UNIVERSIDADE FEDERAL DO MARANHÃO

Curso de Ciência da Computação

Pedro Vinnícius Bernhard

Estudo Comparativo de Large Language Models aplicados à classificação de documentos de Prestação de Contas Públicas

São Luís

2023

Pedro Vinnícius Bernhard

Estudo Comparativo de Large Language Models aplicados à classificação de documentos de Prestação de Contas Públicas

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. João Dallyson Sousa de Almeida

São Luís

2023

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Bernhard, Pedro Vinnícius.
Estudo Comparativo de Large Language Models aplicados à classificação de documentos de Prestação de Contas Públicas / Pedro Vinnícius Bernhard. - 2023.
55 f.

Orientador(a): João Dallyson Sousa de Almeida.
Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luís - MA, 2023.

1. Classificação de Documentos. 2. Large Language Models. 3. Prestação de Contas. 4. Processamento de Linguagem Natural. 5. TCE/MA. I. Almeida, João Dallyson Sousa de. II. Título.

Pedro Vinnícius Bernhard

Estudo Comparativo de Large Language Models aplicados à classificação de documentos de Prestação de Contas Públicas

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho _____ em São Luís, 19 de dezembro de 2023:

Prof. Dr. João Dallyson Sousa de Almeida
Orientador

Prof. Dr. Anselmo Cardoso de Paiva
Examinador

Prof. Dr. Darlan Bruno Pontes Quintanilha
Examinador

São Luís
2023

Ao meu pai,
Pedro Luiz Bernhard (In Memoriam),
que foi sempre um exemplo de
inteligência, tranquilidade e calma.
Espero um dia estar à sua altura.

Agradecimentos

Agradeço à minha noiva, Rosália, que sempre esteve ao meu lado, me incentivando, ajudando e oferecendo suporte emocional nos momentos mais difíceis, expresso meu amor profundo.

À minha mãe, expresso meus sinceros agradecimentos pelo carinho e apoio constantes, fundamentais na construção de minha jornada.

Aos meus filhos de quatro patas, Gatinha, Miu e Nikki, que me trazem alegrias diárias.

Aos meus sogros Olímpia e Orlando, e cunhada Lya, sempre disponíveis para me ajudar, agradeço de coração.

Ao meu orientador, João Dallyson, pela sua orientação, assistência e sugestões que foram essenciais para o desenvolvimento deste trabalho.

Aos meus amigos e colegas, agradeço pelas discussões produtivas e troca de ideias. Destaco aqui meus bons amigos Estephane, Igor, João, Mário, Saulo, Victor e todos do NCA, a quem tenho grande respeito.

Agradeço aos professores Anselmo, Aristófanes, Darlan, Geraldo, Glaubos, Italo, Portela, Rivero, Salles, Simara, e todos os docentes do NCA, figuras de grande importância em minha experiência acadêmica.

Meus agradecimentos ao Tribunal de Contas do Estado do Maranhão, pela essencial contribuição na disponibilização dos documentos, sem os quais esta pesquisa não teria sido concretizada.

A todos, meu muito obrigado. Agradeço a todos pela contribuição fundamental que tornou possível a realização deste trabalho.

"O futuro já está aqui – só não está uniformemente muito bem distribuído."

William Gibson, em *"The Economist"*

Resumo

O Tribunal de Contas do Estado do Maranhão (TCE/MA) desempenha um papel essencial no controle das contas públicas, utilizando o Sistema de Prestação de Contas Anual eletrônica (e-PCA). Nesse processo, os fiscalizados enviam documentos eletrônicos relacionados às prestações de contas de governo e de gestores, classificados conforme normativas estabelecidas. É importante, portanto, a correta classificação dos documentos para assegurar a conformidade com os padrões estabelecidos pelo tribunal. A utilização de tecnologias avançadas, como *Large Language Models* (LLMs), tem se destacado como uma abordagem promissora para otimizar esse processo. Neste estudo, a investigação concentrou-se na utilização de LLMs para a classificação de documentos referentes às prestações de contas anuais de gestores recebidos pelo TCE/MA. Três modelos de LLMs foram examinados: mBERT, XLM-RoBERTa e mT5. Essas LLMs foram aplicadas a um conjunto de dados de textos extraídos, especificamente compilado para a pesquisa, com base em documentos fornecidos pelo TCE/MA, e avaliadas com base no F1-score. Os resultados revelaram que o modelo XLM-RoBERTa atingiu um F1-score de $98{,}99\% \pm 0{,}12\%$, enquanto o mBERT alcançou $98{,}65\% \pm 0{,}29\%$ e a mT5 apresentou $98{,}71\% \pm 0{,}75\%$. Esses resultados destacam a eficácia das LLMs na classificação de documentos de prestação de contas, proporcionando contribuições para os avanços no campo do processamento de linguagem natural. Essas abordagens têm o potencial de serem exploradas para aprimorar a automação e a precisão nas classificações de documentos.

Palavras-chave: Large Language Models, Processamento de Linguagem Natural, Classificação de Documentos, Prestação de Contas, TCE/MA.

Abstract

The Tribunal de Contas do Estado do Maranhão (TCE/MA) plays an essential role in controlling public accounts, using the electronic annual accountability system (e-PCA). In this process, the auditees send electronic documents related to the rendering of government and management accounts, classified according to established regulations. It is therefore important to classify documents correctly to ensure compliance with the standards set by the court. The use of advanced technologies, such as Large Language Models (LLMs), has been highlighted as a promising approach to the optimization of this process. In this study, the research focused on the use of LLMs to classify documents relating to the annual accounts of managers received by the TCE/MA. Three LLM models were examined: mBERT, XLM-RoBERTa and mT5. These LLMs were applied to a dataset of extracted texts specifically compiled for the research, based on documents provided by the TCE/MA, and evaluated based on the F1-score. The results strongly suggested that the XLM-RoBERTa model achieved an F1-score of $98{,}99\% \pm 0{,}12\%$, while mBERT achieved $98{,}65\% \pm 0{,}29\%$ and mT5 showed $98{,}71\% \pm 0{,}75\%$. These results highlight the effectiveness of LLMs in classifying accountability documents, providing contributions to advances in the field of natural language processing. These approaches have the potential to be exploited to improve automation and accuracy in document classifications.

