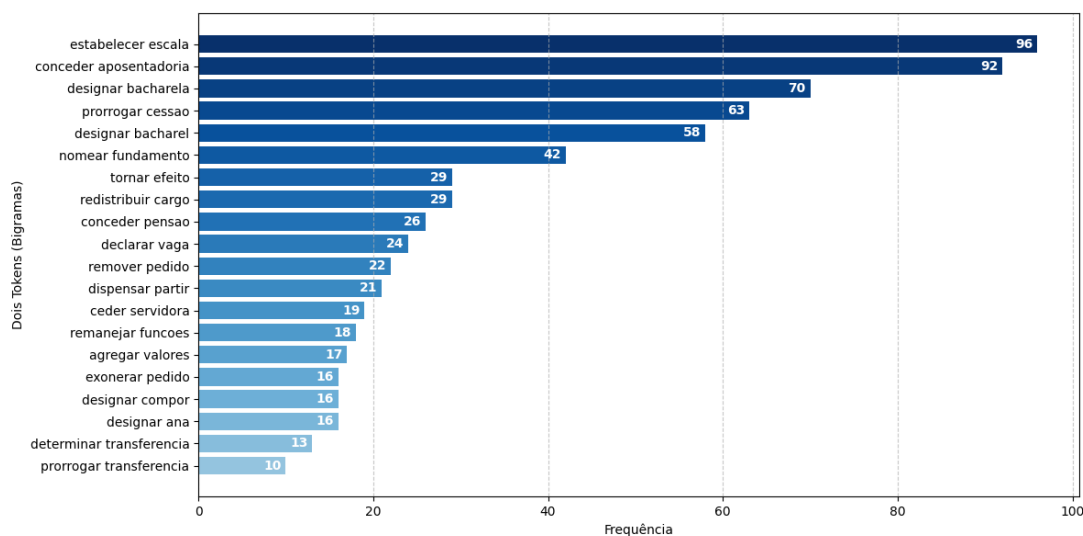


Fonte: Elaboração própria.

Entre os verbos mais frequentes, “designar” se destaca amplamente, com 651 ocorrências (38,13%), seguido por “dispensar” (176; 10,31%) e “conceder” (119; 6,97%). Posteriormente, a frequência dos demais verbos diminui gradativamente, o que reforça a predominância de portarias voltadas à designação.

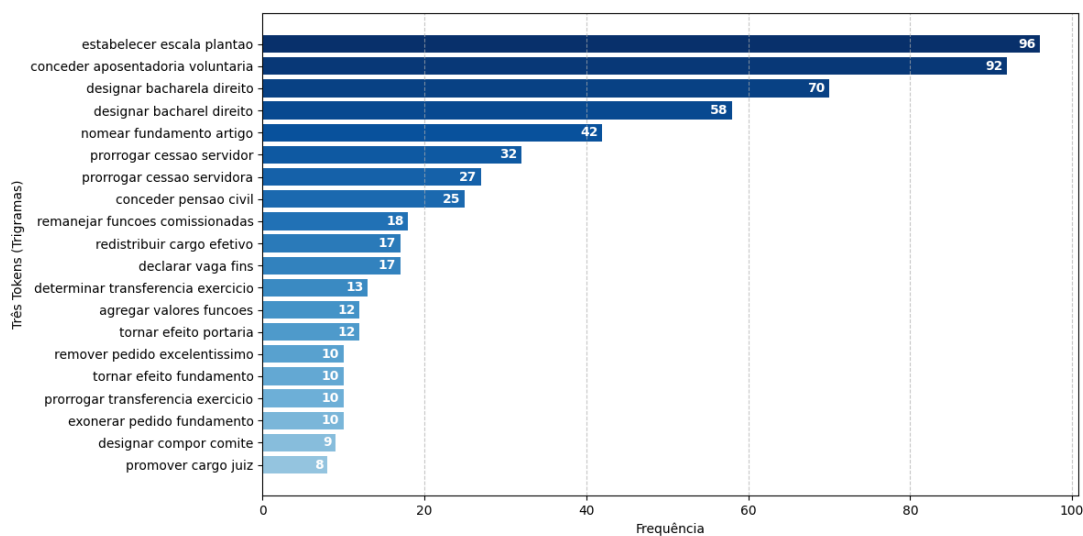
Com o intuito de aprofundar a compreensão dos atos das portarias analisadas, foi realizada uma análise de bigramas e trigramas mais frequentes dos primeiros tokens após “RESOLVE:”, ou seja, sequência de duas e três palavras que ocorrem conjuntamente. O gráfico referido pela Figura 10 ilustra a distribuição dos bigramas com maior frequência nas portarias sob análise e o apresentado pela Figura 11 mostra os trigramas.

Figura 10: Gráfico de barras dos 20 primeiros bigramas mais frequentes



Fonte: Elaboração própria.

Figura 11: Gráfico de barras dos 20 primeiros trigramas mais frequentes



Fonte: Elaboração própria.

A análise conjunta das Figuras 10 e 11 detalha e evidencia a continuidade dos verbos presentes na Figura 9, com recorrência de combinações como “designar bacharel direito”, “conceder aposentadoria voluntária”, “estabelecer escala plantão” e “prorrogar cessão”.

Assim, diante do exposto nesta seção, tem-se um entendimento abrangente sobre a distribuição das portarias ao longo do ano de 2024, bem como sobre os termos e atos mais recorrentes nesses documentos.

4.2 Definição do Modelo de Embedding

A Tabela 3 apresenta as combinações de hiperparâmetros, resultantes do processo de *grid search*, que sucederam em similaridades semanticamente coerentes entre as 10 portarias mais similares à portaria 1963, bem como as medidas adicionais que indicam coesão semântica.

Tabela 3: Modelos avaliados, hiperparâmetros ótimos e métricas de coesão

| Modelo | Hiperparâmetros | Global | Local |
|-----------------------------|--|--------|--------|
| Word2Vec | <code>vector_size=200, window=5, epochs=5, sg=1</code> | 0,8650 | 0,9918 |
| Doc2Vec | <code>vector_size=200, window=5, epochs=5, dm=0</code> | 0,8604 | 0,9907 |
| FastText (Skip-gram) | <code>dim=200, ws=5, epoch=5</code> | 0,8575 | 0,9913 |

Nota-se uma convergência dos hiperparâmetros avaliados como ideais entre os três modelos, especialmente quanto ao tamanho do vetor (`vector_size/dim = 200`), à janela de contexto (`window/ws = 5`) e ao número de épocas (`epochs/epoch = 5`). Além disso, no que diz respeito às arquiteturas, os modelos *Word2Vec* e *FastText* apresentaram resultados mais coerentes ao utilizarem o *Skip-gram*, semelhantemente o *Doc2Vec* obteve melhor desempenho com a variante *DBOW - Distributed Bag of Words*, funcionalmente análoga ao *Skip-gram*.

Desse modo, com base também nos valores médios, global e local, de similaridade elevados, conclui-se que essas configurações favoreceram a geração de *embeddings* semanticamente mais coesos para o conjunto de portarias analisadas e suas respectivas características.

Entretanto, após a avaliação intrínseca e verificação empírica das similaridades propostas pelo modelo *Doc2Vec*, constatou-se que portarias semanticamente distintas foram consideradas como as mais similares à portaria de número 1963, revelando resultados menos coerentes em comparação aos demais modelos.

A seguir, tem-se a Tabela 4 a qual exhibe as dez portarias mais semelhantes à portaria 1963, conforme identificado pelos modelos *Word2Vec* e *FastText*, utilizando a métrica de similaridade cosseno e as combinações de hiperparâmetros previamente definidas.

Tabela 4: Top 10 portarias mais semelhantes à portaria de referência segundo Word2Vec e FastText

| Posição | Word2Vec | Similaridade | FastText | Similaridade |
|---------|---------------|--------------|---------------|--------------|
| 1 | Portaria 489 | 0,9995 | Portaria 489 | 0,9994 |
| 2 | Portaria 1598 | 0,9991 | Portaria 1598 | 0,9990 |
| 3 | Portaria 1951 | 0,9982 | Portaria 1951 | 0,9974 |
| 4 | Portaria 1472 | 0,9964 | Portaria 1472 | 0,9967 |
| 5 | Portaria 1597 | 0,9958 | Portaria 1766 | 0,9966 |
| 6 | Portaria 1766 | 0,9956 | Portaria 1597 | 0,9962 |
| 7 | Portaria 1033 | 0,9943 | Portaria 1033 | 0,9956 |
| 8 | Portaria 1709 | 0,9941 | Portaria 1709 | 0,9948 |
| 9 | Portaria 1524 | 0,9917 | Portaria 1524 | 0,9924 |
| 10 | Portaria 1767 | 0,9893 | Portaria 1407 | 0,9907 |

