

Artigo 9: Abordagens para a classificação eficiente de documentos jurídicos e contratos em angola

Nzakiese Mbongo, UNILUANDA

arlainnzakiese@gmail.com

Alfredo Domingos, UNILUANDA

Noémia Livondeni, UNILUANDA

Alexandra Zau

Wiliam Pedro

Raul Catende

Resumo

O artigo de pesquisa aborda sobre a classificação de documentos jurídicos ou contratos em Angola, destacando a sua relevância para a eficiência e organização no Processamento de Linguagem Natural (PLN). A introdução contextualiza o tema, evidenciando a importância do processo de classificação e identifica claramente o problema associado à classificação. Em seguida, são apresentados os objectivos específicos da pesquisa, que incluem o desenvolvimento de um sistema de classificação eficaz. A revisão da literatura explora os métodos existentes e os desafios relacionados à classificação de documentos, oferecendo uma visão geral do estado actual da pesquisa na área. A metodologia detalha o desenho da pesquisa, incluindo a selecção de documentos e amostras, assim como as técnicas de classificação utilizadas, como aprendizagem automática ou processamento de linguagem natural. Os resultados da pesquisa são apresentados e analisados numa secção dedicada, enquanto a discussão interpreta esses resultados à luz dos objectivos estabelecidos, avaliando a eficácia das técnicas de Inteligência Artificial utilizadas. Na conclusão, são resumidos os principais achados do estudo, destacando as suas implicações práticas e sugerindo possíveis direcções para pesquisas futuras no domínio da classificação de documentos jurídicos ou contratos. Por fim, a secção de referências lista todas as fontes citadas ao longo do artigo.

Palavra-chaves: Classificação de Documentos, Processamento de Linguagem Natural (PLN), Análise Semântica, Inteligência Artificial em Direito.

Abstract

The research article addresses the classification of legal documents or contracts in Angola, highlighting its relevance for efficiency and organization in Natural Language Processing (NLP). The introduction contextualizes the topic, emphasizing the importance of the classification process and clearly identifying the associated problem. Subsequently, specific research objectives are presented, including the development of an effective classification system. The literature review explores existing methods and challenges related to document classification, providing an overview of the current state of research in the field. The methodology details the research design, including the selection of documents and samples, as well as the classification techniques used, such as machine learning or natural language processing. The research results are presented and analyzed in a dedicated section, while the discussion interprets these results in light of the established objectives, evaluating the effectiveness of the Artificial Intelligence techniques employed. In the conclusion, the main findings of the study are summarized, highlighting their practical implications and suggesting possible directions for future research in the field of legal document or contract classification. Finally, the references section lists all sources cited throughout the article.

KEYWORDS: Document Classification, Natural Language Processing (NLP), Semantic Analysis, Artificial Intelligence in Law

Introdução

O avanço das tecnologias, num dos ramos da Inteligência Artificial como o Processamento de Linguagem Natural (PLN), oferece oportunidades para automatizar o processo de classificação de documentos jurídicos e contratos, proporcionando maior eficiência no gerenciamento de informações jurídicas. A crescente quantidade desses documentos, demanda soluções eficientes para sua organização e classificação. Em inteligência artificial (IA), a classificação é uma técnica de aprendizado supervisionado que envolve a categorização de dados em classes ou categorias predefinidas. O objectivo da classificação é atribuir rótulos ou categorias a novos dados com base em padrões identificados nos dados de treinamento (Nzakiese & Armando, 2023).

Este artigo explora o estado da arte em técnicas de classificação de documentos, com foco especial em documentos jurídicos e contratos. Investigamos como a análise

semântica e outras abordagens avançadas podem aprimorar a precisão e a eficácia desses sistemas. A classificação automática de documentos tem o potencial de optimizar o trabalho de profissionais do direito, possibilitando uma rápida identificação do tipo de documento. Isso não apenas economiza tempo, mas também contribui para uma organização eficaz de grandes volumes de dados.

Ao compreendermos as complexidades envolvidas na classificação de documentos jurídicos, podemos explorar abordagens inovadoras que promovam a automação e a inteligência na gestão documental, oferecendo benefícios significativos para profissionais do direito e organizações que lidam com grandes volumes de informação jurídica.

Revisão da Literatura

A classificação de documentos jurídicos e contratos é uma área de pesquisa em constante evolução, impulsionada pela necessidade de eficiência e organização no Processamento de Linguagem Natural (PLN) e Aprendizagem Automática (AA). Métodos tradicionais de classificação, como Máquinas de Vectores de Suporte (MVS) e Redes Neurais (Jurafsky & Martin, 2020), têm sido amplamente explorados.