Keywords: Large Language Models, Natural Language Processing, Document Classification, Accountability, TCE/MA.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

BERT	*Bidirectional Encoder Representations from Transformers*
BPE	*Byte-Pair Encoding*
CNN	*Convolutional Neural Network*
CSV	*Comma-Separated Values*
DCASP	*Demonstrações Contábeis Aplicadas ao Setor Público*
DIC	*Document Image Classification*
e-PCA	*Sistema de Prestação de Contas Anual eletrônica*
FN	*Falso Negativo*
FP	*Falso Positivo*
GPU	*Graphics Processing Unit*
IMDb	*Internet Movie Database*
LLM	*Large Language Model*
LSTM	*Long-Short Term Memory*
mBERT	*Multilingual BERT*
MLM	*Masked Language Modeling*
MLP	*Multilayer Perceptron*
mT5	*Multilingual T5*
NLP	*Natural Language Processing*
NLU	*Natural Language Understanding*
NSP	*Next Sentence Prediction*
PDF	*Portable Document Format*
PLN	*Processamento de Linguagem Natural*
RNA	*Rede Neural Artificial*

RNN	*Recurrent Neural Network*
RoBERTa	*Robustly Optimized BERT Pretraining Approach*
ROC AUC	*Area under the Receiver Operating Characteristic Curve*
T5	*Text-To-Text Transfer Transformer*
TCE	*Tribunal de Contas do Estado*
TCE/MA	*Tribunal de Contas do Estado do Maranhão*
TPE	*Tree-structured Parzen Estimator*
UFMA	*Universidade Federal do Maranhão*
VN	*Verdadeiro Negativo*
VP	*Verdeiro Positivo*
XLM-RoBERTa	*Cross-lingual Language Model RoBERTa*

Sumário

1 Introdução

O Tribunal de Contas do Estado do Maranhão (TCE/MA) é uma instituição de fundamental importância no contexto de controle e fiscalização das contas públicas do Estado e dos Municípios do Maranhão (LENZA, 2020). Através do Sistema de Prestação de Contas Anual eletrônica (e-PCA), os fiscalizados pelo órgão podem enviar informações e documento eletrônicos presentes nos registros de prestação de contas anual de governo e de gestores (TCE/MA, 2023c). Estes documentos devem estar em conformidade com Instruções Normativas estabelecidas nos regulamentos específicos do Tribunal de Contas do Estado (TCE), onde esses documentos são classificados por descrições distintas (TCE/MA, 2023b). O titular, responsável técnico da entidade ou terceiro devidamente credenciado pelo responsável, deve então preencher cada documento em seu devido tipo, realizando o *upload* de cada documento separadamente (TCE/MA, 2023a).

No cenário atual, a automatização dos processos de classificação de documentos pode trazer benefícios como uma melhor eficiência do processo de prestação de contas deste órgão e uma otimização mais eficaz dos recursos humanos. A capacidade de organizar e extrair conhecimento valioso a partir de grandes conjuntos de dados textuais é essencial para empresas, pesquisadores e profissionais de diversas áreas (WAN et al., 2019). Nesse contexto, a aplicação de técnicas de Processamento de Linguagem Natural (PLN ou NLP, do inglês *Natural Language Processing*) e redes neurais tem se mostrado extremamente promissora (KHURANA et al., 2023). Um modelo que possa classificar esses documentos de maneira automática pode, portanto, permitir uma melhoria no uso efetivo de recursos e tempo, tornando o serviço dos servidores mais ágil e eficiente.

As redes neurais têm se destacado na resolução de problemas complexos devido à sua capacidade de aprender padrões e representar informações de forma não linear. Ao combinar essa abordagem com o processamento de linguagem natural, é possível extrair características relevantes dos documentos e utilizar essa informação para classificá-los de maneira eficiente e precisa (KHURANA et al., 2023).

Recentemente, tem-se visto um aumento considerável do interesse público em modelos de inteligência artificial para o processamento de linguagem natural como os Grandes Modelos de Linguagem (LLMs, do inglês Large Language Models) (NAVEED et al., 2023). No domínio de processamento de linguagem natural, a classificação de texto é uma área com poucos trabalhos em português com foco em documentos reais, de múltiplas páginas. Em muitos casos, a tarefa é aplicada para textos curtos e bem formatados, como classificação de comentários curtos, resumo de artigos acadêmicos ou e-mails.

Com base nisso, esta pesquisa tem como objetivo demonstrar o uso de LLMs para a classificação de documentos recebidos pelo TCE/MA através do e-PCA.

1.1 Objetivos

O objetivo geral deste trabalho é avaliar e comparar diversas LLMs no contexto da classificação de documentos de prestação de contas anuais de gestores recebidos pelo TCE/MA através do e-PCA, identificando, portanto, a LLM mais eficaz e adequada para a tarefa de categorização destes documentos, com o propósito adicional de aprimorar os sistemas de processamento de linguagem natural para aplicações de organização e classificação de textos em português.

1.1.1 Objetivos Específicos

- Preparar e organizar um conjunto de dados representativo contendo prestações de contas do governo;
- Implementar e aplicar as LLMs selecionadas na tarefa de classificação dos documentos;
- Realizar experimentos controlados para avaliar o desempenho de cada LLM na classificação de documentos, considerando métricas de acurácia, *F1-score*, ROC AUC, precisão e sensibilidade (*recall*);
- Analisar os resultados obtidos.