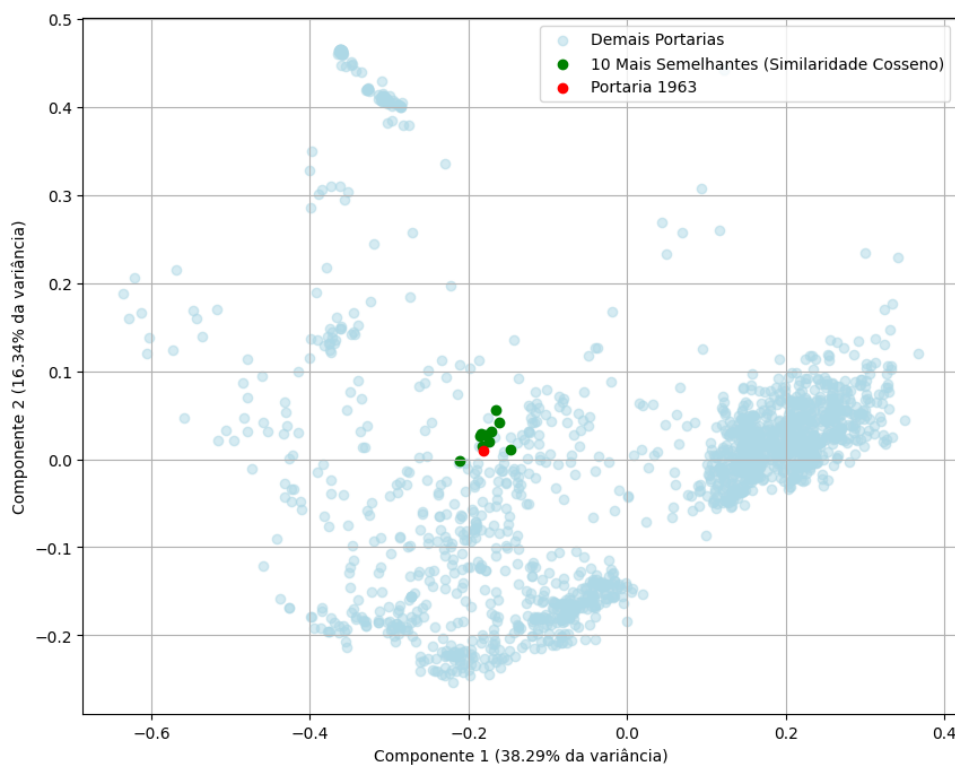
Conforme apresentado na Tabela 4, observa-se uma forte concordância entre os dois modelos, com todas as portarias coincidindo nas duas listas, embora ocupem posições ligeiramente distintas, com exceção das últimas posições nas quais há divergência quanto às portarias mais similares identificadas.

Diante desse cenário, a verificação intrínseca da coerência dos resultados de similaridade representa um primeiro indicativo de que ambos os modelos foram capazes de representar numericamente, de maneira adequada, os dados textuais das portarias.

4.3 Visualização em Dimensionalidade Reduzida dos Embeddings Gerados

As Figuras 12 e 13 apresentam a projeção bidimensional obtida pela Análise de Componentes Principais aplicada aos vetores gerados pelos modelos *Word2Vec* e *FastText*, respectivamente. O propósito é avaliar a distribuição das portarias em um espaço vetorial reduzido e, assim, verificar, de forma visual, a veracidade desta representação com base na posição relativa da portaria 1963 em relação às suas dez portarias mais semelhantes definidas através da similaridade cosseno.

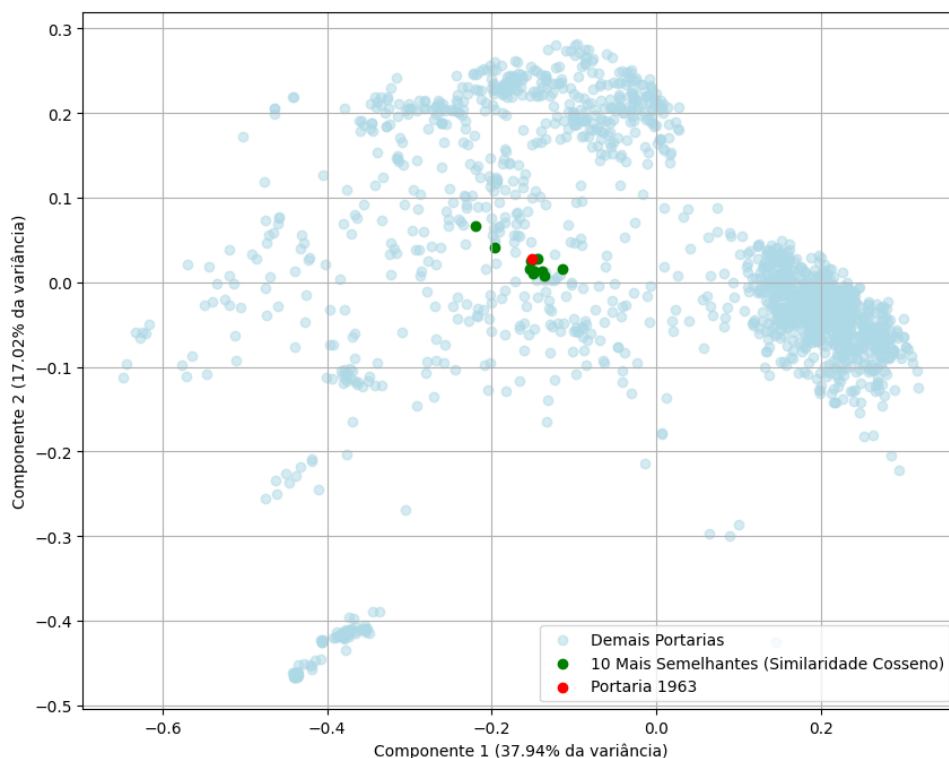
Figura 12: PCA - Word2Vec



Fonte: Elaboração própria.

Desprende-se da Figura 12 que os dois primeiros componentes principais explicam 54,63% da variância total (componente 1: 38,29% e componente 2: 16,34%). Observa-se, também, que a portaria 1963 (destacada em vermelho) e suas dez mais similares (em verde) foram representadas próximas uma das outras, indicando boa preservação da estrutura local no espaço reduzido, apesar do valor não tão expressivo da variância total explicada.

Figura 13: PCA - FastText



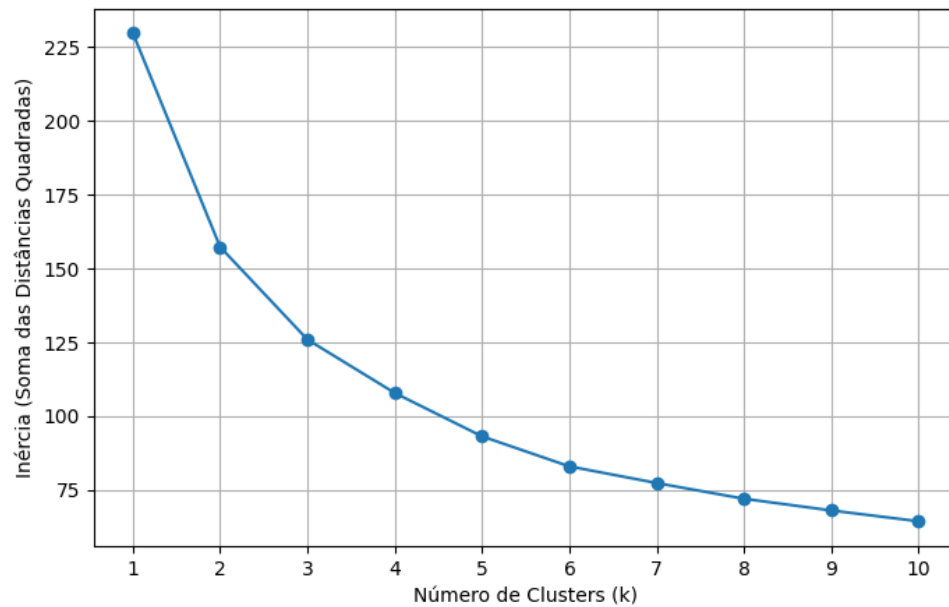
Fonte: Elaboração própria.

Já por meio da Figura 13, verifica-se uma variância total explicada semelhante, porém ligeiramente superior (componente 1: 37,94% e componente 2: 17,02%), totalizando 54,96%. Embora as representações dos documentos mais similares também se posicionem próximas da portaria 1963, observa-se uma leve dispersão maior entre elas, o que sugere uma menor capacidade da técnica de redução de dimensionalidade, quando aplicada aos vetores *FastText*, em preservar a integridade semântica das portarias mais similares, em comparação à aplicação da mesma técnica sobre os vetores *Word2Vec*.