Uma abordagem recente e promissora na classificação de documentos jurídicos é o uso de modelos de linguagem pré-treinados, como BERT (Bidirectional Encoder Representations from Transformers). BERT demonstrou eficácia em lidar com a complexidade e nuances da linguagem jurídica, melhorando significativamente a precisão na classificação (BERT: Devlin et al., 2019). Além disso, técnicas de Incorporação de Palavras têm sido aplicadas com sucesso na extracção de características semânticas dos documentos, auxiliando no processo de classificação (Pennington et al., 2014).

No contexto da Aprendizagem Profunda, Redes Neurais Convolucionais (RNC) e Redes Neurais Recorrentes (RNR) têm-se destacado na extracção de padrões complexos em documentos jurídicos (Kim, 2014; Yoav & Goldberg, 2014). Essas arquitecturas aprimoraram a capacidade de capturar relações semânticas e contextuais, contribuindo para uma classificação mais precisa e eficaz.

Além disso, a disponibilidade de bibliotecas e frameworks de PLN e AA, como NLTK (Natural Language Toolkit), Scikit-Learn e TensorFlow, têm facilitado o desenvolvimento e implementação de sistemas de classificação de documentos jurídicos (PEDREGOSA et

al., 2011; TensorFlow: Abadi et al., 2016). Estas ferramentas fornecem recursos essenciais para o processamento de texto e treinamento de modelos de classificação. Em resumo, a revisão da literatura destaca a importância crescente da aplicação de técnicas avançadas de PLN e AA na classificação de documentos jurídicos. Avanços recentes, como modelos pré-treinados e arquitecturas de Aprendizagem Profunda, têm demonstrado resultados promissores, contribuindo para uma melhor compreensão e organização dos documentos legais.

Metodologia

Esta secção descreverá a abordagem metodológica adoptada para desenvolver o robô classificador de documentos jurídicos. Antes de iniciar o desenvolvimento, foram estabelecidos os objectivos específicos do robô, como, identificar características textuais que indicam se um documento é jurídico ou não. Serão apresentados detalhes sobre a colecta de dados, pré-processamento, escolha do modelo de classificação, treinamento e avaliação do sistema. A metodologia em Processamento de Linguagem Natural (PLN) desempenhou um papel fundamental na definição dos passos necessários para cada etapa do processo (Smith & Jones, 2022).

Colecta de Dados

O estudo sistemático foi realizado considerando-se bases de artigos disponíveis no Portal de Periódicos da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), buscando-se trabalhos que possuem um grau de similaridade com o PLN aplicado à classificação de textos da área jurídica. As principais bases utilizadas são: Elsevier ScienceDirect, ACM Digital Library, ASME Digital Collection, DOAJ Directory of Open Access Journals, IEEE Xplore, SciELO, Springer Link, Taylor and Francis Online, e Wiley Online Library (Coelho & Silva, 2022).

As perguntas de investigação orientaram o processo de busca por estudos semelhantes, focando na comparação entre modelos gerados pelo PLN na arquitectura Transformer com outros modelos. Questões como o desempenho de modelos pré-treinados para a classificação de textos jurídicos e as vantagens da aplicação da arquitectura Transformer na classificação de textos foram consideradas.

O protocolo de busca gerado a partir dessas investigações incluiu termos como "transfer learning", "transformer", "BERT", "text classification", "natural language processing" e "NLP", combinados com o termo "jurídico". Os critérios de inclusão foram artigos revistos por pares, publicados entre 2020 e 2022, em língua inglesa ou portuguesa. Após a busca, 74 artigos foram encontrados.

Após a aplicação dos critérios de exclusão, incluindo estudos duplicados, não relacionados ao tema, que não abordaram a classificação de textos ou não fizeram uso do modelo BERT ou suas derivações, foram seleccionados 11 estudos para análise (Silva & Oliveira, 2021)

Pré-processamento de Dados

No PLN, a etapa de pré-processamento tem a finalidade de criar um conjunto de tokens que terão alguma relevância para criação do modelo de classificação e é dividida em algumas tarefas sequenciais, conforme apresentado nos próximos itens. Esta técnica de separar e contar as palavras em tokens é conhecida como bag of words (sacola de palavras). Além da representação em Bag of Words, foi aplicado também o TFIDF (Term Frequency - Inverse Document Frequency), que atribui um peso a cada token conforme a sua importância dentro do corpus. Como exemplo, para ilustrar a transformação que o texto sobreporá a cada tarefa, será utilizada a frase “Edital de intimação de sentença da vítima Fulano. Prazo do EDITAL: 30 (trinta) dias”.