1.2 Contribuições

A maioria das pesquisas relacionadas ao uso de LLMs para classificação de texto têm se concentrado no idioma inglês, enquanto a disponibilidade de recursos para o português é limitada, assim para com documentos extensos.

Logo, do ponto de vista acadêmico, essa pesquisa contribui para a área de processamento de linguagem natural em português, preenchendo uma lacuna importante ao contribuir com perspectivas relevantes sobre o uso efetivo de LLMs para a tarefa de classificação de documentos em nossa língua. Isso não apenas amplia o escopo de aplicação dessa poderosa técnica de processamento de linguagem natural, mas também abre novas perspectivas para a compreensão e o aproveitamento de recursos linguísticos em idiomas menos estudados, como o português. Além disso, essa pesquisa contribui para o fortalecimento e o desenvolvimento contínuo da pesquisa em PLN no contexto brasileiro, o que é crucial para inúmeras aplicações, desde a análise de sentimentos até a recuperação de informações e outras finalidades.

Do ponto de vista prático, essa pesquisa visa contribuir para a eficiência nos processos de prestação de contas anuais de governo e gestores para o TCE/MA, automatizando o processo de classificação dos documentos enviados para prestar contas.

1.3 Trabalhos Relacionados

Um dos primeiros trabalhos a utilizar o modelo BERT (*Bidirectional Encoder Representations from Transformers*) para classificação de documentos, foi o DocBERT (ADHIKARI et al., 2019a). Este trabalho criou um modelo *Long-Short Term Memory* (LSTM) aplicando destilamento de conhecimento (HINTON; VINYALS; DEAN, 2015) no modelo BERT para a classificação de documentos, chamado *KD-LSTM*$_{reg}$ (*Knowledge Destillation Long-Short Term Memory*), utilizando quatro conjuntos de dados: Reuters-21578 (APTé; DAMERAU; WEISS, 1994), arXiv Academic Paper dataset (YANG et al., 2018), avaliações na Internet Movie Database (IMDb), e avaliações de 2014 na Yelp. Foram obtidos valores de F1-score de 88,9% ± 0,5 no conjunto de dados Reuters, mas ainda que esses conjuntos de dados tenham grandes quantidades de documentos, os documentos em si são pequenos em relação ao tamanho do texto, com uma média de 175 palavras por documento (ADHIKARI et al., 2019a).

Anteriormente a este trabalho, o estado da arte em classificação de documentos era o modelo LSTM$_{reg}$ proposto por Adhikari et al. (2019b) com F1-score de 87% ± 0,5 no conjunto de dados Reuters. Outros trabalhos semelhantes foram apresentados, como (FEIJO; MOREIRA, 2020) (F1-score de 94% no conjunto de dados Folha UOL News), também em conjuntos de dados com documentos com pouco texto.

Diferente dos anteriores, um trabalho que lida com documentos longos, é o (WAN et al., 2019). Neste trabalho, o documento é dividido em várias partes antes de ser submetido ao modelo, obtendo um F1-score de até 98,2% realizando uma classificação *multi-label*.

Song et al. (2022) aborda classificação de documentos com vários rótulos no domínio jurídico. Os autores apresentam o POSTURE50K, um conjunto de dados multi-rótulo exclusivamente jurídico, e propõem uma arquitetura de aprendizagem profunda com pré-treinamento específico de domínio e um mecanismo de atenção ao rótulo. A metodologia proposta alcança um resultado de 81,2% de micro F1-score e 27,6% de macro F1-score.

Peña et al. (2023) faz uma análise de LLMs para a classificação multi-label de documentos públicos espanhóis. Recuperando documentos através de uma ferramenta baseada em *regex*, foram obtidos 33.000 documentos com 30 classes diferentes. Os autores utilizaram o RoBERTa, treinando um modelo diferente para cada classe, e reportaram as métricas sensibilidade e especificidade. A maior sensibilidade alcançada para 1 das classes foi de 93,07% com uma máquina de vetores de suporte (SVM) como classificador.

O presente trabalho faz uma classificação *single label* de documentos longos em português, dessa forma não é possível comparar diretamente com os trabalhos anteriores, visto que nenhum deles trata de documentos em português cujo tema seja prestação de contas (accountability) além de não utilizarem um modelo multilíngua como nesta pesquisa. Porém, o presente trabalho alcança resultados competitivos, superando todas as métricas mencionadas anteriormente na tarefa de classificação de documentos.

1.4 Organização do Trabalho

Este trabalho está organizado em cinco capítulos, de forma a apresentar o conteúdo mais claramente, conforme os parágrafos a seguir.

Neste capítulo, o Capítulo 1, é apresentado o contexto da problemática tratada por esta pesquisa, o objetivo geral, a lista de objetivos específicos, e as contribuições desta pesquisa.

O capítulo 2 aborda e explica os conteúdos que fundamentam a pesquisa, além de um breve resumo dos trabalhos relacionados.

Já o capítulo 3 é acerca da metodologia do presente trabalho, desde a aquisição dos documentos e modelos avaliados, até o processo de avaliação dos mesmos.

O capítulo 4 apresenta os resultados obtidos para cada modelo, analisando esses resultados de acordo com suas métricas.

No capítulo 5, tem-se a conclusão juntamente com sugestões para trabalhos futuros.

2 Fundamentação Teórica

Este capítulo explora uma base conceitual necessária às etapas subsequentes desta pesquisa, proporcionando alicerce teórico para seu desenvolvimento e assegurando que o leitor compreenda o conteúdo. Primeiramente, é discutido o formato e estrutura dos arquivos que compõem o conjunto de dados utilizado neste estudo. Na sequência, discorre-se sobre redes neurais artificiais e *transformers*.