4.4 Agrupamento das Portarias

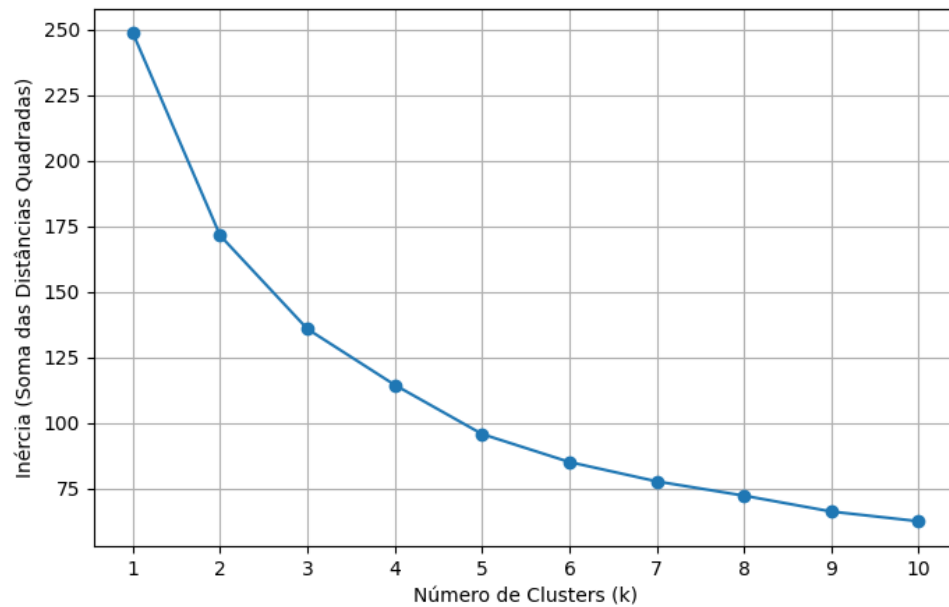
Nesta seção, são apresentados os resultados do agrupamento das portarias por meio do algoritmo *K-Means*, com o objetivo de identificar grupos semanticamente semelhantes. Diante disso, nas Figuras 14 e 15 são representadas as curvas para definição do número ótimo de clusters.

Figura 14: Determinação do número ótimo de clusters - Word2Vec



Fonte: Elaboração própria.

Figura 15: Determinação do número ótimo de clusters – FastText



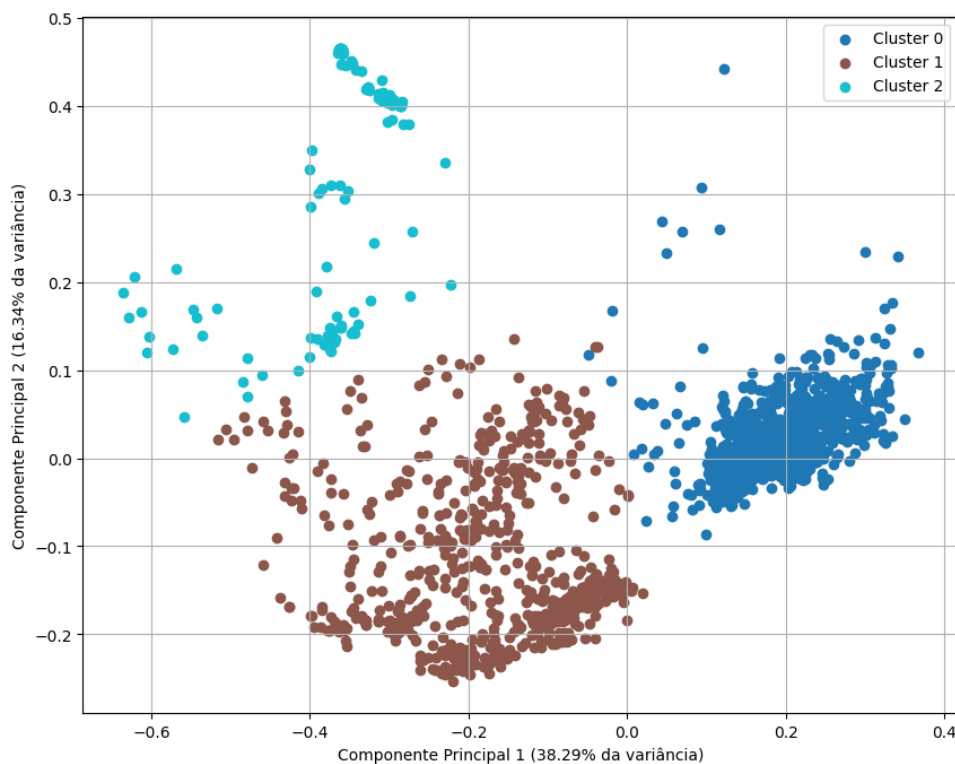
Fonte: Elaboração própria.

Em ambas as curvas, que apresentam comportamentos muito semelhantes, observa-se um declínio acentuado da curva entre $k = 1$ e $k = 3$. Além do mais, a partir do ponto $k = 4$, a curva se torna mais suave, indicando melhora pouco significativa na compactação dos grupos. Dessa forma, foram testados valores entre $k = 2$ e $k = 4$ e constatou-se que a

configuração com três agrupamentos apresentou uma separação visualmente mais coerente entre os clusters, reforçando a escolha de $k = 3$ como o número ótimo de agrupamentos.

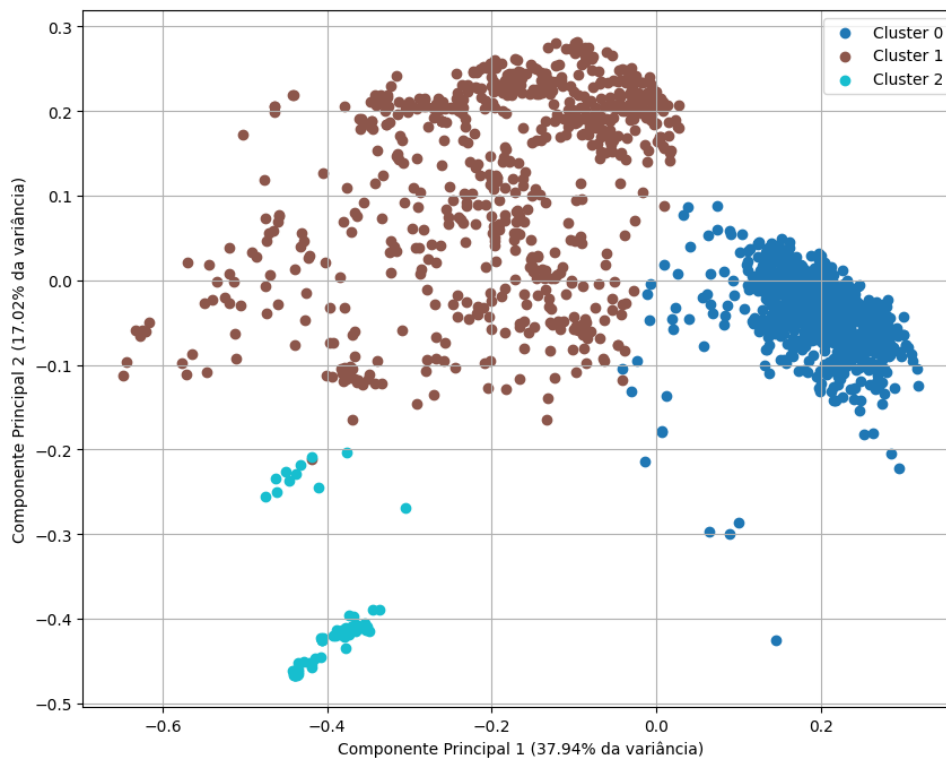
A partir da definição do número ótimo de agrupamentos, foi implementado o algoritmo de agrupamento *K-Means* com $k = 3$, aplicado às representações vetoriais *Word2Vec* e *FastText* das portarias. Nessa perspectiva, os gráficos presentes nas Figuras 16 e 17 apresentam as disposições espaciais das portarias agrupadas, evidenciando os três clusters gerados.

Figura 16: Clusterização das portarias - Word2Vec



Fonte: Elaboração própria.

Figura 17: Clusterização das portarias - FastText



Fonte: Elaboração própria.

De acordo com as Figuras 16 e 17, observa-se, de modo geral, que os agrupamentos produzidos pelo algoritmo *K-means* a partir dos vetores *Word2Vec* e *FastText* são bastante semelhantes, com clusters visualmente bem definidos e poucas observações sobrepostas. No entanto, ao se realizar uma análise mais minuciosa, nota-se que o Cluster 1, no *K-means* aplicado às representações do *Word2Vec*, apresenta-se de forma mais condensada do que aquele formado com os embeddings do *FastText*. Por outro lado, os Clusters 0 de ambos não aparentam demonstrar diferenças significativas em sua distribuição. Por fim, os Clusters 2, menos representativos, mostram-se visualmente mais densos quando originados a partir dos vetores *FastText*, em comparação com aqueles formados pelo *Word2Vec*.

De forma geral, os agrupamentos gerados por ambas as representações demonstraram desempenho satisfatório na segmentação das portarias. Contudo, optou-se por seguir com os agrupamentos obtidos a partir do *Word2Vec* nas análises subsequentes, em detrimento do *FastText*, devido às diferenças observadas no Cluster 1 e aos valores ligeiramente superiores evidenciados pela Tabela 3 na Seção 4.2.