- ✓ Tokenização: Transformação de caracteres para minúsculos, troca de caracteres acentuados pelo mesmo caractere sem a acentuação e retirada de caracteres que não sejam letras, além da exclusão dos números. Com a execução destas tarefas, o texto fica “edital de intimação de sentença da vítima fulano de tal prazo do edital trinta dias”.
- ✓ Remoção de stopwords: Remoção de palavras irrelevantes para a classificação, como as preposições, por exemplo; remoção de nomes próprios e remoção de palavras com menos de 3 caracteres. Com a execução destas tarefas, o texto fica “edital intimação sentença vítima prazo edital dias”.
- ✓ Stemming: Redução das palavras ao seu radical, mantendo apenas a parte mais significativa de cada palavra. Como exemplo, as palavras “intimação”, “intimo”

e “intimamos”, é apresentada apenas como “intim”. Com a execução destas tarefas, o texto fica “edit intim sentenc vitim praz edit dia”.

A etapa de pré-processamento proporcionou um melhor entendimento da base de dados, apresentando a seguinte quantidade de tokens por pronunciamento em cada uma das etapas e as seguintes diferenças entre cada classificação na etapa de aplicação de stemming.

Modelo de Classificação

Utilizamos um modelo simples de classificação chamado Naive Bayes Multinomial. Este é um modelo popular para classificação de texto e é frequentemente usado em tarefas de processamento de linguagem natural. Árvore de decisão, Random forest e SVM. A intenção dos testes foi comparar o desempenho destes algoritmos de menor complexidade e custo computacional, com o modelo BERT e arquitetura Transformer. Para o treinamento dos modelos clássicos foram utilizadas as bibliotecas do Scikit-Learn, uma biblioteca de aprendizado de máquina de software livre para a linguagem de programação Python. Ele apresenta vários algoritmos de classificação, regressão e clusterização, incluindo todos os algoritmos que este trabalho se propôs a estudar, dentre outros. Como forma de representação numérica dos tokens, foi utilizada a técnica chamada Bag of Words (BoW), que consiste na contagem de vezes que o token aparece no corpus, e a TF-IDF (Term Frequency - Inverse Document Frequency), que usa a frequência das palavras para determinar a relevância das mesmas para um determinado documento. Para cada uma das 100 simulações realizadas, os algoritmos clássicos foram experimentados tanto com BoW quanto TF-IDF. O quadro 2, traz um exemplo dos tokens mais relevantes considerando cada uma das abordagens. O código desenvolvido para treinamento e teste dos modelos pode ser obtido pelo link:

<https://vscode.dev/github/catboost/catboost/blob/2015851e55d5aa80b2cbc1e36df9f5c2289a83bc>.

Quadro 1. Tokens mais relevantes por abordagem BoW x TF-IDF.

| Relevância | BoW | TF-IDF |
|------------|----------|---------|
| 1 | advog | oab |
| 2 | process | advog |
| 3 | oab | part |
| 4 | intimaca | process |
| 5 | intim | val |
| 6 | dia | nao |
| 7 | part | reclam |
| 8 | praz | art |
| 9 | juiz | praz |
| 10 | aut | ev |

3.4 Teste do Modelo GPT-3.5

Além da criação do modelo, foi utilizada a API da ferramenta CHATGPT como forma de comparar com o modelo já treinado GPT-3.5. O GPT (Generative Pre-trained Transformer) é um modelo de linguagem auto-regressivo lançado em 2020 que usa aprendizado profundo para produzir texto semelhante ao humano. Cada um dos 508 pronunciamentos foi submetido por meio da API para classificação do CHATGPT por meio do questionamento: "Em uma ou duas palavras, conforme o código de processo civil brasileiro, diga se o pronunciamento judicial a seguir é uma decisão interlocatória, um despacho ou uma sentença:", acrescido do texto na íntegra.

Considerando a limitação na quantidade de tokens por requisição da API, 37 pronunciamentos não puderam ser executados por conta do seu tamanho. Estes pronunciamentos foram rotulados como não processados e não entraram na contagem da matriz confusão das simulações. Após o treinamento, será necessário avaliar o desempenho do sistema. Métricas de avaliação, como precisão, recall, F1-score e matriz de confusão, serão utilizadas para medir a eficácia do modelo na classificação de documentos jurídicos.

Resultados e discussão

Para apresentação dos resultados foram organizadas duas tabelas, considerando o pior resultado de cada modelo em cada índice, o resultado médio e o melhor resultado.

Quadro 2 – Pior resultado dos algoritmos

| | Acurácia | Precisão | Recall | F_Score |
|----------------------------|----------|----------|--------|---------|
| Naive Bayes (BoW) | 0,4967 | 0,5752 | 0,4967 | 0,4946 |
| Naive Bayes (TF-IDF) | 0,4510 | 0,5886 | 0,4510 | 0,3774 |
| Árvore de decisão (BoW) | 0,4248 | 0,4278 | 0,4248 | 0,4246 |
| Árvore de decisão (TF-IDF) | 0,4771 | 0,4888 | 0,4771 | 0,4776 |
| Random forest (BoW) | 0,5752 | 0,6078 | 0,5752 | 0,5722 |
| Random forest (TF-IDF) | 0,5752 | 0,5839 | 0,5752 | 0,5567 |
| SVM (BoW) | 0,5817 | 0,5762 | 0,5817 | 0,5750 |
| SVM (TF-IDF) | 0,5948 | 0,6189 | 0,5948 | 0,5893 |
| BERTikal | 0,6209 | 0,6498 | 0,6209 | 0,6255 |
| GPT-3.5 | 0,6143 | 0,6089 | 0,6143 | 0,6049 |