Em seguida, abordaremos os princípios e técnicas de PLN, que desempenharão um papel fundamental na análise dos documentos. Além disso, examinaremos alguns modelos de linguagem baseados em aprendizado profundo, que têm revolucionado a área de PLN. Por fim, exploraremos a avaliação dos resultados, abordando métricas de desempenho e técnicas de validação cruzada, essenciais para garantir a robustez e confiabilidade dos resultados obtidos neste estudo.

2.1 Documentos

Através do e-PCA, o TCE/MA recebe as informações e arquivos que compõem os processos de prestação de contas anual de governo e de gestores (TCE/MA, 2023c). Esses arquivos são, em sua maioria, documentos em PDF (Portable Document Format) ou planilhas em CSV (Comma-Separated Values) que devem obedecer às Instruções Normativas do TCE/MA. Neste trabalho, foram utilizados estes documentos em PDF.

PDF é um formato de arquivo criado pela Adobe em 1992 com a finalidade de que, independente de hardware, sistema operacional ou aplicação, sua apresentação seja a mesma, obedecendo ao padrão da norma ISO 32000 (ADOBE, 2023). Desses arquivos, apenas o texto é extraído, ignorando informações de coordenadas ou estilo, para então se realizar o processamento de linguagem natural do documento através do aprendizado de máquina.

2.1.1 Prestações de Contas

A obrigatoriedade da Prestação de Contas Anual, instituída pela Constituição, impõe-se tanto ao Presidente da República quanto aos gestores de órgãos e entidades do setor público, conforme estabelecido nos artigos 70 e 71 da Constituição Federal (BRASIL, 1988, Art. 70, Art. 71). O Presidente, por sua vez, é encarregado de apresentar as contas consolidadas de toda a administração pública. Já aos demais administradores, recai a responsabilidade de prestar contas sobre os resultados obtidos na administração dos recursos confiados a sua gestão, em conformidade com os objetivos de interesse coletivo

delineados pelo poder público. Essa obrigação de prestação de contas assume a forma de uma autoavaliação, na qual os administradores avaliam e reportam os resultados alcançados em relação aos objetivos estabelecidos, demonstrando transparência e responsabilidade na gestão dos recursos públicos (BRASIL, 1988).

2.2 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) constituem uma classe de modelos em aprendizado de máquina, inspiradas no funcionamento do sistema nervoso biológico. Compostas por unidades interconectadas, conhecidas como neurônios, as RNAs conseguem aprender padrões complexos a partir de dados. Este paradigma computacional tem ganhado proeminência devido à sua versatilidade e sucesso em uma variedade de aplicações, incluindo reconhecimento de padrões, processamento de linguagem natural e visão computacional (LECUN; BENGIO; HINTON, 2015; ZHENG et al., 2023).

Segundo LeCun, Bengio e Hinton (2015), a popularização das RNAs tem sido impulsionada por avanços em algoritmos de treinamento, arquiteturas eficientes, e o aumento do poder computacional disponível. O desenvolvimento de redes mais profundas, como as redes neurais profundas convolucionais (CNN, do inglês Convolutional Neural Network) e redes neurais recorrentes (RNN, do inglês Recurrent Neural Network), tem permitido a modelagem de relações mais complexas nos dados. A combinação de técnicas inovadoras e a crescente disponibilidade de conjuntos de dados volumosos têm impulsionado significativamente o progresso nas capacidades das RNAs, consolidando seu papel como uma ferramenta poderosa no cenário da inteligência artificial.

2.3 Transformers

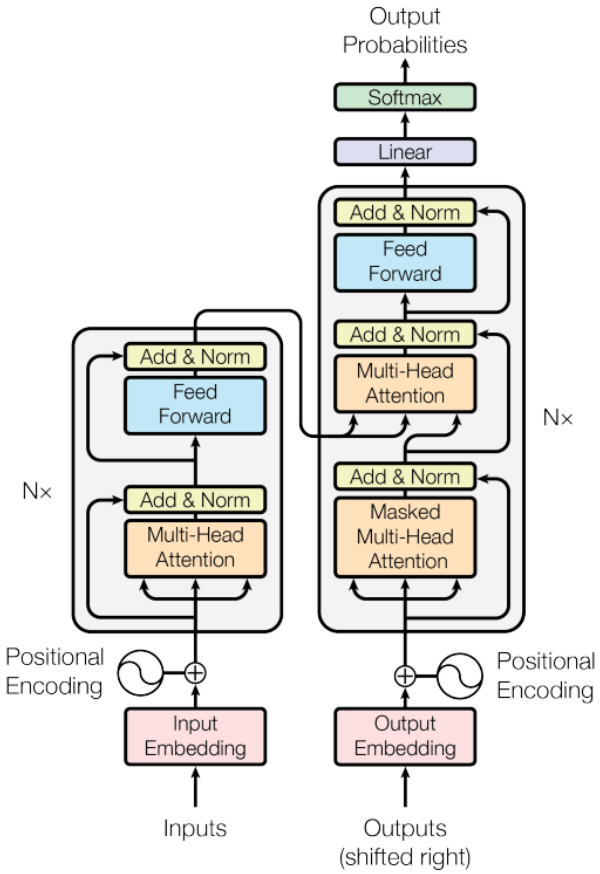
O *Transformer* é um inovador modelo de rede neural que revolucionou o campo de Processamento de Linguagem Natural. Introduzido por Vaswani et al. (2017) em 2017, o Transformer afasta-se das arquiteturas de redes neurais sequenciais convencionais, como as recorrentes, empregando uma abordagem baseada em mecanismos de atenção. Sua estrutura permite que o modelo considere todas as posições das palavras em uma sequência de entrada simultaneamente, superando as limitações de dependências de contexto em tarefas complexas (VASWANI et al., 2017).