4.5 Análise Exploratória dos Dados Agrupados

Nesta etapa, são apresentados os resultados da análise exploratória realizada a partir dos dados agrupados, com o objetivo de compreender e avaliar extrinsecamente a coerência dos padrões identificados e, conseqüentemente, a qualidade dos *embeddings* gerados. Diante disso, a Tabela 5, a seguir, apresenta a distribuição da quantidade de portarias por cluster, permitindo uma visão geral sobre a representatividade de cada grupo no conjunto analisado.

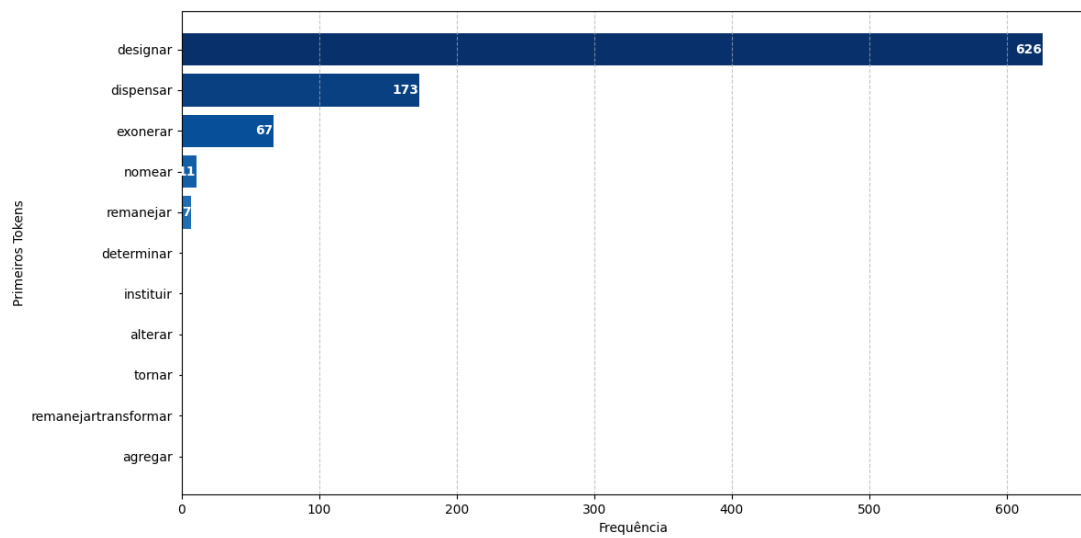
Tabela 5: Distribuição de portarias por cluster

| Cluster | Quantidade | Percentual (%) |
|--------------|-------------|----------------|
| Cluster 0 | 890 | 52,13% |
| Cluster 1 | 660 | 38,66% |
| Cluster 2 | 157 | 9,19% |
| Total | 1707 | 100% |

Com auxílio da Tabela 5 e da Figura 16, observa-se que a maior parte das portarias encontra-se concentrada no Cluster 0, que representa 52,13% do total. O Cluster 1 abrange 38,66% das portarias, enquanto o Cluster 2 reúne apenas 9,19%, sugerindo a presença de um grupo mais específico ou com menor similaridade em relação aos demais.

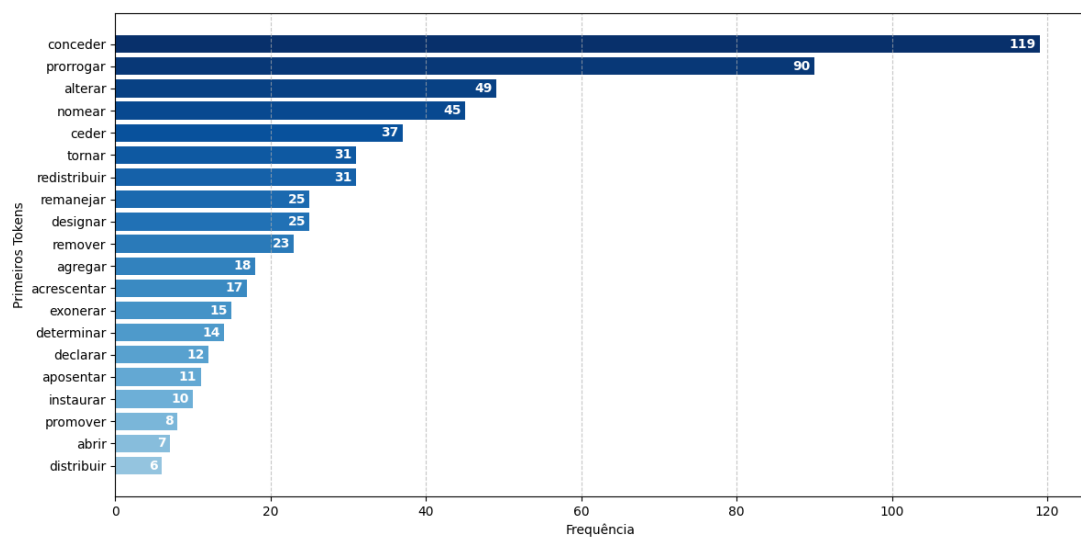
Seguindo a lógica adotada na Seção 4.1, as Figuras 18, 19 e 20 apresentam as distribuições de frequência dos verbos que sucedem diretamente à expressão “RESOLVE:” por agrupamento, com o propósito de evidenciar as ações mais recorrentes nas portarias e, assim, compreender as temáticas predominantes entre os agrupamentos analisados.

Figura 18: Gráfico de barras dos primeiros verbos mais frequentes - Cluster 0



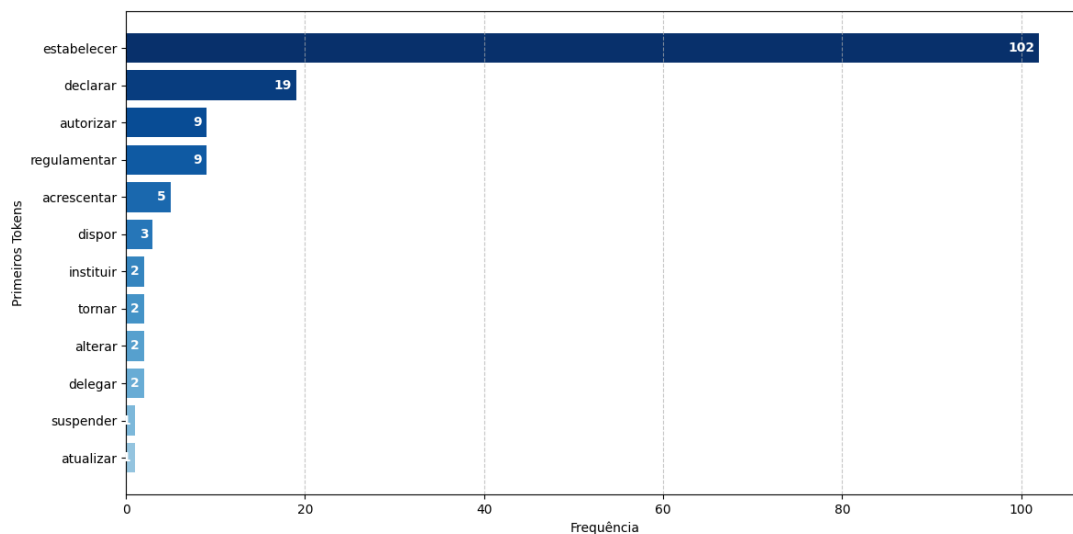
Fonte: Elaboração própria.

Figura 19: Gráfico de barras dos primeiros verbos mais frequentes - Cluster 1



Fonte: Elaboração própria.

Figura 20: Gráfico de barras dos primeiros verbos mais frequentes - Cluster 2



Fonte: Elaboração própria.

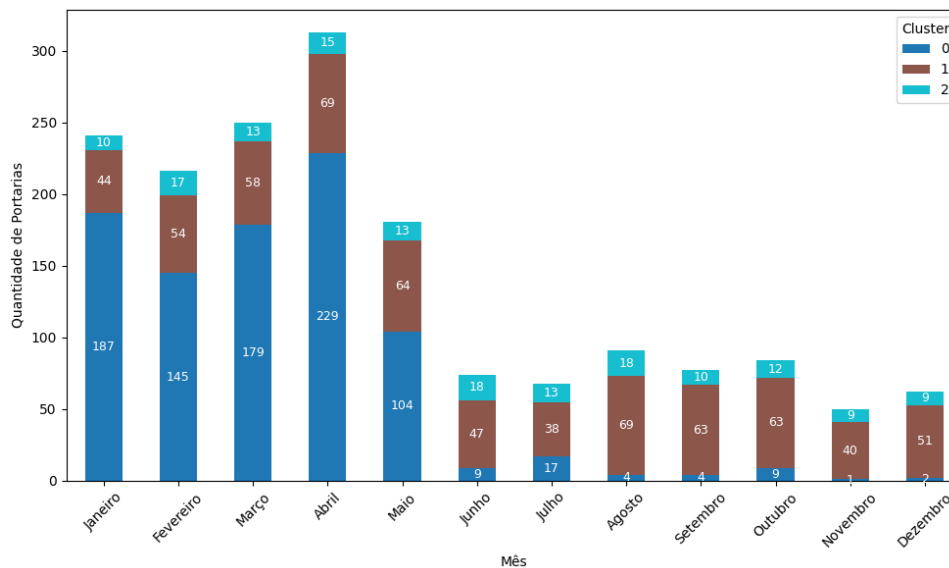
Com base nas Figuras 18,19 e 20, capta-se que o Cluster 0 apresenta forte concentração em torno do verbo designar, com 626 ocorrências, o que representa aproximadamente 70,3% dos casos dentro desse grupo. Em sequência, têm-se o verbo dispensar (173) e exonerar (67).