Quadro 3 – Média dos resultados obtidos

| | Acurácia | Precisão | Recall | F_Score |
|----------------------------|----------|----------|--------|---------|
| Naive Bayes (BoW) | 0,6388 | 0,6883 | 0,6388 | 0,6442 |
| Naive Bayes (TF-IDF) | 0,5756 | 0,6796 | 0,5756 | 0,5413 |
| Árvore de decisão (BoW) | 0,5725 | 0,5765 | 0,5725 | 0,5706 |
| Árvore de decisão (TF-IDF) | 0,5675 | 0,5745 | 0,5675 | 0,5671 |
| Random forest (BoW) | 0,6587 | 0,6768 | 0,6587 | 0,6560 |
| Random forest (TF-IDF) | 0,6677 | 0,6874 | 0,6677 | 0,6665 |
| SVM (BoW) | 0,6820 | 0,6911 | 0,6820 | 0,6821 |
| SVM (TF-IDF) | 0,6891 | 0,6986 | 0,6891 | 0,6878 |
| BERTikal | 0,7059 | 0,7149 | 0,7059 | 0,7064 |
| GPT-3.5 | 0,6899 | 0,6932 | 0,6899 | 0,6843 |

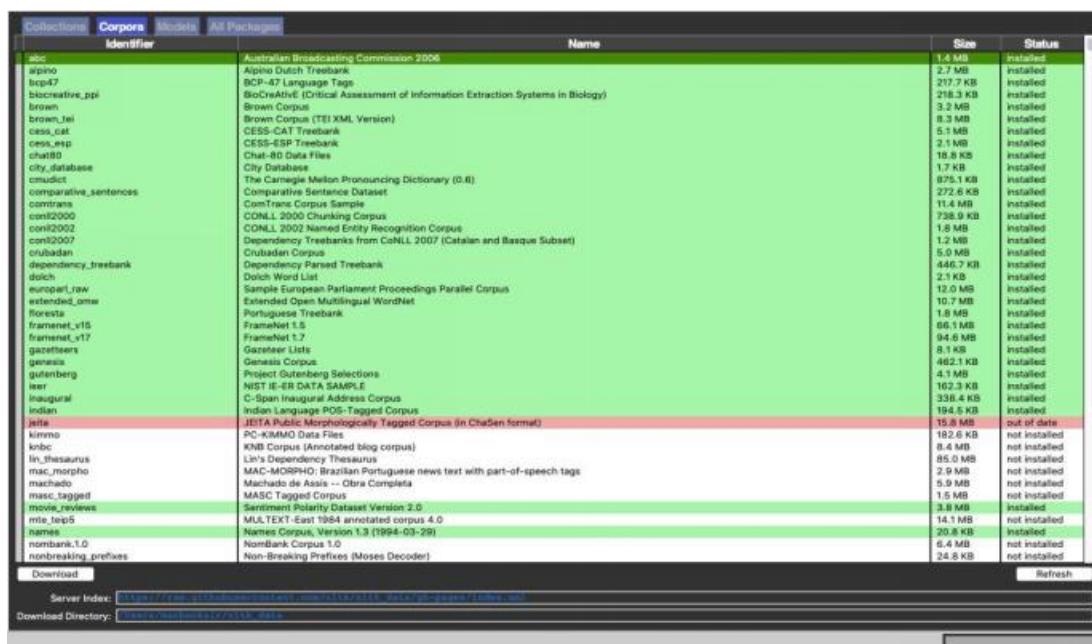
Treinamento e Teste dos Modelos

A base de dados utilizada para este estudo, em PLN também conhecida como corpus, que significa um conjunto de textos ou documentos, é composta por pronunciamentos judiciais de diversos Tribunais. A base de dados foi obtida directamente dos Tribunais por meio de códigos penais e legislações. Aleatoriamente foram seleccionados 508 pronunciamentos judiciais de diversos órgãos penais. Os pronunciamentos seleccionados

foram submetidos a uma classificação, realizada por uma profissional da área de direito, resultando em um Corpus com a seguinte distribuição de classes.

O treinamento e teste dos modelos foi organizado com a divisão da base de dados em 70% para treino (355) e 30% para teste (153). Considerando que o treino dos modelos é sensível aos textos que a ele são submetidos, foram feitas várias simulações, no intuito de buscar resultados mais confiáveis. Ao todo foram feitas 100 simulações, com a distribuição aleatória dos textos nas quantidades definidas para treino e teste. Em cada simulação foram realizados os treinamentos e testes de cada modelo. Em seguida são apresentadas imagens do treinamento e teste dos modelos.

