UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Disciplina: Processamento de Linguagem Natural
Docente: Arlindo Rodrigues Galvão Filho
Discentes: Héber Júnior Rodrigues Amaral - 202005483
Marcos Vinicius Satil Medeiros - 202005497
Pedro Augusto De Almeida Mattos - 202005499
Pedro Koziel Diniz - 202005500

CLASSIFICAÇÃO DE TEXTOS JURÍDICOS COM DIFERENTES REPRESENTAÇÕES

INTRODUÇÃO

Este trabalho foi desenvolvido tendo como base o projeto "Classificação de Documentos e Extração Inteligente de Informações em Textos Jurídicos", realizado em parceria entre o CEIA e a empresa COREJUR. O projeto inteiro aborda também outras frentes, mas para esse trabalho final iremos dar ênfase às atividades de classificação de texto.

A classificação de texto é uma tarefa importante na área de processamento de linguagem natural (NLP), especialmente em aplicações jurídicas, onde a precisão e a confiabilidade são críticas. A representação do texto é uma parte fundamental da classificação, pois afeta a capacidade do modelo de capturar e utilizar as características relevantes do texto.

No estágio atual do projeto está sendo utilizado um modelo BERT pré-treinado, em que é realizado fine-tuning em cima dos dados do projeto, a fim de gerar um modelo para realizar a classificação de textos jurídicos. A fim de expandir essa visão, nesse trabalho será apresentado também outras formas de representação de palavras, com o objetivo de realizar um comparativo entre os resultados obtidos, utilizando o mesmo dataset, na task de classificação. As métricas analisadas foram o tempo de execução e F1, e as representações experimentadas foram TF-IDF, Word2Vec (pré-treinado e gerado a partir dos próprios dados) e transformers-based (BERT).

DADOS

O conjunto de dados utilizado é gerado durante o andamento do projeto, de modo que é utilizado uma ferramenta de extração para extrair os textos dos documentos escaneados, e depois esses textos são passados para a ferramenta de anotação, em que profissionais da área jurídica realizam a anotação desses dados, colocando a devida classe de cada texto. São utilizados documentos jurídicos, mais especificamente da área bancária, por isso a necessidade de profissionais da área para realizar a anotação.

Como os textos originais possuem um tamanho grande, foi optado por realizar a anotação por parágrafo dos documentos. Também é feita a anotação para geração de um conjunto de dados da task de Extração de Entidade Nomeada (NER), mas iremos utilizar apenas o conjunto de dados para Classificação.

No final, o que é recebido para classificação é um dataset em formato csv, em que as colunas que são utilizadas para o treinamento dos modelos são as “text”, que contêm os textos dos parágrafos, e “label”, que contêm a respectiva classe a que o texto pertence.

Como pré-processamento dos textos, em todas as representações foram removidas stopwords, símbolos, pontuação e acentos, além de transformar todas as letras em minúsculas. A exceção foi na representação Word2Vec com um modelo Skip-Gram pré-treinado, em que foi utilizado o mesmo pré-processamento utilizado para o treino desse modelo, que foi desenvolvido pelo NILC.

METODOLOGIA

Para cada tipo de representação foram realizados o treino e testes na task de classificação, buscando manter iguais o dataset utilizado, a divisão entre treino e teste (na proporção 80/20), e a seed utilizada (17).

Para o TF-IDF, após a geração da representação numérica foi utilizado o classificador Random Forest para realizar a classificação, visto que entre os classificadores testados esse foi o que gerou melhor resultado. Os parâmetros utilizados para gerar a representação e no classificador foram os padrões da biblioteca sklearn.

Para o Word2Vec, primeiro foi testado o uso de uma representação pré-treinada, disponibilizada pelo NILC (disponível em: http://143.107.183.175:22980/download.php?file=embeddings/word2vec/skip_s300.zip). Para isso, o pré-processamento utilizado foi adequado para ser igual àquele modelo pré-treinado. Foi utilizado um modelo Skip-Gram de 300 dimensões.

Também foi realizado o treinamento de representações Word2Vec com base nos próprios dados de treino do dataset, também com a estratégia Skip-Gram, a fim de

analisar o desempenho de cada um. Em ambos os modelos Word2Vec, após a obtenção dos embeddings foi utilizado o classificador RandomForest para a task de classificação, visto que foi o que gerou melhores resultados dentre os classificadores testados.

Por fim, foi utilizado a representação Transformers, com o modelo BERT (BERTimbau Base). O tamanho máximo das sequências foi reduzido para 128, e o batch_size para 16, devido aos custos computacionais. Foi utilizado o optimizador AdamW, com taxa de aprendizado de 5e-5, e os outros hiperparâmetros foram mantidos na configuração padrão, assim como nas representações anteriores.

RESULTADOS

Após a execução dos algoritmos, foram obtidos os seguintes resultados:

	TF-IDF	Word2Vec pré-treinado	Word2vec no dataset	BERT
F1	0.81	0.65	0.77	0.86
Tempo	4m 3s	5m 52s	6m 5s	1h20min
Processamento	Processador	Processador	Processador	GPU

Percebe-se que mesmo o TF-IDF sendo uma representação mais simples do que o Word2Vec os resultados gerados foram melhores, o que pode indicar a presença de palavras que facilitam a distinção entre diferentes classes.

Também é visto que o Word2Vec pré-treinado foi inferior ao Word2Vec treinado em cima dos dados de treino do dataset, provavelmente devido ao fato que os dados utilizados são do contexto jurídico, então uma representação treinado já em cima desse domínio, mesmo que com menos dados, pode gerar melhores resultados.

Por fim, percebe-se que o modelo BERT foi o que obteve melhor resultado, porém seu tempo de execução foi bem maior do que as outras representações.

CONCLUSÃO

A representação numérica de textos é crucial para o sucesso da classificação de textos, e existem diversas formas de realizá-la. Neste trabalho foram testadas 3 representações distintas, TF-IDF, Word2Vec e Transformers (BERT), com o objetivo de analisar o desempenho de cada uma.

O modelo transformers foi o que gerou melhor resultado (F1 score), o que era esperado devido ao fato de possuir uma arquitetura mais complexa do que as outras representações. Porém, o tempo e recursos computacionais necessários para sua

utilização foram maiores, o que para esse projeto não é um problema, mas que em outras situações pode não ser desejado.

Por outro lado, a representação TF-IDF performou melhor do que o Word2Vec, provavelmente por causa de, na task de classificação, algumas palavras serem melhores indicativas de determinadas classes.

Dessa forma, observa-se que cada representação possui tanto pontos positivos quanto negativos, sendo necessário analisar as necessidades e situação para cada aplicação. No caso deste trabalho, a representação que gerou melhor resultado foi utilizando o transformers