UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Classificação automática de publicações judiciais

Ricardo Rocha Botti

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

Ricardo Rocha Botti

Classificação automática de publicações judiciais

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Rafael Geraldeli Rossi

Versão original

São Carlos

2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

Botti, Ricardo
S856m Classificação automática de publicações judiciais / Ricardo Rocha Botti ; orientador Rafael Geraldeli Rossi. – São Carlos, 2024.
51 p. : il. (algumas color.) ; 30 cm.

Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. Processamento Natural de Linguagem. 2. Classificação automática de textos. 3. Classe USPSC. 4. Tese. 5. Documentos eletrônicos. I. ROSSI, R. G., orient. II. Título.

Ricardo Rocha Botti

Automatic classification of judicial publications

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. Rafael Geraldeli Rossi

Original version

São Carlos

2024

Este trabalho é dedicado à minha família. Meus pais, minha amada e paciente esposa Luciana, que sempre me deu todo o apoio e carinho, e filhas que compreenderam, na maior parte das vezes, minha ausência durante este período. Saibam que o esforço compensou e foi por vocês.

"Sem liberdade de pensamento, não pode haver conhecimento;
e não há qualquer liberdade pública sem liberdade de expressão."
Benjamin Franklin

RESUMO

BOTTI, Ricardo R. **Classificação automática de publicações judiciais**. 2024. 51p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

Diante do crescente volume de processos judiciais tramitando eletronicamente no Brasil, torna-se essencial desenvolver ferramentas que auxiliem advogados e departamentos jurídicos a lidar com a complexidade e a quantidade de informações disponíveis de forma a evitar falhas que possam levar a perdas de prazos em processos, diminuição de produtividade e até prejuízos financeiros. A classificação de publicações judiciais auxilia os interessados a obter informações corretas e rápidas, o que possibilita uma atuação processual eficiente. Existem diferentes possibilidades para aplicar a classificação automática em textos em português no domínio jurídico, desde soluções criadas a partir do zero, ou fazendo uso de recursos já existentes para o domínio. Visando obter uma solução assertiva, o objetivo deste trabalho de conclusão de curso foi utilizar um processo de classificação de textos baseado na metodologia *framework* CRISP-DM para construir e avaliar soluções para a classificação de atos judiciais. Mais especificamente, foram consideradas abordagens inteiramente construídas para o problema em questão como o BERTikal, que é um modelo *BERT-base cased* para a linguagem jurídica brasileira treinado a partir do *checkpoint* do BERTimbau utilizando textos jurídicos brasileiros. Ainda assim, este modelo não apresentou os resultados esperados. Como principais resultados, têm-se que apesar da expectativa de superioridade dos modelos pré-treinados, abordagens mais tradicionais como *Bag-of-Words* (BoW) combinadas com Regressão Logística podem apresentar resultados superiores em termos de precisão e acurácia.

Palavras-chave: Classificação automática de textos, Classificação de atos judiciais, Publicações Judiciais, CRISP-DM.

ABSTRACT

BOTTI, Ricardo R. **Automatic classification of judicial publications**. 2024. 51p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

Given the growing volume of judicial processes being handled electronically in Brazil, it is essential to develop tools that assist lawyers and legal departments in managing the complexity and amount of information available, in order to avoid errors that could lead to missed deadlines, decreased productivity, and even financial losses. The classification of judicial publications helps stakeholders obtain correct and timely information, thereby enabling efficient procedural actions. There are different possibilities for applying automatic classification to Portuguese texts in the legal domain, ranging from solutions built from scratch to leveraging existing resources for the domain. Aiming to achieve an accurate solution, the objective of this course conclusion paper was to use a text classification process based on the CRISP-DM framework to build and evaluate solutions for the classification of judicial acts. More specifically, approaches entirely designed for the problem at hand were considered, such as BERTikal, a BERT-base cased model for the Brazilian legal language trained from the BERTimbau checkpoint using Brazilian legal texts, which, nevertheless, did not yield the expected results. The main findings show that, despite the anticipated superiority of pre-trained models, more traditional approaches like Bag-of-Words (BoW) combined with Logistic Regression can deliver superior results in terms of precision and accuracy.

Keywords: Automatic text classification, Judicial act classification, Judicial Publications, CRISP-DM.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AI	Artificial Intelligence (Inteligência Artificial)
API	Application Programming Interface (Interface de Programação de Aplicações)
BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag of Words
CNJ	Conselho Nacional de Justiça
CNN	Convolutional Neural Network (Rede Neural Convolucional)
CRISP-DM	Cross Industry Standard Process for Data Mining
GPT	Generative Pre-trained Transformer (Transformador Pré-Treinado Generativo)
IBGE	Instituto Brasileiro de Geografia e Estatística
LLM	Large Language Model (Modelo de Linguagem de Grande Escala)
LLaMA	Large Language Model Meta AI
MLM	Masked Language Modeling
mBERT	Multilingual BERT
SVM	Support Vector Machine
sBERT	Sentence-BERT
TJ-SP	Tribunal de Justiça de São Paulo
USP	Universidade de São Paulo
USPSC	Campus USP de São Carlos

SUMÁRIO

1 INTRODUÇÃO

O Brasil fechou o ano de 2023 com mais de 83 milhões de processos judiciais ativos, sendo que 99,6% tramitam por via eletrônica, percentual que era de apenas 11,10% em 2009 (CNJ, 2023). As movimentações e andamentos destes processos são acompanhados via publicações que têm como objetivo atender ao princípio da publicidade, que traz transparência e permite que a população tenha conhecimento dos atos praticados pelo Poder Judiciário.

No ano de 2023, de janeiro a outubro, somente no TJ-SP, foram publicados, em média, 3,2 milhões de atos processuais mensalmente (Comunicação Social TJSP, 2023), entre decisões interlocutórias e monocráticas, despachos, sentenças e acórdãos. No final de outubro de 2023, o TJ-SP possuía pouco mais de 18 milhões de processos pendentes segundo o Painel CNJ (2023) . Extrapolando essa média para todo o Brasil, teríamos algo em torno de 14,5 milhões de publicações de atos processuais mensais no país.

Escritórios de advocacia e setores jurídicos de empresas possuem grandes desafios ao lidar com um volume enorme de textos não estruturados, como é o caso das publicações de atos judiciais. Pode-se citar: falhas no agendamento de audiências, perdas de prazo de manifestações, interpretações erradas ou incompletas dos comandos judiciais, atribuições de tarefas a processos errados, falhas de contingenciamento, desatualização da posição da carteira, e perda de produtividade ao designar responsáveis menos adequados. Todos estes problemas, caso tratados incorretamente, podem gerar prejuízos financeiros significativos a depender do processo específico. Além disso, quanto maior a carteira de processos de um escritório ou empresa, maior será o volume de dados tratados e os problemas apontados anteriormente aumentam na mesma proporção e se tornam um limitador para o crescimento.

A classificação de textos de publicações judiciais é essencial para que os agentes possuam informações corretas e rápidas seja na esfera operacional para atuação processual eficiente, seja na esfera estratégica para análise de carteiras e jurimetria. Para realizar a classificação automática de textos para um domínio específico, pode-se: i) usar soluções comerciais como o *AutoML* do *Vertex AI (Google Cloud, 2018), Natural Language AI (Google) (Google Cloud, 2016)*, e o *Amazon Comprehend (Amazon, 2017)* ; ii) fazer uso de APIs comerciais de *Large Language Models* (LLMs), como o *ChatGPT (Open AI) (OpenAI, 2020)*; implementando *pipelines* de Mineração de Textos gerando as representações e construindo os modelos de classificação (AGGARWAL, 2015), na qual na etapa de representação pode-se fazer uso de representações construídas especificamente para a coleção de textos, ou ainda fazer uso de modelos pré-treinados como o LegalBERT-pt (SILVEIRA *et al.*, 2023), JurBERT (MASALA *et al.*, 2021), JurisBERT (VIEGAS; COSTA;

ISHII, 2023), BERTikal (POLO *et al.*, 2021b) e BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020).

Vale ressaltar que deve-se ficar atento à análise de custo versus desempenho de cada uma das soluções para que a operação seja viável com grande volume de dados. Deve-se atentar também para a performance de classificação, já que não existe uma única técnica que irá prover os resultados mais acurados para um determinado domínio.