Uma característica distinta do *Transformer* é sua capacidade de paralelizar o treinamento, tornando-o mais eficiente em comparação com arquiteturas sequenciais. Além disso, sua arquitetura escalável e adaptável permitiu sua aplicação em diversas tarefas, desde tradução automática até sumarização de texto. O sucesso do *Transformer* marcou

um marco significativo no avanço do estado-da-arte em PLN, influenciando subsequentes desenvolvimentos na área (HUANG et al., 2023).

A arquitetura do modelo *Transformer* é ilustrada na Figura 1, sua arquitetura compreende camadas de codificadores e decodificadores, chamadas também de camada de *encoder* e camada de *decoder*, cada uma equipada com múltiplos blocos nos quais o mecanismo de atenção é aplicado de forma ponderada. Cada bloco contém camadas totalmente conectadas e módulos de normalização. Essa abordagem contribui para a eficiência computacional e a capacidade do *Transformer* em capturar dependências de longo alcance, tornando-o uma escolha proeminente em diversas tarefas de Processamento de Linguagem Natural (HUANG et al., 2023; VASWANI et al., 2017).

Figura 1 – Arquitetura do modelo Transformer



Fonte: (VASWANI et al., 2017)

Explicando mais detalhadamente, no bloco *encoder* a entrada é tokenizada, codificada com uma codificação de posição (um vetor que é adicionado em cada entrada), particionada em vetores de consulta, chave e valor que são aplicados a um bloco de *multi-head attention*. Este bloco pondera as informações relevantes da sequência de entrada e tem a saída normalizada e concatenada com um conexão residual. Após este passo há uma camada *feed-forward* e uma repetição da conexão residual e normalização (BA; KIROS; HINTON, 2016). O bloco *encoder* é então repetido N vezes. Ao final da camada

de *encoder*, a saída é passada para a camada de *decoder*, que possui uma arquitetura semelhante, mas adiciona uma máscara no bloco de atenção que impede a camada de conhecer posições futuras durante o treinamento para garantir que as previsões sejam autoregressivas (VASWANI et al., 2017).

2.4 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN), uma interseção entre Inteligência Artificial e Linguística, visa capacitar os computadores a captarem a complexidade das declarações ou palavras expressas nas linguagens humanas. Essa disciplina, emergindo da necessidade de uma interação mais intuitiva entre usuários e sistemas computacionais, concentra-se em tornar a comunicação mais acessível. Ao eliminar a barreira da fluência em linguagens específicas de máquina, a PLN se torna uma ferramenta poderosa para democratizar o acesso à tecnologia (KHURANA et al., 2023).

No contexto dessa área, destacam-se avanços notáveis, como *word2vec* (MIKOLOV et al., 2013a), *doc2vec* (MIKOLOV et al., 2013b) e *embeddings*, que desempenham um papel crucial na representação semântica e contextual de palavras e documentos. O *word2vec*, por exemplo, introduziu uma abordagem inovadora ao mapear palavras em vetores numéricos, capturando relações semânticas e contextuais. Da mesma forma, o *doc2vec* estende essa ideia para documentos inteiros, atribuindo representações vetoriais a textos mais extensos. Esses avanços contribuíram significativamente para a melhoria do desempenho em tarefas de PLN (MIKOLOV et al., 2013b; MIKOLOV et al., 2013a; BENGIO et al., 2003).

Esses métodos de representação vetorial demonstram a evolução constante do campo de PLN, impulsionando a capacidade dos sistemas de processar e entender a linguagem humana de maneira mais eficaz. À medida que a disciplina continua avançando, abordagens inovadoras na tokenização, outra faceta crucial do PLN, tornam-se relevantes para aprimorar ainda mais a capacidade dos sistemas de processar e entender textos de maneira significativa (ALMEIDA; XEXéO, 2019; JABBAR, 2023).

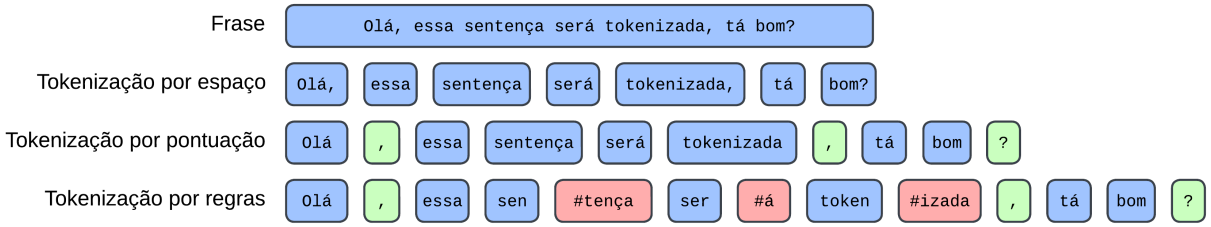
2.4.1 Tokenização

Na fase inicial de qualquer projeto de Processamento de Linguagem Natural, a principal tarefa é o pré-processamento do texto. Essa etapa é crucial para estabelecer dados de maneira previsível e analisável. Entre as técnicas de pré-processamento, a tokenização se destaca, sendo essencial em processamento de linguagem natural com redes neurais (JABBAR, 2023; TORAMAN et al., 2023).

Tokenização, ilustrada na Figura 2 é o ato de quebrar um fluxo contínuo de dados textuais em elementos discretos, como palavras, termos, sentenças ou símbolos, conhecidos como tokens. Um tokenizador desempenha um papel crucial na transformação da

entrada bruta em um formato estruturado e analisável, permitindo análise e interpretação posteriores (TORAMAN et al., 2023).