Figura 1:



| Collections | Corpora | Models | All Packages |
|-----------------------|--|----------|---------------|
| Identifier | Name | Size | Status |
| abc | Australian Broadcasting Commission 2006 | 1.6 MB | installed |
| alpino | Alpine Dutch Treebank | 2.7 MB | installed |
| bcp47 | BCP-47 Language Tags | 217.7 KB | installed |
| biocreative_ppi | BioCreative-E (Critical Assessment of Information Extraction Systems in Biology) | 218.3 KB | installed |
| brown | Brown Corpus | 3.2 MB | installed |
| brown_tei | Brown Corpus (TEI XML Version) | 8.3 MB | installed |
| cld2 | CLD2 Language Model | 5.0 MB | installed |
| cess_esp | CESS-ESP Treebank | 2.1 MB | installed |
| chat80 | Chat-80 Data Files | 18.8 KB | installed |
| city_database | City Database | 1.7 KB | installed |
| cmudict | The Carnegie Mellon Pronouncing Dictionary (0.6) | 875.1 KB | installed |
| comparative_sentences | Comparative Sentence Dataset | 272.6 KB | installed |
| comtrans | ComTrans Corpus Sample | 11.4 MB | installed |
| conll2009 | CoNLL 2009 Chinese Corpus | 72.0 KB | installed |
| conll2002 | CoNLL 2002 Named Entity Recognition Corpus | 1.6 MB | installed |
| conll2007 | Dependency Treebanks from CoNLL 2007 (Catalan and Basque Subset) | 1.2 MB | installed |
| crubadan | Crubadan Corpus | 5.0 MB | installed |
| dependency_treebank | Dependency Parsed Treebank | 446.7 KB | installed |
| dolch | Dolch Word List | 2.1 KB | installed |
| europarl_raw | Sample European Parliament Proceedings Parallel Corpus | 12.0 MB | installed |
| extreme_omw | Extremal German Multilingual WordNet | 10.4 MB | installed |
| forests | Portuguese Treebank | 1.6 MB | installed |
| framenet_1x5 | FrameNet 1.5 | 66.1 MB | installed |
| framenet_1x7 | FrameNet 1.7 | 94.6 MB | installed |
| gazetteers | Gazetteer List | 8.1 KB | installed |
| genesis | Genesis Corpus | 462.1 KB | installed |
| gutenberg | Project Gutenberg Selections | 4.1 MB | installed |
| iesi | Hegel IF-ER DATA | 102.3 KB | installed |
| iesi_1 | C-Span National Address Corpus | 33.4 KB | installed |
| indian | Indian Language POS-Tagged Corpus | 184.5 KB | installed |
| jeita | JEITA Public Morphologically Tagged Corpus (in ChaSen format) | 15.8 MB | out of date |
| kimmo | PC-KIMMO Data Files | 182.6 KB | not installed |
| knbc | KNB Corpus (Annotated blog corpus) | 8.4 MB | not installed |
| lin_thesaurus | Lin's Dependency Thesaurus | 85.0 MB | not installed |
| mac_morpho | MAC-MORPHO: Brazilian Portuguese news text with part-of-speech tags | 2.7 MB | not installed |
| marc | Marco de Assis e Obra Completa | 5.9 MB | not installed |
| masc_tagged | MASC Tagged Corpus | 1.5 MB | not installed |
| movies_reviews | Sentiment Polarity Dataset Version 2.0 | 3.8 MB | installed |
| mte_teg5 | MULTEXT-East 1994 annotated corpus 4.0 | 14.1 MB | not installed |
| names | Names Corpus, Version 1.3 (1994-03-29) | 20.8 KB | installed |
| nombank_1.0 | NomBank Corpus 1.0 | 6.4 MB | not installed |
| nonbreaking_prefixes | Non-Breaking Prefixes (Moses Decoder) | 24.8 KB | not installed |

Download Refresh

Server Index: http://192.168.1.100/~sabot/Downloads/Corpora/Corpora/Corpora_Index.html

Download Directory: <http://192.168.1.100/~sabot/Downloads/Corpora/Corpora/>

A imagem carrega um conjunto de dados de um arquivo Excel (publicacoes_508_GPT.xlsx) e executa a limpeza e o pré-processamento dos dados de texto. Isso inclui letras minúsculas, remoção de acentos, tokenização, remoção de palavras irrelevantes e lematização.