1.1 Objetivos & Perguntas de Pesquisa

Este projeto de conclusão de curso tem como objetivo a aplicação e avaliação de algoritmos de pré-processamento de textos e de aprendizado de máquina para a automatização da classificação de publicações judiciais com base no conteúdo de cada ato. Dessa maneira, serão implementadas etapas metodológicas que orientarão a execução do estudo, delineando um caminho claro para atingir o objetivo proposto, conforme abaixo:

- Utilizar uma base pré-rotulada de publicações judiciais já existente no banco de dados de uma empresa que trata cerca de 25.000 publicações mensais. Essas publicações são rotuladas manualmente;
- Fazer uso de modelos pré-treinados para o domínio jurídico e em português;
- Implementar técnicas de pré-processamento de modo a deixar a base preparada para aplicação dos algoritmos de aprendizagem de máquina;
- Aplicar algoritmos de aprendizado de máquina para classificação ou extração de padrões utilizando os textos das publicações judiciais;
- Avaliar e obter quais as combinações de técnicas de pré-processamento ou modelos pré-treinados e técnicas de aprendizado de máquina são mais adequadas para a classificação dos atos processuais.

Esse objetivos visam responder às seguintes perguntas:

- **P1**: Modelos de linguagem pré-treinados para o domínio jurídico de textos em português são capazes de superar significativamente representações geradas sem pré-treinamento?
- **P2**: O uso de *Large Language Models* de fato consegue fornecer melhorias significativas na acuidade da classificação de textos jurídicos?
- **P3**: Qual a melhor combinação de técnicas (pré-processamento de textos e algoritmos de aprendizado de máquina) para a classificação de atos judiciais escritos em português?

1.2 Organização do Texto

O restante do texto está organizado da seguinte forma. No Capítulo 2, é apresentada a fundação das etapas de um processo de mineração de textos com foco na tarefa de classificação. No Capítulo 3, são apresentados os trabalhos da literatura que envolvem classificação de textos jurídicos em português. No Capítulo 4, são apresentados os detalhes de cada etapa de um processo de mineração de textos utilizados para cumprir com os objetivos propostos neste trabalho de conclusão de curso. Já no Capítulo 5, são apresentados e discutidos os resultados. Por fim, no Capítulo 6, são apresentadas as considerações finais, limitações e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

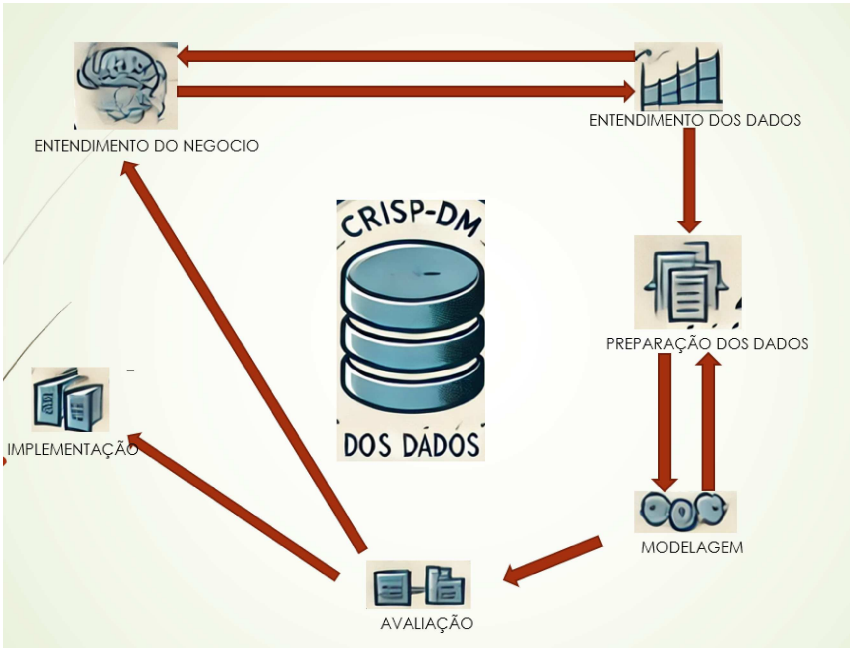
Segundo Aggarwal (2015), a mineração de dados é o ato de coletar, limpar, processar, analisar e obter insights úteis a partir de dados. A mineração de textos é uma subárea da mineração de dados, porém dada a natureza não estruturada dos textos, são necessários passos específicos para lidar com dados com esse tipo de característica (REZENDE, 2003).

A definição de um *framework* para mineração de textos é chave para garantir a estruturação e o sucesso de projetos que visam extrair informações significativas a partir de grandes volumes de dados textuais. O *framework* sistematiza o trabalho por meio de processos estruturados, desde a coleta inicial de dados até a aplicação prática de insights obtidos.

Dentre as opções disponíveis, este trabalho de conclusão de curso adotará o *Cross-Industry Standard Process for Data Mining* (CRISP-DM) chapman2000crisp. A escolha está fundamentada pelo sucesso e o grande uso deste *framework*, sendo alvo de estudos e validações por pesquisadores como Schröer, Kruse and Gómez (2021) e tem sido recomendado em literaturas especializadas como a de Martínez-Plumed *et al.* (2019).

Ao implementar o CRISP-DM, conforme pode ser verificado na Figura 1, a compreensão inicial do negócio e dos dados é seguida por um processo de preparação de dados, que é crítico para a classificação de textos, como enfatizado por Peker and Kart (2023).

Figura 1 – Fases do CRISP-DM



Baseado em Chapman *et al.* (2000)

A modelagem subsequente, um dos núcleos do *framework*, permite a aplicação e

a avaliação de diferentes algoritmos de classificação, adequando-se às peculiaridades do conjunto de dados textual em questão. A fase de avaliação, conforme discutida por Schröer, Kruse and Gómez (2021), é decisiva para assegurar a precisão e a relevância dos modelos de classificação automática desenvolvidos, enquanto o estágio de desdobramento garante a integração dos modelos na prática operacional.

A adequação do CRISP-DM para classificação automática reside na sua estrutura modular e iterativa, que é particularmente vantajosa para o refinamento dos modelos de classificação, conforme sugerido por Kurgan and Musilek (2006).Portanto, é possível voltar a etapas anteriores, caso o resultado da etapa atual não seja desejável. Além disso, o processo é continuo, podendo ser repetido conforme novos dados chegam ou ocorram mudanças no entendimento do negócio.

Este trabalho de conclusão de curso está voltado para o desenvolvimento de classificação automática de textos jurídicos, que é uma das subáreas da mineração de textos. Segundo Jurafsky and Martin (2020), o objetivo da classificação é pegar uma única observação, extrair algumas características úteis e, assim, classificar a observação em uma das várias classes discretas.

Na Tabela 1 são detalhados os passos que compõem o modelo CRISP-DM. Nessa tabela são destacadas as principais tarefas, em negrito, e os resultados esperados de cada etapa, em itálico, trazendo uma visão mais clara e estruturada do procedimento como um todo.

Nas próximas seções são apresentados mais detalhes sobre as etapas do *framework* CRISP-DM com foco na classificação automática de textos.

2.1 Entendimento do Negócio

A fase de Entendimento do Negócio é importante para que os responsáveis pelo projeto entendam de forma mais ampla o funcionamento da organização. Ela requer uma definição exata do problema a ser abordado, de acordo com as orientações do CRISP-DM (MARTÍNEZ-PLUMED *et al.*, 2019). Este momento inicial exige um mergulho profundo nos objetivos estratégicos e nos desafios particulares que o projeto visa resolver. É fundamental, nesta fase sincronizar os objetivos técnicos minuciosos da classificação com as metas mais amplas da organização (DAVENPORT; HARRIS, 2007).

Desenvolver um plano de projeto claro e estruturado é essencial. O plano não deve se limitar apenas descrever o problema, mas precisa propor estratégias concretas e aplicáveis para sua solução. A definição precisa e bem fundamentada do problema cria uma base sólida para o desenvolvimento do projeto, assegurando que a tarefa de classificação de textos siga com foco e direção claros.