Figura 2 – Exemplos de Tokenização



Fonte: Autor (2023)

Uma característica interessante da tokenização é sua capacidade de gerar ocorrências simbólicas em um documento, utilizadas diretamente como vetor representativo do documento. Em suma, a tokenização facilita a segmentação de texto e a conversão de dados em um formato aproveitável para tarefas subsequentes, como representação de documentos e extração de recursos em aplicações de PLN (MIKOLOV et al., 2013b; MIKOLOV et al., 2013a).

2.4.2 WordPiece

O algoritmo de tokenização de subpalavras utilizado pelo BERT (Seção 2.5.1) é conhecido como *WordPiece* (DEVLIN et al., 2018). Foi proposto inicialmente por Schuster e Nakajima (2012) e implementado pela Google em 2016. No processo inicial, o *WordPiece* inicializa o vocabulário incorporando todos os caracteres presentes nos dados de treinamento. Progressivamente, o algoritmo aprende um determinado número de regras de mesclagem. Para fazer a escolha de que símbolos devem ser mesclados, o algoritmo seleciona o par que maximiza a probabilidade de os dados de treinamento serem incorporados ao vocabulário (WU et al., 2016).

Maximizar a probabilidade dos dados de treinamento significa identificar o par de símbolos cuja probabilidade, quando dividida pelas probabilidades de seu primeiro e segundo símbolos isolados, é a mais elevada entre todos os pares de símbolos. De maneira intuitiva, o WordPiece avalia as perdas ao mesclar dois símbolos para garantir que a fusão seja verdadeiramente vantajosa (WU et al., 2016).

2.4.3 SentencePiece

SentencePiece, proposto por Kudo e Richardson (2018) em 2018, é um algoritmo que representa uma reinterpretação das unidades de sub-palavras, uma estratégia eficaz para atenuar os desafios associados ao vocabulário aberto em tradução automática. Este algoritmo assume o papel de tokenizador de texto não supervisionado, destinado

principalmente a sistemas de geração de texto fundamentados em redes neurais (KUDO; RICHARDSON, 2018).

O algoritmo aborda a entrada de forma a considerá-la como um contínuo fluxo de entrada bruto, incorporando, dessa maneira, o caractere "espaço" no conjunto de caracteres a ser utilizado. Posteriormente, recorre ao uso do algoritmo Byte-Pair Encoding (BPE) (GAGE, 1994) ou do unigrama com a finalidade de construir o vocabulário adequado para a aplicação em questão (SENNRICH; HADDOW; BIRCH, 2016). Essa abordagem proporciona flexibilidade ao processo, permitindo a adaptação dinâmica às nuances presentes nos dados. Em essência, ele oferece a possibilidade de eliminar a necessidade de depender de processamentos específicos de idioma, seja antes ou depois do treinamento (KUDO; RICHARDSON, 2018).

2.5 Large Language Models

Os Large Language Models (LLMs) são modelos de aprendizado de máquina capazes de desempenhar uma ampla gama de tarefas em PLN. Suas funcionalidades abrangem desde a automação na geração de textos até a resposta a perguntas, passando pela tradução automática, classificação de documentos, geração de representações de texto, entre outras aplicações (KADDOUR et al., 2023). A base central desses modelos, em geral, incorporada a arquitetura de *transformers* (DEVLIN et al., 2018; VASWANI et al., 2017).

Conforme sugerido pelo próprio nome, os LLMs são modelos que se destacam por sua magnitude e habilidade de processar extensas quantidades de texto. Essa característica confere a esses modelos a capacidade de captar os significados do texto e gerar respostas legíveis na linguagem humana, o que, por conseguinte, os torna aptos a serem aplicados em diversos domínios e contextos dentro do campo de processamento de linguagem natural (RADFORD et al., 2018).

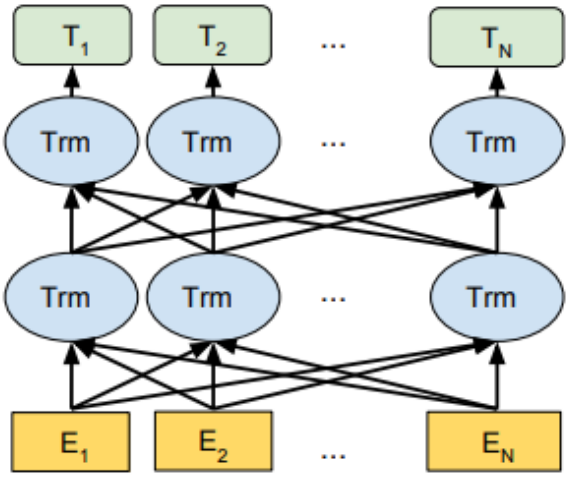
Os resultados eficientes alcançados pelos LLMs derivam-se, na maioria, de sua capacidade de gerar uma saída com um nível satisfatório de compreensão. Essa habilidade é proveniente da arquitetura desses modelos, que foi aprimorada e refinada por meio de treinamento auto supervisionado extensivo, utilizando um vasto conjunto de dados textuais. Essa eficácia no processamento linguístico representa um avanço notável no campo, com modelos como BERT sendo reconhecidos como precursores dessa abordagem (DEVLIN et al., 2018; RICHARDSON; HECK, 2023).

2.5.1 mBERT

BERT é um modelo amplamente utilizado em PLN, proposto por Devlin et al. (2018), desenvolvido pela Google, e apresenta características distintivas que influenciam sua eficácia e desempenho em tarefas específicas. O pré-treinamento do BERT envolve dois

objetivos principais: *masked language modeling* (MLM) e *next sentence prediction* (NSP). Sua habilidade eficiente na previsão de tokens mascarados e em Tarefas de *natural language understanding* (NLU) em geral, é amplamente reconhecida. No entanto, é importante notar que o BERT, embora seja eficaz em diversas áreas, não se destaca particularmente na geração de texto (CLARK et al., 2019; TENNEY; DAS; PAVLICK, 2019).