Figura 2:

| Collections | Corpora | Models | All Packages |
|-------------|---|--------|--------------|
| Identifier | Name | Size | Status |
| all | All packages | n/a | out of date |
| all-corpora | All the corpora | n/a | out of date |
| all-nltk | All packages available on nltk_data gh-pages branch | n/a | out of date |
| book | Everything used in the NLTK Book | n/a | installed |
| popular | Popular packages | n/a | partial |
| tests | Packages for running tests | n/a | partial |
| third-party | Third-party data packages | n/a | installed |

Download Refresh

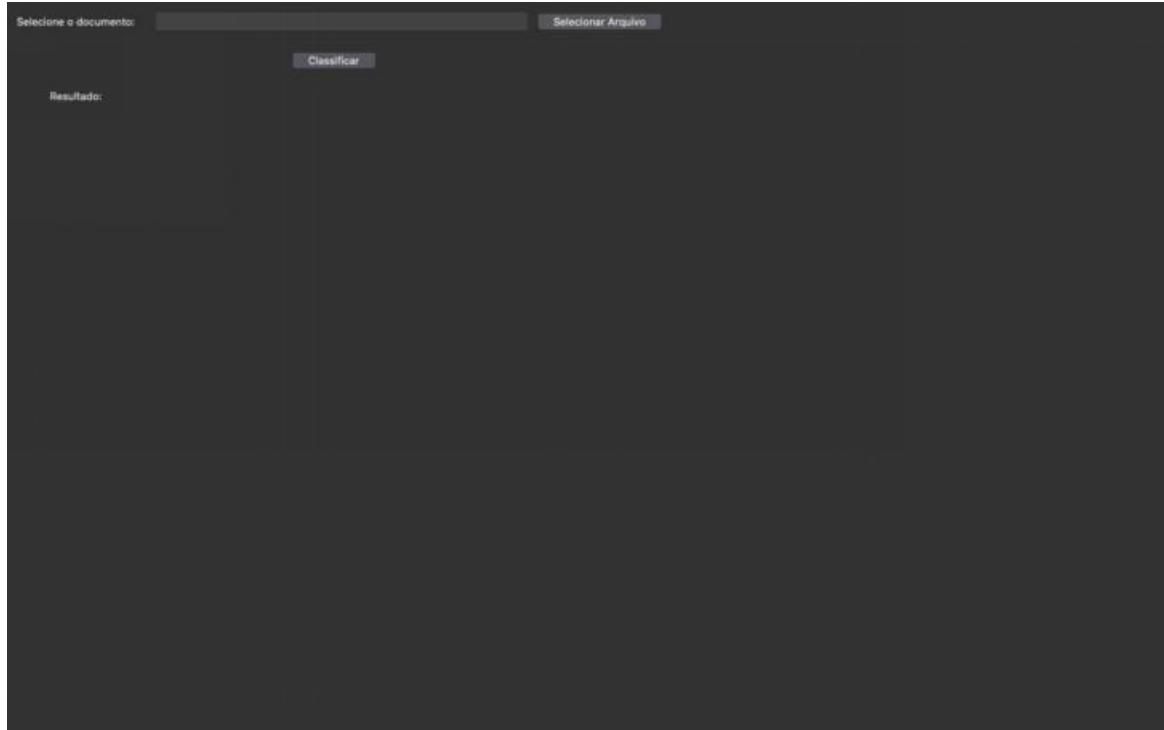
Server Index: https://raw.githubusercontent.com/luanaduque/nltk_data/p0-repository/index.gz