Tabela 1 – **Processo de Mineração de Dados**

Entendimento do Negócio	**Entendimento dos Dados**	**Preparação dos Dados**	**Modelagem**	**Avaliação**	**Implementação**
Determinar Objetivos de Negócio	**Coletar Dados Iniciais**	**Selecionar Dados**	**Selecionar Técnicas de Modelagem**	**Avaliar Resultados**	**Planejar Implementação**
- Histórico	- Relatório de Coleta de Dados	- Justificativa para Inclusão/ Exclusão	- Técnica de Modelagem	- Avaliação dos Resultados	- Plano de Implementação
- Objetivos de Negócio	**Descrever os Dados**	**Limpar os Dados**	- Definições de Modelagem	- Critérios de sucesso	**Planejar Monitoramento e Manutenção**
- Critérios de Sucesso	- Relatório de Descrição	- Relatório de Limpeza	**Gerar Projeto de Teste**	- Modelos Aprovados	- Plano de Monitoramento e Manutenção
Avaliar Situação	**Explorar os Dados**	**Construir Dados**	- Projeto de Teste	**Revisar Processo**	**Produzir Relatório Final**
- Inventário de Recursos	- Relatório de Exploração	- Atributos Derivados	**Construir Modelo**	- Revisão do Processo	- Relatório Final
- Requisitos	**Verificar Qualidade**	- Dados Gerados	- Configurações de Parâmetros	**Determinar Próximos Passos**	- Apresentação Final
- Definições, Restrições	- Relatório de Qualidade	**Integrar Dados**	- Modelos	- Lista de Ações Possíveis	**Revisar Projeto**
- Riscos e Contingências		- Dados Mesclados	- Descrição do Modelo		- Documentação de Experiência
- Custos e Benefícios		**Formatar Dados**	**Avaliar Modelo**		
Determinar Metas de Mineração de Dados		- Dados Reformatados	- Avaliação do Modelo		
- Metas de Mineração		- Descrição do Conjunto	- Revisar Configurações		
- Critério de Sucesso					
Produzir Plano de Projeto					
- Plano de Projeto					
- Avaliação Inicial de Ferramentas					

Baseado em Chapman *et al.* (2000)

2.2 Entendimento dos dados

A análise meticulosa dos dados tem papel importante no sucesso de iniciativas focadas na classificação automática de textos. Este estágio abrange desde a coleta inicial até a minuciosa descrição, exploração e avaliação da integridade dos dados (SCHRÖER; KRUSE; GÓMEZ, 2021). Tal abordagem não apenas revela *insights* valiosos sobre o conjunto de dados em questão, mas também destaca eventuais desafios que possam afetar negativamente as etapas futuras do projeto.

Assim, os participantes conseguem ter uma compreensão mais aprofundada da qualidade e configuração dos dados.

Podem ser usadas técnicas para exploração dos dados como gráficos de dispersão e

histogramas, além de estatísticas descritivas como média, mediana e desvio padrão. Essas informações permitem que a compreensão dos dados coletados seja maior.

Portanto, a etapa de verificação da qualidade dos dados permite a identificação de características da base de dados que possam interferir na performance do modelo como valores ausentes, inconsistências, *ouliers*, ruídos, e desbalanceamento de classes.

2.3 Preparação dos Dados

Transformar textos não estruturados em dados prontos para análise constitui um desafio técnico significativo. O processo de preparação dos dados se torna o alicerce para a criação de modelos de classificação eficientes e precisos. Esta fase inicial é fundamental para organizar os dados de forma que permita uma análise aprofundada e eficaz. Assim, tenta-se assegurar que os modelos criados possam operar no seu potencial máximo.

Diferentemente dos dados puramente numéricos ou categorizados, o texto incorpora riqueza semântica e diversidade que exigem um entendimento aprofundado das técnicas envolvidas, contextos e sutilezas específicos. A complexidade existente em textos não estruturados é um fator que demanda uma abordagem diferenciada (AGGARWAL, 2015).

Para obter êxito na análise de textos, é essencial a implementação de processos de limpeza e transformação dos dados. Essas fases causam grande impacto na eficácia dos modelos de aprendizado automático, já que dados não otimizados podem trazer ambiguidade nos significados dos textos, dificultando sua correta interpretação. Algumas medidas comuns são a eliminação de erros ortográficos, *stopwords* (palavras pouco discriminativas, como artigos e preposições) e caracteres incomuns, simplificação dos termos através da radicalização, que mantém apenas o radical das palavras ou lematização, que substitui as palavras considerando o infinitivo dos verbos e masculino singular dos substantivos e adjetivos. Dependendo do domínio, outros tipos de limpezas podem ocorres, como remoção de tags ou outras marcações textuais características de um domínio de aplicação. Uma vez realizada a limpeza e padronização dos textos, passa-se para a fase de gerar uma representação estrturada da coleção de textos. Comumente, adota-se uma representação no formato de vetor de características (AGGARWAL, 2015), formato esse também comum ao processo de Mineração de Dados. A *Bag of Words* (BoW) (SALTON; MCGILL, 1986) é uma das estratégias mais adotadas para a geração de vetores de características para os textos de uma coleção. Primeiramente, é construído um vocabulário com as palavras únicas dentro da coleção de textos. Cada palavra se torna um atributo ou dimensão do vetor gerado. O valor (ou peso) de cada atributo para um vetor que representa o texto é baseado na ocorrência ou frequência daquele atributo no texto. Comumente são utilizados como pesos a frequência do atributo no texto *term-frequecy (TF)*, ou a frequência do termo ponderada pela frequência inversa de documento do inglês, *term-frequency- inverse document frequency - (TF-IDF).*

Após os processos de limpeza e padronização, é construído um vocabulário com as palavras únicas dentro do texto. Cada palavra se torna um atributo ou dimensão do vetor gerado, que tem o mesmo tamanho do vocabulário. Cada posição no vetor, portanto, representa uma palavra e o valor dado em cada posição reflete a quantidade de vezes que a palavra aparece no texto. Em função disso, a dimensionalidade e esparsidade desse tipo de representação tendem a ser altos.

Outro ponto relevante é que palavras semanticamente próximas ou até sinônimos como cachorro e cão ou gato e felino são tratadas como palavras distintas. Além disso, não leva em consideração o contexto e a ordem das palavras no texto. Isso é uma das maiores críticas à técnica, já que o significado das palavras podem variar de acordo com o contexto.

Uma outra forma comum de se representar textos por meio de vetores é fazendo uso de *embeddings* (AGGARWAL, 2015). As embeddings representam uma informação por um vetor numérico não esparso e geralmente com dimensionalidade menor que a representação bag-of-words, além de permitir capturar e representar relações implícitas, como a semântica. Dentro do domínio textual, modelo aprendido para gerar tais *embeddings* também é conhecido como modelo de linguagem (AGGARWAL, 2015). As *embeddings* primeiramente se popularizaram com as *word embeddings*, ou seja, representam palavras.

Dentre as abordagens de *word embeddings*, pode-se destacar a Word2Vec (MIKOLOV *et al.*, 2013) e a Glove (PENNINGTON; SOCHER; MANNING, 2014). A abordagem Word2Vec que introduziu a utilização de redes neurais para a geração de *word embeddings* e permitiu o tratamento de grandes volumes de texto de maneira eficiente, através das técnicas CBOW e Skip-gram (MIKOLOV *et al.*, 2013). O *Continuous Bag of Words* (CBOW) tem como objetivo prever uma palavra baseado nas palavras vizinhas (contexto). Aqui é utilizada uma rede neural com apenas uma camada oculta e as palavras são representadas por vetores densos, o que significa que cada dimensão traz informações sobre a palavra e seu contexto e contribui para identificar relações semânticas e contextuais, diferentemente de uma simples contagem de frequência.

Já o *Skip-gram* tem o funcionamento inverso ao CBOW. O seu objetivo é prever o contexto a partir de uma palavra. Também utiliza uma rede neural com apenas uma camada oculta e traz como resultado quais seriam as palavras do contexto. Ele é mais eficiente em termos de processamento que o CBOW, principalmente, em vocabulários longos, já que trata individualmente cada palavra e é mais adequado para tarefas em que o contexto é mais importante (MIKOLOV *et al.*, 2013).

Posteriormente, surge o GloVe (*Global Vectors for Word Representation*), tem como objetivo utilizar os coocorrência entre palavras que ocorrem no corpus todo, e não só as palavras vizinhas, para aprender os vetores que irão representar as palavras (PENNINGTON; SOCHER; MANNING, 2014). Outra diferença em relação ao Word2Vec

é que o aprendizado das *embeddings* se dá utilizando fatoração de matrizes ao invés de utilizar redes neurais.

Como Word2Vec ou glove geram vetores para as palavras, posteriormente são necessário o emprego de técnicas para se gerar as *embeddings* dos textos. Comumente empregam-se operações nos vetores, como a soma dos vetores das palavras ou a média dos vetores das palavras para gerar a *embedding* do texto (PENNINGTON; SOCHER; MANNING, 2014). Alternativamente ao emprego dessas operações, pode-se utilizar a abordagem Doc2Vec (LE; MIKOLOV, 2014), a qual é capaz de gerar *embeddings* para gerar representações sentenças, parágrafos ou até documentos inteiros.