Figura 3 – BERT



Legenda: E_n = Token Embeddings; Trm = Codificadores Transformers; T_n = Token.

Fonte: Adaptado de (DEVLIN et al., 2018)

O BERT é composto por uma pilha de codificadores (*encoders*) do Transformer, ilustrada na Figura 3, que passaram por um extenso processo de treinamento. No que diz respeito à arquitetura específica do BERT, ambas as variantes, conhecidas como *Base* (110M de parâmetros) e *Large* (340M de parâmetros), apresentam um grande número de camadas de codificadores, referidas no paper como *Transformer Blocks*. A versão *Base* tem doze dessas camadas, enquanto a versão *Large* tem vinte e quatro camadas. Além disso, incorpora redes *feedforward* mais extensas, com 768 camadas ocultas na versão *Base* e a versão *Large* ampliando ainda mais para 1024 camadas ocultas. Da mesma forma, em termos de *heads* de atenção, a versão *Base* integra 12, enquanto a versão *Large* apresenta um total de 16 *heads* de atenção. Diferente do *Transformer* no paper original, que compreende apenas 6 camadas de codificadores, 512 camadas ocultas e 8 cabeças de atenção (DEVLIN et al., 2018).

Durante o processo de pré-treinamento, o modelo corrompe intencionalmente os tokens, gerados pelo algoritmo de tokenização *WordPiece* (WU et al., 2016), de entrada por meio de mascaramento aleatório, onde uma determinada porcentagem de tokens (geralmente 15%) é mascarada com um token de máscara especial (80% de probabilidade), um token aleatório diferente (10% de probabilidade) e o mesmo token original (10% de probabilidade). Outra característica é que o primeiro token é um token especial dedicado

à classificação do texto em questão. Além disso, o BERT tem um segundo objetivo relacionado à previsão de frases. Suas entradas consistem em duas frases, A e B, com um *token* de separação entre elas. O modelo, com uma probabilidade de 50%, lida com frases consecutivas no corpus, enquanto nos restantes 50%, as frases não têm relação entre si. Portanto, o modelo é desafiado a prever não apenas a frase original, mas também a relação de continuidade entre as frases A e B. Essa abordagem diversificada contribui para a versatilidade do BERT em diversas aplicações de processamento de linguagem natural (DEVLIN et al., 2018). Este modelo foi posteriormente pré-treinado em 104 línguas, sendo esta versão multilíngue conhecida como mBERT.

2.5.2 XLM-RoBERTa

Desenvolvido pela antiga Facebook (atual Meta) em 2019, o modelo RoBERTa cujo nome significa "abordagem de pré-treinamento BERT robustamente otimizada" (do inglês Robustly Optimized BERT Pretraining Approach) é uma evolução do BERT. Enquanto o BERT se baseia em pré-treinamento com o objetivo de prever a frase seguinte, o RoBERTa diverge desta abordagem adotando *mini-batches* significativamente maiores, de 256 para 8 mil. Embora compartilhe a mesma arquitetura básica do BERT, o RoBERTa utiliza um tokenizador baseado em Byte-Pair Encoding (BPE), que separa os tokens a nível de *bytes*, semelhante ao GPT-2 (RADFORD et al., 2019), e implementa um esquema de pré-treino distinto (LIU et al., 2019; SENNRICH; HADDOW; BIRCH, 2016) .

Diferenciando-se do BERT, o RoBERTa incorpora diversas melhorias no pré-treino. Destacam-se práticas como o mascaramento dinâmico, em que os *tokens* são mascarados de maneira variada em cada época, ao contrário do BERT, que realiza o mascaramento de uma só vez para todos os *tokens*. Além disso, o RoBERTa adota a estratégia de agrupar até atingir 512 tokens, o que pode abranger várias frases e documentos de uma vez, e treina com *batches* maiores (LIU et al., 2019).

Os autores do RoBERTa sustentam que o BERT, em sua implementação original, está subtreinado e propõem uma série de melhorias para abordar essa limitação. Essas melhorias incluem o aumento significativo na quantidade de dados de treino, com 16G para o BERT e 160G para o RoBERTa, a adoção de um padrão de mascaramento dinâmico em vez de um padrão estático, a substituição do objetivo de prever a frase seguinte por frases completas sem *next sentence prediction* e o treinamento em sequências mais longas. Essas adaptações visam aprimorar a eficácia e a generalização do modelo RoBERTa em uma variedade de tarefas de processamento de linguagem natural (LIU et al., 2019).

O modelo XLM-RoBERTa (Cross-lingual Language Model RoBERTa) proposto por Conneau et al. (2019), lançado com base no modelo RoBERTa, representa uma extensão das capacidades linguísticas sendo treinado com um vasto conjunto de dados proveniente de 2.5 terabytes de informações filtradas do *CommonCrawl*, proporcionando uma compreensão

de estruturas linguísticas em diversas línguas. Durante o pré-treinamento desta variação foi utilizado o tokenizador *SentencePiece* ao invés do tokenizador BPE que o RoBERTa original utiliza (CONNEAU et al., 2019).

2.5.3 mT5

O T5, ou *Text-To-Text Transfer Transformer*, proposto por Raffel et al. (2019) e desenvolvido pela Google em 2019, é um modelo pré-treinado envolvido em uma abordagem multitarefa, misturando tarefas tanto não supervisionadas quanto supervisionadas. Esse modelo converte cada tarefa em um formato texto para texto, empregando o tokenizador *SentencePiece* em seu processo. No âmbito do treinamento, adota a técnica de *teacher forcing*, exigindo sempre uma sequência de entrada e sua correspondente sequência de destino (RAFFEL et al., 2019).