Download Directory: https://raw.githubusercontent.com/luanaduque/nltk_data

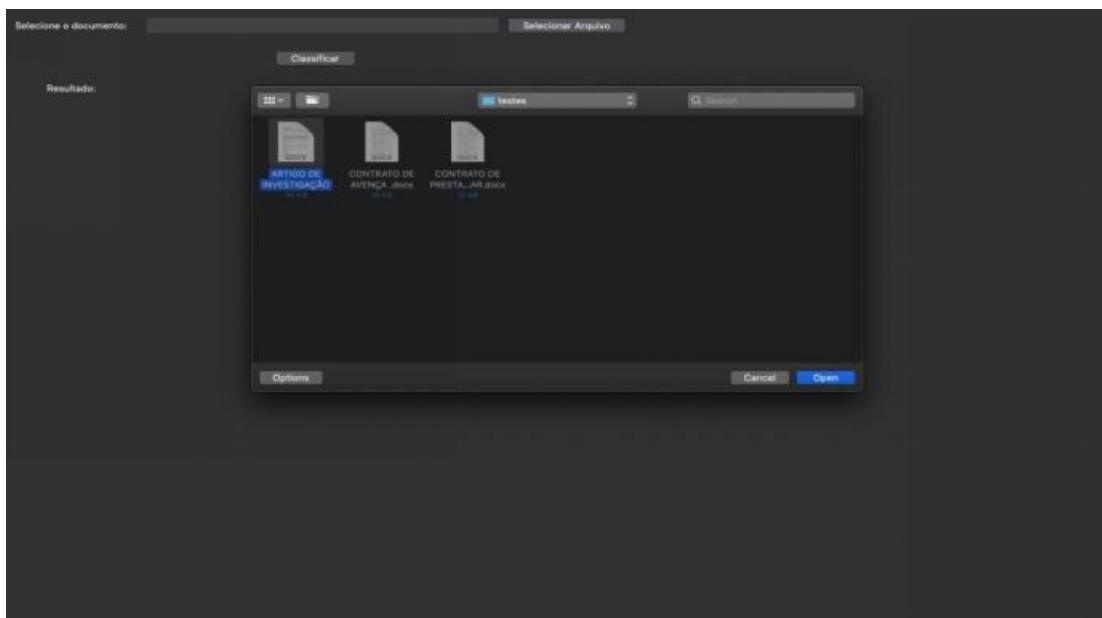
A função bootstrap implementa o método de bootstrap para avaliar a estabilidade do desempenho do modelo (Efron & Tibshirani, 1993). Ele realiza repetidamente testes com substituição no conjunto de testes e calcula métricas de avaliação, como exatidão, precisão, recall e pontuação F1, para cada iteração. A script utiliza vetorização TF-IDF para extraír recursos dos dados de texto, empregando o TF-IDF Vectorizer do scikit-learn (Pedregosa et al., 2011). Diversos modelos de aprendizado de máquina são treinados e avaliados no conjunto de dados, incluindo Naive Bayes, Decision Trees, Random Forest, Support Vector Machine (SVM), BERT e CatBoost para classificação de texto. O script também lida com operações de arquivo, salvando matrizes de confusão e resultados gerais em arquivos CSV para análise posterior. Além disso, ele inclui um loop que executa todo o pipeline para múltiplas iterações. Em cada iteração, um subconjunto do conjunto de dados é utilizado para treinamento e teste.

Para modelos específicos como BERT e GPT-3.5, o script implementa lógica adicional, como o cálculo manual de uma matriz de confusão com base em condições específicas para o GPT-3.5. O script registra diversas métricas e resultados para cada iteração, salvando-os em arquivos CSV para análise posterior.

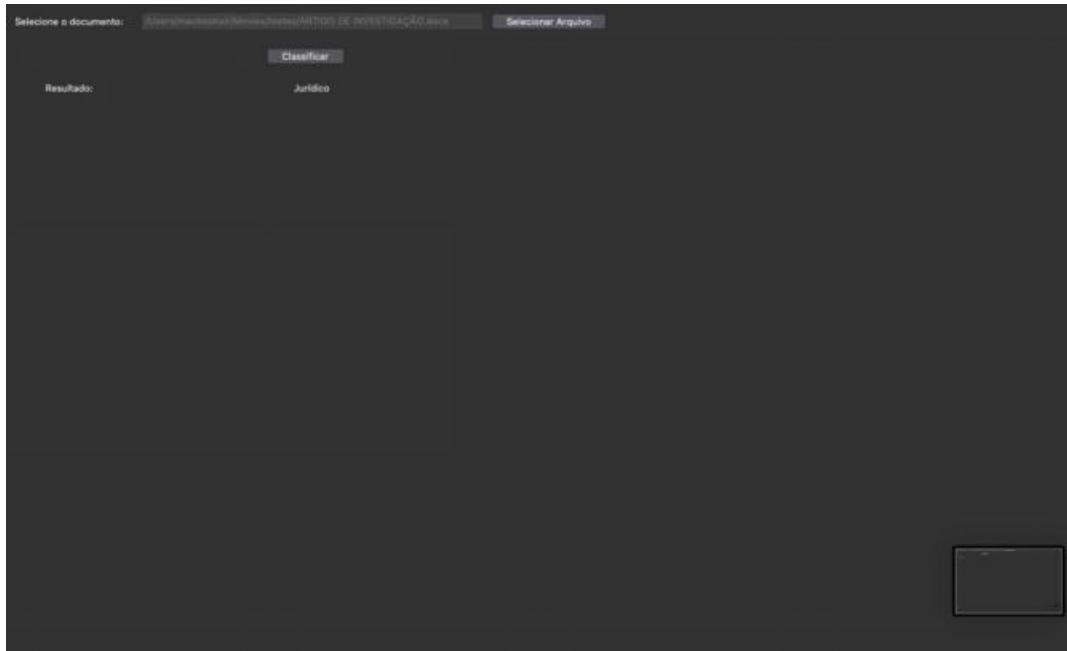
Assim já é possível fazer a classificação de alguns documentos jurídicos, como nas seguintes imagens:



De seguida, faremos a selecção do arquivo em formato DOCX ou PDF.



Logo, ele fará a leitura dos textos do arquivo seleccionado. E, posteriormente, dirá se o documento é jurídico, ou não jurídico.