No que tange ao Cluster 1, observa-se uma distribuição mais equilibrada de verbos, refletindo uma natureza mais geral das portarias deste agrupamento. Os verbos mais frequentes são conceder (199), prorrogar (90) e alterar (49), que representam juntos cerca de 50,2% do total de 660 portarias deste cluster. Além disso, o Cluster 1 também compartilha alguns verbos com o Cluster 0, como designar (25) e exonerar (15), indicando certa sobreposição temática, embora em menor proporção.

Já o Cluster 2 apresenta como seu principal ato o de estabelecer, com 102 ocorrências, o que representa 65% das 157 portarias agrupadas nesse cluster. Os demais verbos mais frequentes, como declarar (19) e autorizar (9), aparecem com frequência consideravelmente menor.

Com o objetivo de investigar padrões temporais na publicação das portarias e verificar possíveis concentrações temáticas ao longo do tempo, foi construída a visualização da Figura 21 que apresenta a quantidade de portarias publicadas por mês e pelos clusters identificados.

Figura 21: Quantidade de portarias publicadas por mês e por cluster



Fonte: Elaboração própria.

A análise conjunta das Figuras 9, 10, 11, 18, 19, 20 e 21 permite identificar que o Cluster 0, majoritariamente referente a designações em especial de bacharéis de direito, concentra a maior parte das portarias mensais no início do ano, especialmente entre janeiro e maio. Em contraste, os Clusters 1, em sua maioria relacionados a concessões de pensão civil, prorrogações de cessão de servidores e alterações, e 2, predominantemente voltado a estabelecer escalas de plantão, apresentam uma distribuição mais estável ao longo do ano.

Perante o exposto, esta seção cumpre com seus objetivos, em especial, ao de realizar a validação extrínseca proposta, funcionando, assim, como uma segunda verificação da coerência na definição do modelo gerador de *embeddings*.

4.6 Reconhecimento de Entidades Nomeadas

Nesta seção, apresentam-se características referentes à proporcionalidade das entidades anotadas no conjunto de portarias anotadas, bem como os desempenhos do modelo de Reconhecimento de Entidades Nomeadas (REN) em diferentes avaliações.

4.6.1 Características do Conjunto de Portarias Anotadas

Quanto às anotações manuais realizadas a partir das 735 portarias selecionadas via amostragem estratificada, a Tabela 6 apresenta a frequência absoluta e relativa de cada uma das categorias de entidades anotadas.

Tabela 6: Quantidade de entidades anotadas

| Entidade | Quantidade | Percentual (%) |
|--------------|------------|----------------|
| ACAO | 1107 | 55,60 |
| SUJEITO | 522 | 26,22 |
| LOCAL | 220 | 11,05 |
| DATA | 142 | 7,13 |
| Total | 1991 | 100,00 |

Com auxílio da Tabela 6, observa-se uma predominância expressiva da entidade ACAO, que representa 55,60% do total das entidades, seguida pelas entidades SUJEITO (26,22%), LOCAL (11,05%) e DATA (7,13%). Assim, esta distribuição evidencia uma considerável desproporção na frequência das entidades anotadas.

De forma semelhante, a Tabela 7 apresenta a distribuição percentual das entidades anotadas por clusters, permitindo observar quais tipos de entidades predominam em cada agrupamento.

Tabela 7: Distribuição das entidades anotadas por cluster

| Entidade | Cluster 0 (%) | Cluster 1 (%) | Cluster 2 (%) |
|--------------|---------------|---------------|---------------|
| ACAO | 682 (65,58) | 317 (50,16) | 108 (33,86) |
| SUJEITO | 349 (33,56) | 164 (25,95) | 9 (2,82) |
| LOCAL | 9 (0,87) | 99 (15,66) | 112 (35,11) |
| DATA | 0 (0,00) | 52 (8,23) | 90 (28,21) |
| Total | 1040 (100,00) | 632 (100,00) | 319 (100,00) |

Por meio da Tabela 7, percebe-se uma variação significativa na composição de entidades em cada agrupamento. O Cluster 0, é majoritariamente composto pelas entidades ACAO (65,58%) e SUJEITO (33,56%), com pequena representação das entidades LOCAL e DATA. Por sua vez, o Cluster 1 apresenta uma distribuição mais balanceada entre as quatro entidades, mas, ainda assim, com predominância das entidades ACAO (50,16%) e SUJEITO (25,95%) seguida de uma parcela mais expressiva das entidades LOCAL (15,66%) e DATA (8,23%) em comparação ao Cluster 0. Em contrapartida, o Cluster 2, em relação aos demais, apresenta uma composição onde as entidades LOCAL (35,11%) e DATA (28,21%) estão com maior representatividade.

Nesse sentido, considerando o desbalanceamento das entidades percebido tanto no total do conjunto de portarias anotadas quanto na divisão por clusters, o modelo de reconhecimento de entidades nomeadas foi avaliado utilizando validação cruzada simples e validação cruzada estratificada. Com isso, o objetivo foi comparar o impacto dessas

diferentes abordagens nas métricas de avaliação.

4.6.2 Avaliação do Modelo com Validação Cruzada Estratificada

Na presente seção, as Tabelas 8 e 9 apresentam o resultado médio do modelo de REN ao longo das partições (*folds*) da validação cruzada. Por conseguinte, a Tabela 8 evidencia as médias das métricas *macro F1-score* e *micro F1-score*. De maneira similar, a Tabela 9 apresenta a média dos valores de *F1-score* obtidos para cada classe de entidade.

Tabela 8: Desempenho médio do modelo - Validação cruzada estratificada

| Média | F1-score |
|-------|---------------------|
| Macro | 0,7973 \pm 0,0471 |
| Micro | 0,8789 \pm 0,0205 |

Tabela 9: F1-score médio por entidades - Validação cruzada estratificada

| Entidade | F1-score |
|----------|---------------------|
| DATA | 0,7504 \pm 0,1172 |
| ACAO | 0,8698 \pm 0,0334 |
| LOCAL | 0,6474 \pm 0,0559 |
| SUJEITO | 0,9217 \pm 0,0155 |

Os resultados apresentados na Tabela 8 indicam que, no geral, o modelo avaliado pela validação cruzada estratificada apresenta bom desempenho. Entretanto, a média *macro F1-score*, que atribui peso igual a todas as categorias, indica queda de desempenho nas categorias de entidades menos frequentes, assim como é observado pelas métricas das entidades DATA e LOCAL na Tabela 9. Desse modo, a diferença observada em relação às métricas médias *micro F1-score* e *macro F1-score* sugere um leve viés em favor das entidades AÇÃO e SUJEITO majoritárias.

4.6.3 Avaliação do Modelo com Validação Cruzada Simples

Analogamente à Seção 4.6.2, aqui são apresentadas as Tabelas 10 e 11, similares às Tabelas 8 e 9, respectivamente, porém referentes aos resultados de desempenho do modelo de REN avaliado por meio de validação cruzada simples.

Tabela 10: Desempenho médio do modelo - Validação cruzada

| Média | F1-score |
|-------|---------------------|
| Macro | $0,8036 \pm 0,0365$ |
| Micro | $0,8766 \pm 0,0180$ |

Tabela 11: F1-score médio por entidades - Validação cruzada

| Entidade | F1-score |
|----------|---------------------|
| DATA | $0,7351 \pm 0,0939$ |
| ACAO | $0,8691 \pm 0,0296$ |
| LOCAL | $0,6972 \pm 0,0465$ |
| SUJEITO | $0,9129 \pm 0,0128$ |

Nota-se que os resultados das Tabelas 10 e 11 foram muito similares aos resultados das Tabelas 8 e 9. Entretanto, observa-se uma leve queda no F1-score médio da entidade DATA e um aumento no desempenho da entidade LOCAL, enquanto as entidades ACAO e SUJEITO apresentaram resultados praticamente equivalentes em ambas as validações. Em relação às médias das métricas, *macro F1-score* teve um pequeno aumento, em contraponto a *micro F1-score* que apresentou uma ligeira redução.