A abordagem Doc2Vec é baseada no Word2Vec, e também possui duas variantes: *Distributed Memory* (PV-DM), equivalente ao DBOW, e *Distributed Bag-of-Words* (PV-DBOW) equivalente ao Skip-gram. A principal diferença é que no Doc2Vec, concomitantemente ao aprendizado das word embeddings, também são geradas / aprendidas as embeddings de parágrafos (ou documentos).

As abordagens de *embeddings* descritas acima são independentes de contexto. Isso significa que a embedding de uma palavra é sempre a mesma, independe das palavras que ocorrem ao redor. Isso significa que um "banco"de instituição financeira, ou um "banco"de assento tem a mesma *embeeding*, o que por diminuir a capacidade de representação desses vetores.

Dada essa lacuna, surgiram concomitantemente modelos dependentes de contexto, baseados na arquitetura de redes neurais do tipo transformers[1], e com um número de parâmetros no modelo muito maior que as abordagens anteriores, os chamados *Large Language Models* (LLMs) (BROWN *et al.*, 2020). Alguns exemplos de LLM são o BERT (DEVLIN *et al.*, 2018b), GPT-3 (BROWN *et al.*, 2020), e suas evoluções, incluindo GPT-4 (OPENAI, 2023) e LLaMA (TOUVRON *et al.*, 2023). Para fins de ilustração do tamanho de tais modelos, o BERT possui 340 milhões de parâmetros, o LLaMA varia entre 7 bilhões a 70 bilhões de parâmetros, o GPT3 possui 175 bilhões de parâmetros, e o GPT4 possui 170 trilhões de parâmetros.

Vale ressaltar que apesar de baseados em transformers, BERT e GPT possuem arquiteturas, finalidades e treinamento diferentes. Por exemplo, o BERT é treinado para

[1] Os transformers são uma rede neural projetada para capturar contexto a partir do processamento de sequências longas de texto. Ao invés de utilizar mecanismos como recorrência ou convolução, que possuem mais dificuldade de aprender dependências em textos longos, os transformers possuem um mecanismo de atenção, que calcula pesos entre as palavras de uma sequência e permite que o modelo capture dependências entre as palavras e sobre o contexto geral sem a necessidade de conexões recorrentes(VASWANI *et al.*, 2017). A transformação trazida por esses modelos, ampliada pela tecnologia dos *transformers*, não apenas aprimorou a compreensão contextual das palavras, mas também a habilidade de gerar textos de alta coerência, rivalizando com a produção humana.

prever palavras que estão faltando dentro de uma contexto, enquato o GPT é treinado para predizer a próxima palavra de um contexto. Porém, ambos são capazes de gerar *embeddings*. Além disso, é possível adaptar um modelo para um domínio específico e especializá-lo para uma tarefa dado um conjunto de treinamento (*fine-tuning*). Por exemplo, é comum a adição de uma rede densa para a classificação em cima de um modelo BERT pré-treinado e fazer uma fine-tuning para uma tarefa de classificação específica (SUN *et al.*, 2019).

2.4 Modelagem

Após a adequada transformação dos dados e balanceamento das classes, é o momento de testar os diversos algoritmos de aprendizado de máquina. Existem modelos que são mais adequados a cada tipo de corpus e objetivo que se pretende alcançar. No entanto, a modelagem envolve muita experimentação e testes. No caso deste trabalho, os testes visam construir um modelo preditivo para classificação de textos e publicações jurídicas.

Alguns dos modelos mais usados são a Regressão Logística, ideal para classificação binária, mas que também pode ser adaptado para o contexto multi-classes. É um modelo matemático que calcula a probabilidade de ocorrência de um evento baseado em variáveis independentes, que poderiam influenciar o resultado. Sua base é a função sigmóide que ajusta qualquer número real para o intervalo entre 0 e 1, sendo que 1 significa 100% de chance do evento ocorrer.

Outro modelo é o *Support Vector Machine* (SVM), que é um modelo de aprendizado supervisionado e se destaca na classificação de dados de alta dimensão. Seu objetivo é identificar o hiperplano que maximiza as margens de separação das diferentes classes. Ainda que os dados não sejam linearmente separáveis, é possível utilizar kernels para mapear os dados para uma nova dimensão em que os dados de cada classe serão linearmente separáveis. Isso o torna eficiente em relação a *overfitting* sem exigir muito recurso computacional (CORTES; VAPNIK, 1995).

Técnicas mais avançadas, como o *fine-tuning* do modelo BERT, também constituem uma excelente opção para construção do modelo de classificação, como explanado no item anterior (DEVLIN *et al.*, 2018b). Essa última é considerada o estado da arte atualmente já que é possível aproveitar um modelo previamente treinado com uma grande quantidade dados e ajustar parâmetros para executar a classificação pretendida.

2.5 Avaliação

É importante avaliar o modelo para estimar o seu desempenho quando for utilizado na prática. Para isso, deve-se definir um esquema de avaliação e uma ou mais métricas de avaliação.

O esquema de avaliação visa remover um subconjunto de exemplos do treinamento

do modelo para depois utilizar esses exemplos para avaliar o modelo, sendo que esse processo de treinamento e avaliação pode ser iterativo. Nesse contexto, têm-se as técnicas hold-out, que visa definir um conjunto de treino e teste, ou a validação cruzada em k pastas (*k-fold cross-validation*), que divide o conjunto de dados em k pastas e iterativamente uma pasta é usada para o teste e as demais para o treinamento do modelo.

Já as métricas de avaliação visam gerar um valor de forma que se possa mensurar os acertos ou erros do modelo de classificação. O levantamento dessas métricas é fundamental para avaliação da qualidade do modelo e para entendimento se a performance atende os requisitos do projeto, além de identificar problemas e melhorias possíveis no modelo. Geralmente a geração dessas métricas é baseada em uma matriz de confusão (ROSSI, 2015):

- Verdadeiro Positivo (VP): É o número de exemplos de teste corretamente classificados como positivos pelo modelo;
- Falso Positivo (FP): É o número de exemplos de teste incorretamente classificados como positivos pelo modelo;
- Verdadeiro Negativo (VN): É o número de exemplos de teste corretamente classificados como negativos pelo modelo;
- Falso Negativo (VN): É o número de exemplos de teste incorretamente classificados como negativos pelo modelo;

A partir desse levantamento podem ser calculadas as seguintes métricas de avaliação:

- Acurácia: Percentual de classificações corretas sobre o total de exemplos de teste;
- Precisão: Percentual de classificações com VP sobre o total de exemplos de teste classificados como positivos;
- Revocação: Percentual de VP sobre todos os exemplos de teste realmente positivos;
- F1-Score: Balanceia Precisão e Revocação em um único número. É a média harmônica entre estes itens e incentiva que haja um equilíbrio entre eles;

Em cenários multiclasse, a matriz de confusão pode ser ampliada para exibir contagens de cada classe. As métricas de avaliação podem ser calculadas para cada classe e depois tirando a média (*macro-averaging*)ou somando os termos utilizados nos cálculos das métricas para todas as classes (*micro-averaging*). Pode ser também utilizada uma matriz de confusão normalizada.

Também pode-se trabalhar otimizando hiperparâmetros e validando a capacidade de generalização dos modelos (GÉRON, 2019). Este processo garante que os modelos finais sejam precisos, confiáveis e aplicáveis a novos dados.

Existem duas estratégias comumente usadas para calcular métricas de desempenho em problemas de classificação multi-classe: Macro-Averaging e Micro-Averaging (ROSSI, 2015).

- Macro-Averaging: Calcula as métricas (como precisão, recall, F1-score) individualmente para cada classe e, em seguida, faz a média aritmética dessas métricas. Dessa forma, todas as classes têm o mesmo peso, independentemente do número de amostras pertencentes a cada classe. É útil quando o desempenho em classes minoritárias é tão importante quanto em classes majoritárias.
- Micro-Averaging: Calcula o desempenho somando os verdadeiros positivos, falsos positivos e falsos negativos de todas as classes e depois aplica essas somas para calcular a métrica desejada. Aqui, o resultado é influenciado pelo tamanho de cada classe, já que considera cada instância de teste igualmente. É útil quando as classes têm distribuição de tamanhos muito diferentes.