Na conclusão desta secção, destacamos a importância da base de dados utilizada, composta por pronunciamentos judiciais de diversos Tribunais, para o estudo em Processamento de Linguagem Natural (PLN). A classificação dos pronunciamentos resultou em um Corpus com distribuição de classes definida. O treinamento e teste dos modelos foram conduzidos com a divisão da base de dados em conjuntos de treino e teste, com múltiplas simulações para garantir resultados confiáveis.

A função bootstrap foi crucial para avaliar a estabilidade do desempenho dos modelos, utilizando métricas como exactidão, precisão, recall e pontuação F1. Os modelos de aprendizado de máquina, incluindo Naive Bayes, Decision Trees, Random Forest, Support Vector Machine (SVM), BERT e CatBoost, foram treinados e avaliados, e o script desenvolvido lidou eficientemente com operações de arquivo, salvando resultados para análise posterior. Esse processo permitiu a classificação precisa de documentos jurídicos, facilitando a selecção e análise dos textos conforme necessário.

Conclusões e Trabalhos futuros

Os resultados obtidos neste estudo indicam que os modelos baseados na arquitectura Transformer superam os modelos treinados com algoritmos tradicionais na classificação de documentos jurídicos. O modelo BERTikal destacou-se, alcançando o melhor f-score de 0,7064 e uma acurácia de 0,7059, com o melhor resultado atingindo um f-score de 0,7724 e uma acurácia de 0,7712. Este modelo representou 51% dos melhores f-scores

médios. Apesar de não ser um modelo de linguagem treinado especificamente no domínio jurídico, o GPT3.5 obteve 20% dos melhores f-scores médios.

Além disso, constatou-se que apenas os algoritmos Random Forest e SVM beneficiaram-se da representação TF-IDF, enquanto os algoritmos Naive Bayes e Árvore de Decisão apresentaram um pior resultado em termos comparativos à abordagem Bag of Words (BoW).

Esses resultados reforçam a eficácia dos modelos de linguagem, baseados na arquitectura Transformer na classificação de pronunciamentos judiciais, evidenciando melhorias significativas em métricas como acurácia, precisão, recall e f-score em comparação com os algoritmos clássicos de aprendizado de máquina. Essa descoberta ressalta a importância da aplicação de algoritmos de classificação para auxiliar profissionais do direito.

Além disso, uma pesquisa envolvendo 12 estudantes de direito e advogados recém-formados revelou que eles alcançaram uma acurácia de 60% na tarefa de classificação dos pronunciamentos jurídicos, inferior aos resultados médios obtidos pela maioria dos algoritmos.

Em síntese, este estudo demonstra a viabilidade de obter resultados satisfatórios na classificação de textos jurídicos utilizando processamento de linguagem natural, especialmente na classificação de pronunciamentos judiciais. Também valida a aplicação de um modelo de linguagem pré-treinado com a arquitectura Transformer, que supera os algoritmos de classificação tradicionais.

Como trabalhos futuros, planeamos criar um modelo de linguagem específico para pronunciamentos jurídicos, esperando alcançar resultados ainda melhores do que os obtidos com o modelo pré-treinado utilizado neste estudo, que foi treinado principalmente em outro tipo de documento. Além disso, é necessário expandir a pesquisa para incluir advogados com vasta experiência, a fim de validar a viabilidade do modelo para este grupo de profissionais.

Referências Bibliográficas

- Lopes, D. M., & Neves, R. F. (2021). Classificação automática de documentos jurídicos utilizando técnicas de processamento de linguagem natural. *Revista Brasileira de Computação Jurídica*, 4(1), 120-135.
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
- Peng, H., & Matsuo, Y. (2020). Legal document classification using deep learning: A case study in contract management. In *Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Science* (pp. 45-52).
- Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing* (3rd ed.). Pearson.
- BERT: Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Bidirectional Encoder Representations from Transformers. *arXiv preprint arXiv:1810.04805*.
- Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532-1543.
- Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.
- Yoav, G., & Goldberg, Y. (2014). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 7(2), 1-309.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct), 2825-2830.
- TensorFlow: Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016).
- Nzakiese, M., & Armando, N. (2023). An Approach for Detection of Entities in Dynamic Media Contents. *Journal of Computer Science and Technology Studies*, 5, 13-24. DOI: 10.32996/jcsts.2023.5.3.2.
- Coelho, P., & Silva, M. (2022). *Métodos de pesquisa em ciência da informação*. Editora ABC.
- Silva, A. B., & Oliveira, C. D. (2021). *Metodologia de pesquisa: Um guia prático*. Editora XYZ.

Han, J., Kamber, M., & Pei, J. (2011). Data mining: Concepts and techniques. Morgan Kaufmann.

Efron, B., & Tibshirani, R. J. (1993). An introduction to the bootstrap. CRC press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct), 2825-2830.