4.6.4 Avaliação do Modelo com Validação Cruzada Simples após sobreamostragem

Visto que os resultados anteriores das Seções 4.6.2 e 4.6.3 evidenciaram desempenhos do modelo de REN inferiores nas entidades LOCAL e DATA em comparação com ACAO e SUJEITO, bem como valores de *macro F1-score* menores que os de *micro F1-score*, esta seção apresenta os resultados do modelo avaliado por validação cruzada após a aplicação de sobreamostragem. Nesse processo, as portarias que continham anotações dessas duas entidades foram duplicadas, com o objetivo de reduzir o desequilíbrio observado. Diante disso, a apresentação dos resultados segue o mesmo padrão adotado nas tabelas das Seções 4.6.2 e 4.6.3.

No que tange ao procedimento de sobreamostragem, foram identificadas 229 portarias anotadas contendo ao menos uma entidade do tipo LOCAL ou DATA. Em vista disso, antes da sobreamostragem, o conjunto anotado contava com 735 documentos anotados e, ao final, passou a totalizar 964 portarias, conforme detalhado na Tabela 12, que apresenta a nova distribuição das entidades após a aplicação da técnica.

Tabela 12: Quantidade de entidades anotadas após sobreamostragem

| Entidade | Quantidade | Percentual (%) |
|--------------|------------|----------------|
| ACAO | 1371 | 51,34 |
| SUJEITO | 576 | 21,56 |
| LOCAL | 440 | 16,47 |
| DATA | 284 | 10,63 |
| Total | 2671 | 100,00 |

A visualização conjunta das Tabelas 6 e 12 faz com que seja perceptível que, após a sobreamostragem, o total de entidades anotadas aumentou de 1991 para 2671, elevando a participação de LOCAL de 11,05% para 16,47% e de DATA de 7,13% para 10,63%. Desse modo, percebe-se um desbalanceamento entre as classes em menor grau do que o observado antes da aplicação da técnica.

Tabela 13: Desempenho médio do modelo - Após sobreamostragem - Validação cruzada

| Média | F1-score |
|-------|---------------------|
| Macro | 0,8957 \pm 0,0200 |
| Micro | 0,9146 \pm 0,0178 |

Tabela 14: F1-score médio por entidades

| Entidade | F1-score |
|----------|---------------------|
| DATA | 0,9376 \pm 0,0364 |
| ACAO | 0,9053 \pm 0,0165 |
| LOCAL | 0,7968 \pm 0,0554 |
| SUJEITO | 0,9432 \pm 0,0211 |

Com fundamentação nas Tabelas 13 e 14 em comparação com as das Seções 4.6.2 e 4.6.3, nota-se, após a aplicação da técnica de sobreamostragem, uma melhora substancial no desempenho do modelo. Portanto, o *macro F1-score* aumentou, em média, de 0,8000 para 0,8957. Já o *micro F1-score*, que antes apresentava média de 0,8777, atingiu 0,9146, representando um ganho expressivo em comparação às validações anteriores.

No que diz respeito às entidades, o destaque vai para DATA, que apresentou uma nítida evolução de *F1-scores* médios inferiores a 0,76 para 0,9376. Similarmente, a entidade LOCAL também apresentou ganho considerável, alcançando um *F1-score* médio de 0,7968, superando os resultados anteriores e demonstrando maior robustez do modelo para essa classe. Por fim, as entidades ACAO e SUJEITO, que já apresentavam desempenho elevado, também registraram avanços, atingindo, respectivamente, 0,9053 e 0,9432.

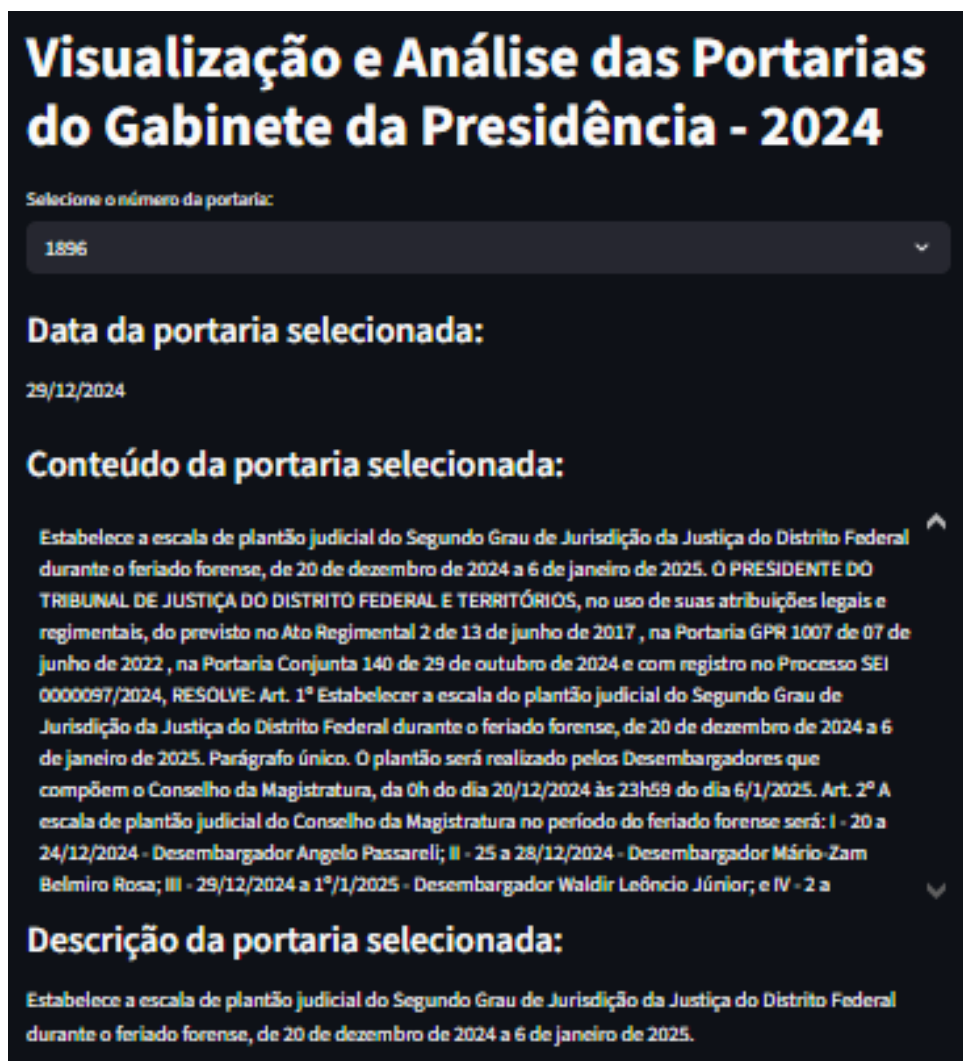
Diante do exposto, a abordagem de balanceamento mostrou-se eficaz na melhora do desempenho do modelo em relação às classes minoritárias. Ademais, os resultados consistentes do modelo de REN obtidos com o uso de *embeddings* configuram validação extrínseca, funcionando como uma terceira verificação da coerência na definição do modelo gerador de *embeddings*.

4.7 Aplicativo Web

Como última parte da Seção 4, o enfoque passa a ser a apresentação da aplicação web interativa proposta, a qual permite ao usuário explorar de forma dinâmica os resultados desenvolvidos ao longo deste trabalho.

Assim sendo, conforme ilustrado na Figura 22, o ponto de partida consiste na escolha do número de uma portaria específica, a partir do qual são exibidas informações relevantes, como a data de publicação, o conteúdo textual completo e sua descrição. Desse modo, o usuário é capaz de explorar qualquer uma das portarias do Gabinete da Presidência do TJDF do ano de 2024.