2.6 Implementação

A etapa de colocar em produção os modelos gerados é crítica para que os modelos sejam utilizados de forma correta e contínua. É fundamental que os sistemas adotados na operação e os modelos sejam integrados e sejam estáveis. A interface deve ser intuitiva para que os usuários consigam perceber o valor do modelo e não tenham dificuldades na utilização (ZAHARIA *et al.*, 2018).

Modelos de classificação de texto, especialmente no mundo jurídico que sofre atualizações de leis, mudanças de entendimento e consolidação de jurisprudências, precisam ser continuamente monitorados para acompanhamento constante da performance e eventual necessidade de retreinamento Sculley *et al.* (2015).

Outros desafios importantes incluem escalabilidade sem comprometer performance, segurança dos dados e das previsões através da implementação de protocolos de segurança robustos para proteger contra acessos não autorizados (KUMAR *et al.*, 2016). Assim, a implementação não é apenas o processo inicial de colocar o modelo em produção, mas a criação de rotinas e processos que favoreçam o acompanhamento contínuo de indicadores que demonstrem a performance e eventual necessidade de otimização/adaptação dos modelos gerados, assegurando a eficácia, segurança e relevância dos modelos em um ambiente operacional em evolução.

3 TRABALHOS RELACIONADOS

A classificação automática de textos jurídicos tem despertado bastante interesse. O domínio jurídico é particularmente desafiador, já que a linguagem adotada é própria e os termos e forma de apresentação são muito diferentes das formas convencionais. O cenário jurídico brasileiro tem muitas peculiaridades e difere significativamente do direito americano. Dado isso, neste trabalho de conclusão de curso foram relacionados trabalhos voltados para a criação de um modelo especializado na realidade jurídica brasileira e, obviamente, na língua portuguesa. Esses trabalhos, certamente, fornecerão uma base de comparação melhor em termos de resultados esperados e alcançados.

O primeiro estudo desenvolveu um modelo nomeado de JurisBERT (VIEGAS; COSTA; ISHII, 2023) e foi dividido em três etapas. Na primeira etapa, foi realizado o treinamento utilizando a modelagem *Masked Language Modeling* (MLM) começando do zero e utilizando apenas textos jurídicos como leis, sentenças, acórdãos, ementas e bibliografia jurídica brasileira. Na segunda etapa, foram realizados experimentos utilizando a arquitetura Sentence-BERT (sBERT) (REIMERS; GUREVYCH, 2019). O modelo sBERT foi projetado para comparação de semelhança semântica entre sentenças, com modelos pré-treinados como o BERT multilíngue composto por 104 idiomas diferentes e o BERTimbau, pré-treinado para o português brasileiro. Por fim, na terceira etapa, foi feito o *fine-tuning* dos modelos *multilingual* BERT (mBERT) and BERTIimbau com o conjunto de dados específico do estudo contendo 24 mil pares de ementas com grau de similaridade entre os textos variando entre 0 e 3 retirados dos sites dos tribunais. O estudo demonstrou que o JurisBERT foi 22% superior ao BERT multilíngue e 12% superior ao BERTimbau sem fine-tuning na métrica F1; com *fine-tuning*, o JURISBERT foi 20% superior ao multilíngue e 4% superior ao BERTimbau.

O segundo estudo realizado desenvolveu um modelo nomeado de LegalBERT-pt (SILVEIRA *et al.*, 2023). Foram criadas duas versões do modelo. A primeira tendo como base o BERTimbau e a segunda foi criada do zero usando apenas um corpus específico com linguagem jurídica. A base jurídica foi composta por 1,5 milhão de documentos legais em português entre petições iniciais, decisões, sentenças e acórdãos obtidos pelo sistema Codex do CNJ. Os textos foram divididos em sentenças com, no máximo, 512 tokens gerando um total de 12 milhões de sentenças. Os resultados demonstraram que o uso de modelos treinados com linguagem jurídica específica produz resultados superiores a modelos com linguagem genérica. O modelo que teve como base o BERTimbau com *fine-tuning* da base jurídica apresentou os menores valores de perplexidade, indicando um melhor entendimento da linguagem, e os maiores *scores* F1. O resultado, portanto, demonstrou a eficácia do *fine-tuning* de textos jurídicos diversificados em um modelo genérico entre 3% a 4,5% a fim

de obter uma maior especialização do domínio e trazer melhores resultados para tarefas específicas.

O terceiro estudo realizado desenvolveu e disponibilizou modelos de linguagem voltados exclusivamente para o idioma jurídico do Brasil (POLO *et al.*, 2021a) . Dentre os modelos, destacam-se: Phraser, Word2Vec, Doc2Vec, FastText e BERTikal. O modelo Phraser mostrou-se eficaz na identificação de combinações significativas de palavras, enriquecendo as representações textuais ao tratar sequências de palavras como unidades únicas. Os modelos Word2Vec/Doc2Vec foram experimentados com configurações variadas, abrangendo tamanhos de vetores de 100 a 300 e diferentes técnicas, refletindo nas representações vetoriais o contexto semântico dos termos jurídicos. Já o modelo FastText destacou-se por considerar a estrutura morfológica das palavras, gerando vetores de alta qualidade mesmo para termos menos comuns, através do aproveitamento das subestruturas das palavras. Por fim, o modelo BERTikal, uma adaptação do BERT focada no domínio jurídico brasileiro, provou ser excepcional em oferecer representações profundas para textos completos, com base em uma metodologia de linguagem mascarada. A aplicabilidade desses modelos foi comprovada em tarefas de PLN no setor jurídico, evidenciando uma evolução significativa. Por exemplo, a combinação de Word2Vec e Doc2Vec com técnicas avançadas de aprendizado, como redes neurais convolucionais (CNNs) e CatBoost, permitiu classificar procedimentos legais com elevada precisão. Os resultados incluíram: Classificação utilizando Word2Vec e CNN: alcançou precisão de 0,84 e F1-macro de 0,80; Classificação com Doc2Vec e CatBoost: atingiu precisão de 0,86 e F1-macro de 0,82; Classificação usando BERTikal e CatBoost: obteve precisão de 0,86 e F1-macro de 0,82.

Todos esses estudos abordam o desempenho de modelos de linguagem especializados na análise de textos jurídicos. Os dois primeiros estudos confirmam a tese de que a adaptação ao domínio específico potencializa a precisão em tarefas de PLN no âmbito jurídico brasileiro. Enquanto o último fornece modelos pré treinados e funções específicas para a linguagem jurídica brasileira.

4 METODOLOGIA DE PESQUISA

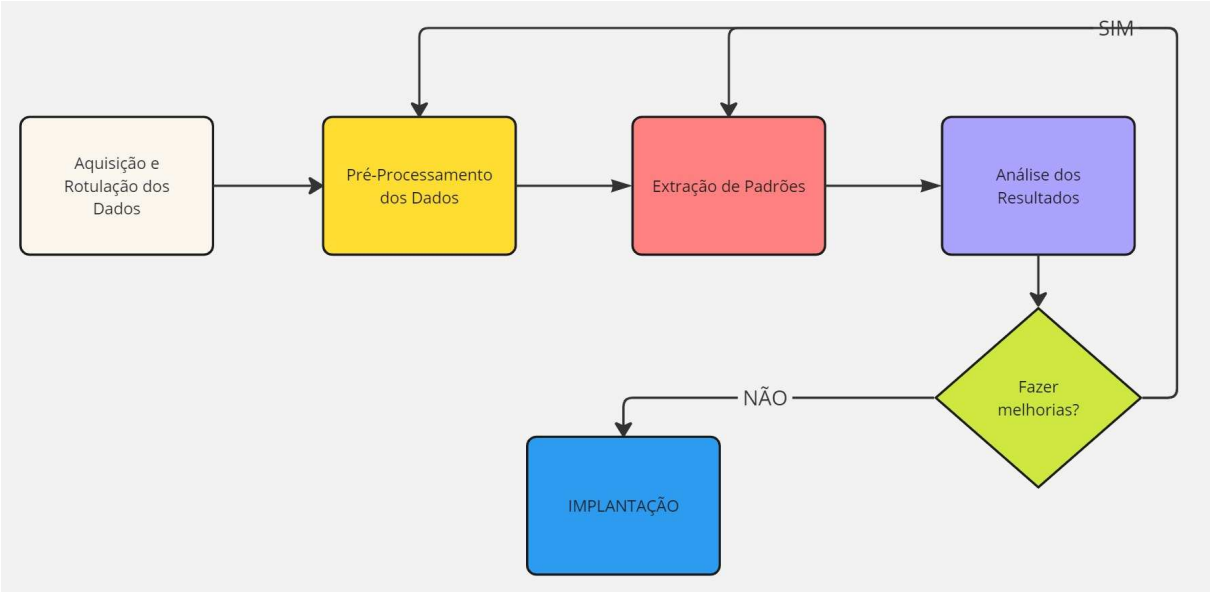
Neste capítulo serão apresentados os detalhes da metodologia de pesquisa adotada no desenvolvimento do presente trabalho, CRISP-DM, apresentada no Capítulo 2. Cada seção a seguir refere-se à uma etapa da metodologia.