Figura 22: Interface do aplicativo - Conteúdos



Visualização e Análise das Portarias do Gabinete da Presidência - 2024

Selecione o número da portaria:

1896

Data da portaria selecionada:

29/12/2024

Conteúdo da portaria selecionada:

Estabelece a escala de plantão judicial do Segundo Grau de Jurisdição da Justiça do Distrito Federal durante o feriado forense, de 20 de dezembro de 2024 a 6 de janeiro de 2025. O PRESIDENTE DO TRIBUNAL DE JUSTIÇA DO DISTRITO FEDERAL E TERRITÓRIOS, no uso de suas atribuições legais e regimentais, do previsto no Ato Regimental 2 de 13 de junho de 2017, na Portaria GPR 1007 de 07 de junho de 2022, na Portaria Conjunta 140 de 29 de outubro de 2024 e com registro no Processo SEI 0000097/2024, RESOLVE: Art. 1º Estabelecer a escala do plantão judicial do Segundo Grau de Jurisdição da Justiça do Distrito Federal durante o feriado forense, de 20 de dezembro de 2024 a 6 de janeiro de 2025. Parágrafo único. O plantão será realizado pelos Desembargadores que compõem o Conselho da Magistratura, da 0h do dia 20/12/2024 às 23h59 do dia 6/1/2025. Art. 2º A escala de plantão judicial do Conselho da Magistratura no período do feriado forense será: I - 20 a 24/12/2024 - Desembargador Angelo Passarelli; II - 25 a 28/12/2024 - Desembargador Mário Zam Belmiro Rosa; III - 29/12/2024 a 1º/1/2025 - Desembargador Waldir Leôncio Júnior; e IV - 2 a

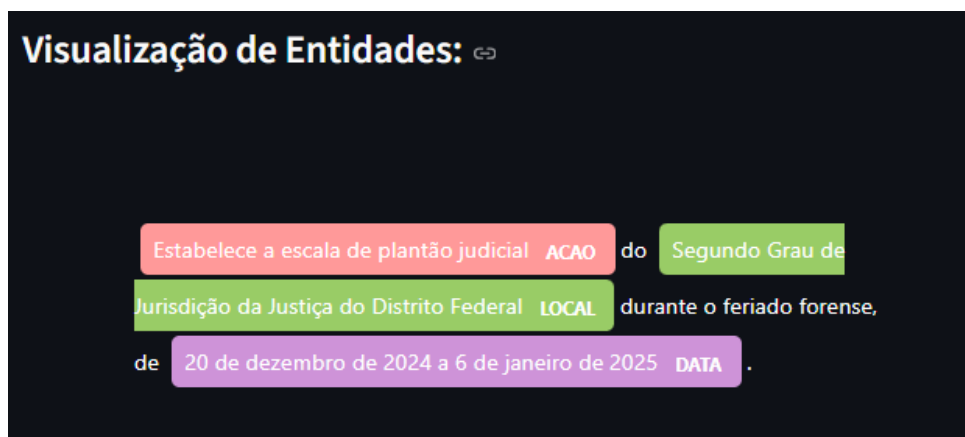
Descrição da portaria selecionada:

Estabelece a escala de plantão judicial do Segundo Grau de Jurisdição da Justiça do Distrito Federal durante o feriado forense, de 20 de dezembro de 2024 a 6 de janeiro de 2025.

Fonte: Elaboração própria.

Em sequência, ao descer a página, é possibilitada ao visitante a visualização das entidades previstas pelo modelo de REN, o que lhe oferece uma visualização estruturada dos principais elementos informacionais contidos na descrição da portaria analisada, como ilustrado na Figura 23.

Figura 23: Interface do aplicativo - Entidades



Fonte: Elaboração própria.

Na etapa seguinte, ilustrada na Figura 24, são listadas as cinco portarias mais semelhantes à portaria selecionada, com base nos cálculos de similaridade do cosseno aplicados aos vetores gerados pelo modelo *Word2Vec*. No site, entretanto, essa funcionalidade disponibiliza as dez portarias mais similares. Optou-se por apresentar apenas cinco na imagem em razão da perda de nitidez causada pelo excesso de informações.

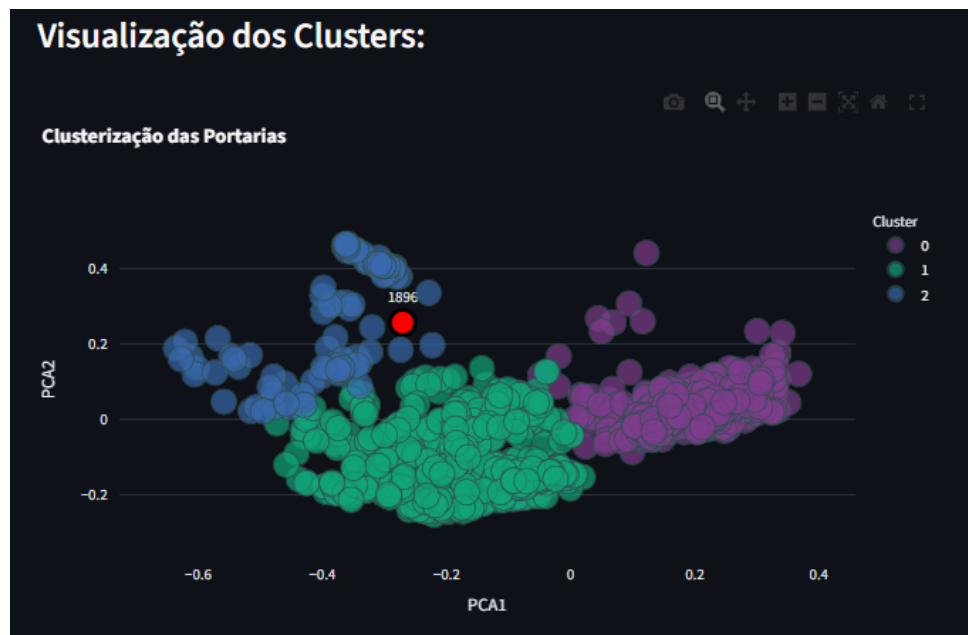
Figura 24: Interface do aplicativo - Similaridade



Fonte: Elaboração própria.

Posteriormente, conforme representado na Figura 25, tem-se a visualização dos agrupamentos gerados pelo algoritmo *K*-Means, combinados com a redução de dimensionalidade via PCA. Além disso, vale ressaltar que o gráfico é interativo, possibilitando a identificação do número da portaria e do cluster ao qual ela pertence ao posicionar o cursor sobre os pontos.

Figura 25: Interface do aplicativo - Agrupamentos



Fonte: Elaboração própria.

Complementarmente, a aplicação disponibiliza ainda visualizações em formato de nuvem de palavras para cada um dos clusters identificados. Além disso, são estampados gráficos análogos aos ilustrados nas Figuras 18, 19 e 20, que destacam as principais ações mais recorrentes em cada agrupamento. Por fim, é apresentada uma visualização da distribuição temporal das portarias agrupadas, conforme demonstrado na Figura 21.

5 Conclusão

Este trabalho dedicou-se à aplicação de técnicas de Processamento de Linguagem Natural (PLN), aprendizado de máquina e aprendizado profundo na análise de portarias emitidas pelo Gabinete da Presidência do Tribunal de Justiça do Distrito Federal e dos Territórios (TJDFT) em 2024. Nesse sentido, as estratégias propostas contemplaram diferentes aspectos do conteúdo textual das portarias, com foco nas tarefas de vetorização numérica, cálculo de similaridade semântica, identificação de agrupamentos e Reconhecimento de Entidades Nomeadas, além do desenvolvimento de uma aplicação web interativa que sistematiza e facilita a visualização dos principais resultados.

No que tange à análise exploratória do conteúdo das portarias (Seções 4.1 e 4.5, foi possível obter uma visão abrangente da distribuição e composição dos documentos publicados em 2024. Em sequência, a comparação dos métodos *Word2Vec*, *Doc2Vec* e *FastText* ressalta a relevância da escolha adequada dos hiperparâmetros para o desempenho dos modelos de vetorização textual (Seção 4.2). Dessa forma, a escolha otimizada dos parâmetros por meio de *grid search* convergiu para configurações semelhantes, com destaque para a arquitetura *Skip-gram* (Tabela 3), que apresentou resultados consistentes na avaliação intrínseca da similaridade entre portarias. Em adição, a similaridade cosseno mostrou-se eficaz para capturar relações semânticas entre os textos, conforme evidenciado pelos documentos mais similares identificados (Tabela 4).

Ademais, a visualização dos *embeddings* em espaço reduzido (Seção 4.3), embora a variância explicada total não tenha sido tão expressiva por meio dos dois componentes principais mais representativos, demonstrou que a aplicação do PCA foi eficaz ao oferecer uma representação visual ainda coerente das relações de similaridade esperadas entre as portarias (Figuras 12 e 13). Dessa forma, observou-se, ainda, um leve destaque para os *embeddings* gerados pelo *Word2Vec* (Figura 12), que manteve as dez portarias mais similares à referenciada mais próximas entre si quando comparados aos produzidos pelo *FastText* (Figura 13).

Adicionalmente, os agrupamentos gerados a partir dos vetores do *Word2Vec* com o algoritmo *K-means* ($k = 3$) (Seção 4.4) apresentaram uma segmentação mais consistente, especialmente no Cluster 1, que se mostrou mais condensado em comparação ao correspondente obtido com o *FastText*. Essa diferença, em conjunto com os resultados ligeiramente superiores observados nas medidas de coesão (Tabela 3), justificou a escolha do *Word2Vec* para conduzir as análises posteriores. Dessa forma, cabe destacar que a coerência dos agrupamentos identificados e as classificações realizadas pelo modelo de REN corroboram como validações extrínsecas dos *embeddings* produzidos.

Por fim, no que se refere ao Reconhecimento de Entidades Nomeadas (Seção

4.6), os resultados obtidos demonstraram que, apesar do desbalanceamento nas classes anotadas, o modelo BiLSTM foi capaz de identificar com precisão as entidades, sobretudo após a aplicação do processo de sobreamostragem. Ainda assim, ressalta-se que tanto o processo de anotação quanto a aplicação da sobreamostragem devem ser realizados com cuidado e rigor, uma vez que impactam diretamente a qualidade do treinamento e podem introduzir vieses nos resultados. Como resultado, o modelo final apresentou desempenho satisfatório, com *macro F1-score* médio em torno de 0,8957 evidenciando seu potencial para tarefas de REN em domínios jurídicos.

Diante do exposto, as principais contribuições deste trabalho residem na análise detalhada das portarias, possibilitada pela aplicação das técnicas de Processamento de Linguagem Natural em conjunto com as de aprendizado de máquina e aprendizado profundo. Essa abordagem permitiu identificar padrões temporais e semânticos relevantes, além de oferecer uma segmentação consistente dos documentos, contribuindo para uma melhor compreensão e organização das portarias analisadas.

Como proposta para trabalhos futuros, sugere-se a ampliação deste estudo com a incorporação da técnica de agrupamento *DBSCAN* (*Density-Based Spatial Clustering of Applications with Noise*), que pode gerar agrupamentos mais concisos, excluindo portarias muito diferentes ("ruidosas") das demais. Além disso, a implementação de modelos baseados em arquiteturas de *Transformers*, como *BERT*, *RoBERTa* conjuntamente com transferência de aprendizado (*transfer learning*) possui potencial para aprimorar a qualidade da representação semântica dos textos e Reconhecimento de Entidades Nomeadas (BARROS et al., 2024).

6 Referências

- BARROS, F. M. d. C. et al. Processamento de linguagem natural como ferramenta de suporte em documentos jurídicos: uma revisão sistemática. *Revista de Casos e Consultoria*, v. 15, n. 1, p. e36701, ago. 2024. Disponível em: <https://periodicos.ufrn.br/casoseconsultoria/article/view/36701>.
- BOLFARINE, H.; BUSSAB, W. O. *Elementos de Amostragem*. São Paulo: Instituto de Matemática e Estatística, Universidade de São Paulo, 2004.
- CASELI, H. d. M.; NUNES, M. d. G. V. *Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português – 2ª Edição*. [S.l.]: BPLN, São Carlos, 2024. Disponível em: <https://brasileiraspln.com/livro-pln/2a-edicao/>. ISBN 978-65-00-95750-1.
- CASTRO, P. V. Q. *Aprendizagem profunda para reconhecimento de entidades nomeadas em domínio jurídico*. Dissertação (Dissertação (Mestrado em Ciência da Computação)) — Universidade Federal de Goiás, Goiânia, 2019. 125 f. Disponível em: <http://repositorio.bc.ufg.br/tede/handle/tede/10276>.
- COPATTI, B. S. *Pré-processamento de dados de texto de requisitos de software utilizando técnicas de aprendizado de máquina*. 2022. Monografia (Especialização em Ciências de Dados) - Universidade Tecnológica Federal do Paraná, Dois Vizinhos. Disponível em: <https://repositorio.utfpr.edu.br/jspui/bitstream/1/31718/3/processamentotextoaprendizadomaquina.pdf>.
- CORDEIRO, T. V. B. *Predição de Default de Empresas: Técnicas de Machine Learning em Dados Desbalanceados*. Dissertação (Dissertação de Mestrado) — Fundação Getúlio Vargas, Escola de Economia de São Paulo, São Paulo, SP, 2020. Disponível em: <https://repositorio.fgv.br/server/api/core/bitstreams/2b4515b8-f04f-4e5c-8e21-debdd7c18e99/content>.
- COSTA, R. P. *Reconhecimento de entidades nomeadas em textos informais no domínio legislativo*. Dissertação (Dissertação (Mestrado em Ciência da Computação)) — Universidade Federal de Goiás, Goiânia, 2023. 70 f. Disponível em: https://files.cercomp.ufg.br/weby/up/1289/o/DISSERTACAO_Final_corrigida.pdf.
- FREITAS, L. J. G. *Clusterização de textos aplicada ao tratamento de dados jurídicos desbalanceados*. Dissertação (Dissertação (Mestrado em Estatística)) — Universidade de Brasília, Departamento de Estatística, Brasília, 2023. Disponível em: <http://repositorio.unb.br/handle/10482/48841>.
- GARCIA, G. C. *Reconhecimento de Entidades Nomeadas na base de notificações de eventos adversos e queixas técnicas de dispositivos médicos no Brasil*. Dissertação (Dissertação (Mestrado Profissional em Computação Aplicada)) — Universidade de Brasília, Brasília, ago 2021. Data de defesa: 31 de agosto de 2021. Disponível em: <http://repositorio.unb.br/handle/10482/42718>.

GERG, F.; SCHMIDHUBER, J. Recurrent nets that time and count. In: . [S.l.: s.n.], 2000. v. 3, p. 189 – 194 vol.3. ISBN 0-7695-0619-4.

GOCHHAIT, D. S. Comparative analysis of machine and deep learning techniques for text classification with emphasis on data preprocessing. *Qeios*, 05 2024.

GRUS, J. *Data Science do zero: Primeiras regras com o Python*. Rio de Janeiro: Alta Books, 2016. ISBN 978-85-508-0387-6.

GÉRON, A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2. ed. Sebastopol, CA: O'Reilly Media, Inc., 2019. Disponível em: <https://www.rasa-ai.com/wp-content/uploads/2022/02/Aurlien-Gron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-OReilly-Media-2019.pdf>. Acesso em: 14 jun. 2025. ISBN 978-1-492-03264-9.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. ed. New York: Springer, 2009. (Springer Series in Statistics). Disponível em: <https://hastie.su.domains/ElemStatLearn/>. Acesso em: 14 jun. 2025. ISBN 978-0-387-84858-7.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, v. 9, p. 1735–80, 12 1997.

I., D.; OBUNADIKE, G. Analysis and visualization of market segmentation in banking sector using kmeans machine learning algorithm. *FUDMA JOURNAL OF SCIENCES*, v. 6, n. 1, p. 387 – 393, Apr. 2022. Disponível em: <https://fjs.fudutsinma.edu.ng/index.php/fjs/article/view/910>.

JOULIN, A. et al. *FastText.zip: Compressing text classification models*. 2016. Disponível em: <https://arxiv.org/abs/1612.03651>.

LE, Q. V.; MIKOLOV, T. *Distributed Representations of Sentences and Documents*. 2014. Disponível em: <https://arxiv.org/abs/1405.4053>.

LOCA, A. L. da S. *Uma metodologia experimental para avaliar abordagens de aprendizado de máquina para diagnóstico de falhas com base em sinais de vibração*. Dissertação (Dissertação de Mestrado) — Universidade Federal do Espírito Santo, Vitória, ES, 2023. Disponível em: https://sappg.ufes.br/tese_drupal/tese_14522_Disserta%E7%E3o_Antonio_Loca.pdf.

MAGALHÃES, L. H. de. *Agrupamento Automático de Notícias de Jornais Online Usando Técnicas de Machine Learning para Clustering de Textos no Idioma Português*. Dissertação (Dissertação de Mestrado) — Universidade Federal de Minas Gerais, Escola de Ciência da Informação, Belo Horizonte, MG, 2020. Disponível em: <http://hdl.handle.net/1843/37525>.

MIKOLOV, T. et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. Disponível em: <https://arxiv.org/abs/1301.3781>.

MORAIS, J. T. G. *Análise de Componentes Principais Integrada a Redes Neurais Artificiais para Predição de Matéria Orgânica*. Dissertação (Dissertação de Mestrado) — Universidade Federal da Bahia, Salvador, 2011. Disponível em: <https://repositorio.ufba.br/bitstream/ri/18678/1/Dissertacao%20Tabita.pdf>.

NASCIMENTO, I. V. L. do. *DeepREF: um Framework para Classificação de Relações Baseado em Deep Learning*. Dissertação (Dissertação de Mestrado) — Universidade Federal Rural de Pernambuco, Departamento de Estatística e Informática, Programa de Pós-Graduação em Informática Aplicada, Recife, PE, 2022. Disponível em: <https://www.ppgia.ufrpe.br/sites/default/files/testes-dissertacoes/DEEPREF%20UM%20FRAMEWORK%20PARA%20CLASSIFICA%C3%87%C3%83O%20DE%20RELA%C3%87%C3%95ES%20BASEADO%20EM%20DEEP%20LEARNING.pdf>.

OLIVEIRA, R.; NASCIMENTO, E. G. S. Clustering by similarity of brazilian legal documents using natural language processing approaches. In: _____. [s.n.], 2021. ISBN 978-1-83969-887-3. Disponível em: https://www.researchgate.net/publication/354579623_Clustering_by_Similarity_of_Brazilian_Legal_Documents_Using_Natural_Language_Processing_Approaches.

SCHNABEL, T. et al. Evaluation methods for unsupervised word embeddings. In: MÁRQUEZ, L.; CALLISON-BURCH, C.; SU, J. (Ed.). *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, 2015. p. 298–307. Disponível em: <https://aclanthology.org/D15-1036/>.