4.1 Entendimento dos Dados e do Negócio

Os dados foram coletados da base de uma *legaltech* que atua com direito do consumidor em todos os estados brasileiros. Os dados são atos processuais publicados pelos diversos tribunais brasileiros.

Estas publicações são capturadas por empresas especializadas e disponibilizadas via API para empresas que desejam consumir esses dados, conforme ilustrado na Figura 2. A empresa em questão processa essas publicações com advogados e faz a rotulação manual para que as devidas providências sejam tomadas dentro do processo, conforme ilustrado na Figura 3.

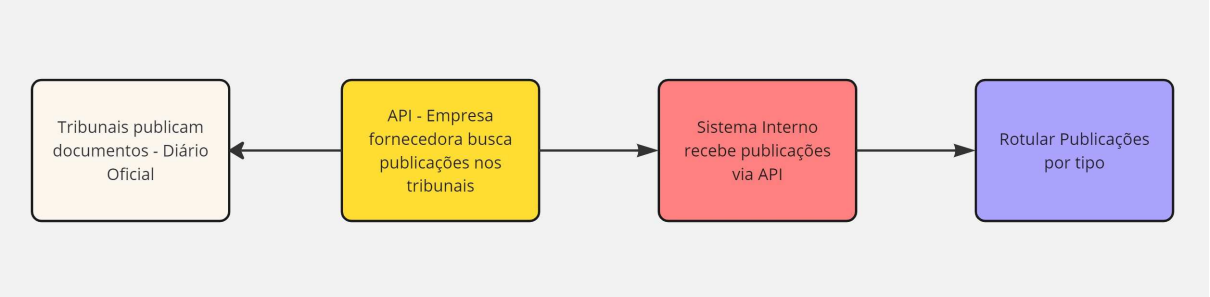
Figura 2 – Processo de Mineração de Textos.



Fonte: Elaborada pelo autor.

A base utilizada para esse estudo possui 12.767 publicações classificadas manualmente por advogados da empresa. A base de dados completa possuía mais de 250 mil exemplos, mas, em função da necessidade de revalidação, ajustes na rotulação e disponibilidade de recursos computacionais foi necessionário reduzir o período de coleta dos dados resultando no *dataset* representado abaixo. As classes e o número de exemplos por classes é apresentada na Tabela 2.

Figura 3 – Processo de Mineração de Textos adotado por legaltech para classificação de publicações.



Fonte: Elaborada pelo autor.

Tabela 2 – **Número de exemplos por classe da base de publicações.**

Nome	**Freq**
Contrarrazões/vista do recurso	998
Especificação de provas	997
Certidão de trânsito – despacho de retorno dos autos da segunda instância	995
Contrarrazões/vista dos embargos	985
Determinada expedição de MLE	983
Despacho em geral (nenhum das anteriores)	878
Declínio de competência	769
Despacho saneador	735
Vista da contestação/defesa prazo para réplica	731
Dilação deferida	651
Sentença de indeferimento inicial	488
Sentença/decisão de embargos de SENTENÇA (1a instância)	480
Sentença de extinção da execução	455
Sentença/Decisão de embargos de ACÓRDÃO (2a instância)	433
Sentença de mérito	430
Acórdão	416
Suspensão IRDR	356
Decisão não admitindo Recurso Especial	241
Intimação para pagamento autor/réu	196
Vista do depósito	165
Prazo do réu (quando não tiver nenhuma das classificações anteriores)	164
Juntar MLE/pedido de transferência	92
Sentença de agravo	31
Marcação de perícia e apresentação de quesitos	30
Vista dos cálculos. (Se estiver errado apresenta impugnação)	28
Prazo interno (criado manualmente)	26
Vista da impugnação aos cálculos	14
Total	**12.767**

Fonte: Elaborada pelo autor.

4.2 Preparação dos dados

No pré-processamento foram usadas as funções da biblioteca LegalNLP (POLO *et al.*, 2021a), que é pré-treinada para a linguagem jurídica brasileira para limpeza de dados. Ela pode utilizar expressões regulares para mascarar e extrair informações específicas, remover caracteres especiais e converter todas as palavras para minúsculas de forma a garantir que os dados estejam em um formato adequado para alimentar e melhorar o desempenho dos algoritmos de classificação. O uso das funções da biblioteca dependerá de qual será o modelo utilizado para a extração dos padrões dos dados.

Foram usadas quatro técnicas de representação de texto para representar os dados das publicações:

- *Bag-of-words* (BoW): Representa cada documento como um vetor de frequências de palavras, capturando a presença e a contagem de termos sem considerar a ordem;

- Word2vec: Foram gerados vetores de palavras de tamanho 200 com uma janela de contexto de 15, representando cada documento pela média dos vetores das palavras que o compõem, excluindo palavras fora do vocabulário utilizando o método CBoW;

- Doc2vec: Foram gerados vetores de documentos de tamanho 200 com uma janela de 15, representando cada documento pela média dos vetores dos documentos individuais que o formam utilizando a abordagem *Distributed Memory* (DM);

- FastText: Foram gerados vetores de palavras de tamanho 200 com uma janela de 15, representando cada documento pela média dos vetores das palavras que o compõem, incluindo subpalavras e removendo palavras fora do vocabulário. Foi utilizado o parâmetro padrão do n-gram que considera sequências de 3 a 6 caracteres como n-grams.

Os tamanhos das representações e janelas utilizados foram definidos de acordo com parâmetros sugeridos na biblioteca do LegalNLP (POLO *et al.*, 2021a).

Para as representações *Bag-of-words, word embeddings* e *doc2vec* foram padronizadas as caixas, removidas as *stopwords* e os termos foram radicalizados.

4.3 Modelagem

Quatro algoritmos de classificação foram treinados nos dados de representação do item anterior:

- Regressão Logística: um modelo de Regressão Logística foi treinado nos dados de representação, utilizando os parâmetros padrão do Scikit-learn[1];
- SVM: um modelo SVM com kernel linear foi treinado nos dados de representação, utilizando os parâmetros padrão do Scikit-learn[2];
- BERT: Aqui foram usados o *fine-tuning* de três modelos BERT. Um BERT multilígue (bert-base-multilingual-cased) (DEVLIN *et al.*, 2018a) e um BERT com pré treinamento para o português brasileiro (BERTimbau) (SOUZA; NOGUEIRA; LOTUFO, 2020), os dois sem pré-treinamento para o domínio jurídico e o BERTikal (POLO *et al.*, 2021b), que é um BERT-*base model* para a linguagem jurídica brasileira para comparação do resultado entre eles. Os dados serão tokenizados com um comprimento máximo de 512 e os pesos das classes serão calculados para lidar com o desbalanceamento. O modelo foi treinado por 10 épocas com *early stopping* para evitar o overfitting. Tamanho de lote de treinamento e avaliação de 16 por batch. Foram utilizados 10% dos exemplos de treinamento como passos de aquecimento e um decaimento de peso de 0.01. A estratégia de avaliação será aplicada a cada 10.000 passos.

4.4 Avaliação

Os modelos foram avaliados com base nas métricas de precisão, *recall* e F1-*score*. A precisão indica a proporção de exemplos que foram corretamente identificados. O *recall*, por sua vez, avalia a proporção de exemplos positivos que foram corretamente reconhecidos. O F1-*score* representa a média harmônica ponderada entre a precisão e o *recall*.

[1] <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>. Acesso em: 28 set. 2024.

[2] <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>. Acesso em: 28 set. 2024.

5 AVALIAÇÃO EXPERIMENTAL

Neste capítulo serão apresentados os resultados obtidos após a aplicação dos métodos descritos no capítulo anterior. Na Tabela 3 estão os resultados separados por representação e modelo utilizado. Abaixo serão apresentadas as estatísticas gerais e tamanho do vocabulário nas representações e feitas análises por modelo, por representação e geral. Ao final, será apresentada uma conclusão considerando os modelos utilizados e os resultados obtidos.

5.1 Estatísticas Gerais e Tamanho do Vocabulário

Durante a execução deste trabalho, foram extraídas estatísticas gerais dos textos processados, bem como o tamanho dos vocabulários gerados nas diferentes representações. Esses dados fornecem uma visão importante sobre as características da base de dados utilizada e o impacto dessas características no desempenho dos modelos de classificação.

As estatísticas gerais dos textos processados são as seguintes:

- **Tamanho médio dos textos**: 258,56 palavras
- **Tamanho mínimo dos textos**: 27 palavras
- **Tamanho máximo dos textos**: 14.867 palavras
- **Desvio padrão do tamanho dos textos**: 423,07 palavras

Essas métricas evidenciam uma grande variação no tamanho dos textos, com um desvio padrão considerável. Essa variação pode influenciar o desempenho dos modelos, já que textos mais longos podem exigir maior capacidade computacional e potencialmente introduzir ruído, enquanto textos muito curtos podem não conter informações suficientes para uma classificação adequada.

Os tamanhos dos vocabulários gerados para as representações foram:

- **Bag-of-Words (BoW)**: 34.271 palavras
- **Word2Vec**: 34.312 palavras
- **Doc2Vec**: 34.312 palavras
- **FastText**: 34.312 palavras

O tamanho do vocabulário impacta diretamente a complexidade do modelo e o tempo de processamento. Modelos como Bag-of-Words (BoW) podem resultar em vetores de alta dimensionalidade, o que pode aumentar a complexidade computacional e introduzir esparsidade. Por outro lado, representações baseadas em embeddings, como Word2Vec, Doc2Vec e FastText, utilizam representações densas, o que pode reduzir a dimensionalidade e melhorar a eficiência computacional.

5.2 Análise Geral

De acordo com os resultados apresentados na Tabela 3 pode-se verificar que o modelo SVM com *BoW* superou os demais modelos em todas as métricas atingindo uma acurácia de 0,86, Precisão de 0,86, Revocação de 0,86, *F1-Score Micro* de 0,86 e *F1-Score Micro* de 0,74. Esse resultado não confirma os resultados dos estudos apresentados na seção de trabalhos relacionados que demonstraram performance superior de modelos pré treinados em tarefas específicas.

Outro ponto que chama atenção nos resultados é a diferença entre *F1-Score Macro* e *F1-Score Micro* em todos os testes. Este resultado sugere que o modelo está tendo dificuldades de classificar corretamente as classes minoritárias.

Tabela 3 – Resultados dos Modelos com Diferentes Representações de Texto.

Representação	**Modelo**	**Acurácia**	**F1-Score Macro**	**F1-Score Micro**
BoW	*Regressão Logística*	**0,86**	**0,74**	**0,86**
	SVM	0,85	0,73	0,85
Word2Vec	*Regressão Logística*	0,78	0,65	0,78
	SVM	0,81	0,71	0,81
Doc2Vec	*Regressão Logística*	0,74	0,59	0,74
	SVM	0,05	0,03	0,05
FastText	*Regressão Logística*	0,77	0,65	0,77
	SVM	0,80	0,69	0,80
BERT	*fine tuning*	0,78	0,61	0,78
BERTimbau	*fine tuning*	0,77	0,63	0,77
BERTikal	*fine tuning*	0,78	0,63	0,78

Ao comparar os modelos de Regressão Logística, *Support Vector Machine* (SVM), BERT, BERTimbau e BERTikal para diferentes representações, observou-se variações significativas no desempenho de cada um. Na representação *Bag of Words* (BoW), a Regressão Logística destacou-se, alcançando uma acurácia, precisão, revocação e F1-Score Micro de 0,86. Esses resultados indicam um desempenho consistente do modelo em todas as classes. O SVM também apresentou métricas positivas, com valores ligeiramente inferiores (0,85 para acurácia e F1-Score Micro), embora o F1-Score Macro tenha sido de 0,73, o que sugere uma menor capacidade do modelo em lidar com classes menos representadas.

Para a representação Word2Vec, a Regressão Logística obteve uma acurácia de 0,78 e um F1-Score Macro de 0,65, apontando para uma performance razoável, mas com dificuldades em lidar com classes desbalanceadas. Por outro lado, o SVM obteve um desempenho geral superior, com acurácia de 0,81 e um F1-Score Macro de 0,71, sugerindo uma melhor adequação dessa representação a esse modelo.

No caso da representação Doc2Vec, os resultados foram menos satisfatórios. A Regressão Logística apresentou uma acurácia de 0,74 e um F1-Score Macro de 0,59, enquanto o SVM teve um desempenho muito inferior, com uma acurácia de apenas 0,05 e F1-Score Macro de 0,03, o que evidencia que o SVM não conseguiu capturar informações suficientes dessa representação para realizar a classificação de maneira eficaz.

Na representação FastText, os resultados da Regressão Logística foram semelhantes aos de Word2Vec, com uma acurácia de 0,77 e um F1-Score Macro de 0,65. No entanto, o SVM novamente superou a Regressão Logística, alcançando uma acurácia de 0,80 e F1-Score Macro de 0,69, sugerindo que FastText pode ser uma representação mais eficaz para esse modelo.

Quanto aos modelos BERT, BERTimbau e BERTikal, todos apresentaram desempenho semelhante após o fine-tuning, com acurácia e F1-Score Micro variando entre 0,77 e 0,78, e F1-Scores Macro entre 0,61 e 0,63. Isso demonstra que esses modelos possuem um desempenho equilibrado, mas não excepcional em classes menos representadas.

Em conclusão, a análise dos diferentes modelos de aprendizado de máquina para a tarefa de classificação automática utilizando várias representações revelou resultados interessantes. A Regressão Logística com BoW destacou-se por superar os modelos BERT com fine-tuning, o que indica que, apesar da sofisticação dos modelos pré-treinados, os algoritmos clássicos como a Regressão Logística ainda são uma alternativa viável e eficiente, especialmente em contextos onde há limitações de recursos computacionais.

No que se refere à eficácia das representações, a BoW mostrou-se superior em todas as métricas analisadas, principalmente quando utilizada com Regressão Logística, que apresentou os melhores resultados. Mesmo com SVM, BoW apresentou métricas superiores às obtidas pelos modelos BERT. Em contrapartida, a representação Doc2Vec teve o pior desempenho, especialmente com o SVM, que praticamente falhou ao tentar utilizar essa representação. Embora os modelos BERT, após o fine-tuning, tenham apresentado resultados consistentes, eles não conseguiram superar de maneira significativa as representações tradicionais combinadas com SVM ou Regressão Logística, particularmente em termos de F1-Score Macro. Isso pode indicar que o conjunto de dados utilizado não explora plenamente as capacidades desses modelos mais avançados ou que o volume de documentos disponível para o fine-tuning não foi suficiente para maximizar seu desempenho. Assim, a combinação de BoW com Regressão Logística mostrou-se uma solução robusta e superior às demais.

6 CONCLUSÕES

Este trabalho de conclusão de curso foi motivado pelo desafio crescente enfrentado por advogados e departamentos jurídicos ao lidar com o vasto volume de publicações judiciais eletrônicas no Brasil, especialmente em um contexto onde a automatização de processos se torna cada vez mais necessária para garantir eficiência e precisão na execução de comandos processuais. O objetivo principal deste trabalho foi explorar e avaliar diferentes técnicas de aprendizado de máquina aplicadas à classificação automática de textos jurídicos, com foco na identificação de abordagens que possam oferecer uma solução prática e eficaz para o setor.

Os resultados encontrados neste trabalho são contrários aos descritos na literatura que demonstram a superioridade de modelos de linguagem pré treinados com *fine-tuning* para tarefas específicas de classificação. Os resultados apresentaram um desempenho muito superior da configuração clássica (BoW + Regressão Logística) em relação aos modelos BERT.

A eficiência dos modelos BERT, que normalmente se destacam em datasets grandes com variação significativa, não é tão impressionante aqui. Isso sugere que o *dataset* pode não ser grande o suficiente ou não ter a complexidade necessária para que BERT, mesmo após o fine-tuning, mostre vantagens substanciais sobre métodos mais tradicionais. Modelos grandes como BERT podem sofrer de *overfitting* em bases de dados menores ou mais específicos, onde métodos como BoW ou Word2Vec podem ser suficientes e até preferíveis.

Considerando que o conteúdo do *dataset* é composto por publicações judiciais que são textos técnicos e estruturados, com termos repetidos frequentemente, abordagens simples, como BoW e Regressão Logística podem ser eficazes, enquanto métodos como Doc2Vec e SVM, que procuram padrões mais complexos e distribucionais, não oferecem o mesmo nível de performance.

O desempenho observado sugere que o *dataset* em questão possui características que favorecem abordagens mais tradicionais e menos complexas, como BoW, especialmente em combinação com Regressão Logística. A possível presença de classes desbalanceadas e a falta de complexidade semântica substancial no texto fazem com que métodos mais sofisticados, como BERT, não apresentem uma vantagem clara. Portanto, a escolha do modelo e da representação deve levar em conta a especificidade do dataset, onde simplicidade e frequência de termos podem ser mais eficazes do que representações semânticas complexas.

A abordagem identificada como a mais eficaz para a classificação de atos judiciais, baseada na combinação de *Bag-of-Words* e Regressão Logística, não apenas se destacou em termos de precisão e robustez, mas também apresenta vantagens significativas em

relação aos recursos computacionais exigidos. Diferente dos modelos baseados em BERT, que demandam maior poder de processamento e tempo de execução, a solução proposta requer menos infraestrutura tecnológica, permitindo um processamento mais rápido e eficiente. Isso torna a abordagem não apenas mais acessível, mas também mais adequada para ser implementada em ambientes com limitações de recursos, garantindo eficiência sem comprometer a qualidade dos resultados.

Durante a execução deste trabalho enfrentamos algumas dificuldades como limitação de recursos computacionais para processar o *dataset* completo que possui mais de 270 mil exemplos rotulados, falta de padronização na rotulação dos dados, já que a rotulação foi feita por cinco advogados diferentes, e o desbalanceamento das classes. Foi necessário, portanto, utilizar apenas uma parte do *dataset* para viabilizar o trabalho tanto no aspecto de processamento quanto na possibilidade de revisão de rotulação.

Em função dos resultados obtidos, pode-se considerar para trabalhos futuros que sejam utilizadas técnicas de balanceamento de classes para aumentar a taxa de acerto na classificação das classes minoritórias; seja utilizado um *dataset* com mais exemplos para verificar se a performance dos modelos BERT melhora para tarefa de classificação automática; a rotulação dos dados seja padronizada e validada antes do processamento.

REFERÊNCIAS

AGGARWAL, C. C. **Data Mining: The Textbook**. [*S.l.: s.n.*]: Springer, 2015. ISBN 978-3-319-14142-8.

Amazon. **Amazon Comprehend**. 2017. Acessado em 23 de dezembro de 2023. Available at: <https://aws.amazon.com/pt/comprehend/?nc2=h_ql_prod_ml_comp>.

BROWN, T. B. *et al.* Language models are few-shot learners. 2020.

CHAPMAN, P. *et al.* **CRISP-DM 1.0: Step-by-step data mining guide**. [*S.l.*], 2000.

CNJ. **Justiça em Números 2024**. 2023. <https://www.cnj.jus.br/wp-content/uploads/2024/05/justica-em-numeros-2024.pdf>. Último acesso em 19 de Agosto de 2024.

Comunicação Social TJSP. **TJSP registra 32,1 milhões de atos processuais entre janeiro e outubro deste ano. <https://www.tjsp.jus.br/Noticias/Noticia?codigoNoticia=95464&pagina=1>**. 2023. Último acesso em 23 de Dezembro de 2023.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, 1995.

DAVENPORT, T. H.; HARRIS, J. G. **Competing on Analytics: The New Science of Winning**. [*S.l.: s.n.*]: Harvard Business Press, 2007.

DEVLIN, J. *et al.* BERT: pre-training of deep bidirectional transformers for language understanding. **CoRR**, abs/1810.04805, 2018. Available at: <http://arxiv.org/abs/1810.04805>.

DEVLIN, J. *et al.* Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 2. ed. [*S.l.: s.n.*]: O'Reilly Media, 2019.

Google Cloud. **Natural Language AI**. 2016. Acessado em 23 de dezembro de 2023. Available at: <https://cloud.google.com/natural-language?hl=pt_br>.

Google Cloud. **AutoML on Vertex AI**. 2018. Acessado em 23 de dezembro de 2023. Available at: <https://cloud.google.com/automl?hl=pt_br>.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. [*S.l.: s.n.*]: Draft version, 2020.

KUMAR, V. *et al.* Resource-aware machine learning: A manual for data scientists. **arXiv preprint arXiv:1611.08326**, 2016.

KURGAN, L. A.; MUSILEK, P. A survey of knowledge discovery and data mining process models. **The Knowledge Engineering Review**, v. 21, n. 1, p. 1–24, 2006.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. **31st International Conference on Machine Learning, ICML 2014**, v. 4, 05 2014.

MARTÍNEZ-PLUMED, F. *et al.* Crisp-dm twenty years later: From data mining processes to data science trajectories. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 33, n. 8, p. 3048–3061, 2019.

MASALA, M. *et al.* jurbert: A romanian bert model for legal judgement prediction. *In*: **Proceedings of the Natural Legal Language Processing Workshop 2021**. [*S.l.: s.n.*], 2021. p. 86–94.

MIKOLOV, T. *et al.* Efficient estimation of word representations in vector space. 2013.

OpenAI. **ChatGPT**. 2020. Acessado em 23 de dezembro de 2023. Available at: <https://openai.com/chatgpt>.

OPENAI. **GPT-4 Technical Report**. 2023. <https://cdn.openai.com/papers/gpt-4.pdf>.

Painel CNJ. **Estatísticas do Poder Judiciário. <https://painel-estatistica.stg.cloud.cnj.jus.br/estatisticas.html>**. 2023. Último acesso em 23 de Dezembro de 2023.

PEKER, S.; KART, Ö. Transactional data-based customer segmentation applying crisp-dm methodology: A systematic review. **Journal of Data, Information and Management**, Springer, v. 5, n. 1, p. 1–21, 2023.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. *In*: **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. [*S.l.: s.n.*], 2014.

POLO, F. M. *et al.* Legalnlp - natural language processing methods for the brazilian legal language. **CoRR**, abs/2110.15709, 2021. Available at: <https://arxiv.org/abs/2110.15709>.

POLO, F. M. *et al.* Legalnlp-natural language processing methods for the brazilian legal language. *In*: SBC. **Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional**. [*S.l.: s.n.*], 2021. p. 763–774.

REIMERS, N.; GUREVYCH, I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *In*: **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**. [*S.l.: s.n.*], 2019. p. 3982–3992. Available at: <https://www.aclweb.org/anthology/D19-1410>.

REZENDE, S. **Sistemas inteligentes: fundamentos e aplicações**. Manole, 2003. ISBN 9788520416839. Available at: <https://books.google.com.br/books?id=UsJe_PlbnWcC>.

ROSSI, R. G. **Classificação automática de textos por meio de aprendizado de máquina baseado em redes**. 2015. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2015. Tese (Doutorado em Ciências de Computação e Matemática Computacional).

SALTON, G.; MCGILL, M. J. **Introduction to Modern Information Retrieval**. [*S.l.: s.n.*]: McGraw-Hill, 1986.

SCHRÖER, C.; KRUSE, F.; GÓMEZ, J. M. A systematic literature review on applying crisp-dm process model. **Procedia Computer Science**, Elsevier, v. 181, p. 526–534, 2021.

SCULLEY, D. *et al.* Machine learning: The high-interest credit card of technical debt. *In*: **SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop)**. [*S.l.: s.n.*], 2015.

SILVEIRA, R. *et al.* Legalbert-pt: A pretrained language model for the brazilian portuguese legal domain. *In*: SPRINGER. **Brazilian Conference on Intelligent Systems**. [*S.l.: s.n.*], 2023. p. 268–282.

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

SUN, C. *et al.* How to fine-tune bert for text classification? **arXiv preprint arXiv:1905.05583**, 2019.

TOUVRON, H. *et al.* Llama: Open and efficient foundation language models. **arXiv preprint arXiv:2302.13971**, 2023.

VASWANI, A. *et al.* Attention is all you need. *In*: **Advances in Neural Information Processing Systems**. [*S.l.: s.n.*], 2017. p. 5998–6008.

VIEGAS, C. F. O.; COSTA, B. C.; ISHII, R. P. JurisBERT: A new approach that converts a classification corpus into an STS one. *In*: **Computational Science and Its Applications – ICCSA 2023**. Springer Nature Switzerland, 2023. p. 349–365. Available at: <https://doi.org/10.1007%2F978-3-031-36805-9_24>.

ZAHARIA, M. *et al.* Accelerating the machine learning lifecycle with mlflow. **Databricks**, 2018.