Ao abordar tarefas específicas, um prefixo distinto é adicionado à entrada, adequando-se a uma variedade de contextos. Por exemplo, para tradução, o modelo utiliza um prefixo que denota a ação, como "traduzir inglês para alemão", enquanto para resumos, emprega "resumir". No que tange ao treinamento auto-supervisionado, o T5 recorre à estratégia de *tokens* corrompidos, substituindo aleatoriamente 15% dos *tokens* por *tokens* sentinelas individuais. Se múltiplos *tokens* consecutivos são designados para remoção, todo o conjunto é substituído por um único *token* sentinela. Nesse processo, a entrada do codificador consiste na frase corrompida, e a entrada do decodificador corresponde à frase original, e o alvo são os *tokens* eliminados, delimitados pelos *tokens* sentinelas (RAFFEL et al., 2019).

Uma variante é o mT5, uma extensão multilíngue do T5. Esta versão é pré-treinada em um novo conjunto de dados baseado no *Common Crawl*, abrangendo uma variedade de 101 línguas. A escala dessa versão varia desde a versão *small* com 300 milhões de parâmetros até a versão XXL com 13 bilhões de parâmetros, expandindo assim a capacidade de generalização do modelo para uma ampla diversidade linguística (XUE et al., 2020).

2.6 Métricas de Avaliação

Para avaliar o desempenho dos modelos na execução da tarefa de classificação de documentos, é necessário recorrer à aplicação de métricas de avaliação (GRANDINI; BAGLI; VISANI, 2020). De acordo com Seliya, Khoshgoftaar e Hulse (2009), diversas métricas são comumente empregadas para abordar o problema de classificação, destacando-se, entre elas, a acurácia, precisão, sensibilidade, F1-score e a área sob a curva característica de operação do receptor (AUC ROC, ou área sob a curva ROC). A utilização dessas métricas proporciona uma análise multifacetada do desempenho dos modelos, considerando diferentes aspectos como a precisão na predição das classes, a capacidade de capturar

verdadeiros positivos, a sensibilidade à identificação de casos positivos, e a área sob a curva ROC. Essa variedade de métricas contribui para uma avaliação mais abrangente da eficácia dos modelos no contexto específico da classificação documental.

Adicionalmente, a função de perda (*loss function*) do modelo constitui uma métrica importante na análise do desempenho do mesmo. Conforme destacado por Wang et al. (2022), a função de perda desempenha um papel essencial na concepção, otimização e aprimoramento de algoritmos de aprendizado de máquina. Um exemplo comum de função de perda para problemas de classificação é a Entropia Cruzada (*cross-entropy*), a qual se destaca por sua eficácia na minimização da divergência entre as distribuições de probabilidade previstas e reais.

2.6.1 Entropia Cruzada

Segundo Seliya, Khoshgoftaar e Hulse (2009), a entropia cruzada (ou $\overline{CE}$) é um valor que se estende no intervalo entre 0 e $+\infty$ atingindo o valor ideal de 0 para um classificador perfeito. Ela é definida como na Equação 2.1, onde $x^i \in \mathrm{R^n}$ denota os vetores de entrada do conjunto de dados, $c \in \{0, 1\}$ denota a classe da instância atual, $j \in \{0, 1\}$ denota as classes possíveis, $\hat{p}$ denota as probabilidades da classe prevista, e N o número de instâncias no conjunto de dados. A entropia cruzada demonstra sua relevância na avaliação de quão bem um modelo de aprendizado de máquina consegue aproximar a distribuição de probabilidade prevista da distribuição real dos rótulos.

$$\overline{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=0}^{1} p\left(c = j|x^i\right) \times \ln\left(\hat{p}\left(c = j|x^i\right)\right) \tag{2.1}$$

2.6.2 Acurácia

Considerando N como a quantidade total de instâncias presente no conjunto de dados, VP como a quantidade de verdadeiros positivos e VN como a quantidade de verdadeiros negativos, é possível calcular a acurácia utilizando a formulação expressa na Equação 2.2 (SELIYA; KHOSHGOFTAAR; HULSE, 2009). Sua variação ocorre de 0 a 1, sendo 1 atribuído a um classificador ideal. Esta métrica representa a capacidade do modelo em realizar classificações corretas. A acurácia é um indicador global do desempenho do modelo, refletindo a proporção de predições corretas em relação ao total de instâncias no conjunto de dados.

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{N} \tag{2.2}$$

2.6.3 Precisão

Levando em consideração que VP representa a quantidade de verdadeiros positivos e FP indica a quantidade de falsos positivos, a fórmula para a precisão é explicitada na Equação 2.3 (DAVIS; GOADRICH, 2006). Seu valor situa-se entre 0 e 1, sendo que 1 representa um desempenho perfeito do classificador. Esta medida considera a proporção de instâncias positivas corretamente identificadas em relação ao total de instâncias identificadas como positivas, fornecendo uma medida da capacidade do modelo em realizar predições positivas corretas. Isso torna a métrica de precisão útil para casos em que a ênfase está na minimização de falsos positivos e na maximização da confiança nas predições positivas do modelo.

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{2.3}$$

2.6.4 Sensibilidade

Considerando VP como verdadeiros positivos e FN como falsos negativos, a fórmula para a métrica de precisão é expressa na Equação 2.4 (DAVIS; GOADRICH, 2006). Varia de 0 a 1, com 1 representando um classificador ótimo. Ao considerar a relação entre acertos verdadeiramente positivos e a subestimação de instâncias positivas, essa métrica fornece uma medida da exatidão do modelo na identificação correta das instâncias que realmente pertencem à classe positiva. Ela é útil em cenários onde a ênfase recai na minimização de falsos negativos, ou seja, na redução da ocorrência de instâncias positivas erroneamente classificadas como negativas.

$$\text{Sensibilidade} = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{2.4}$$

2.6.5 F1-score

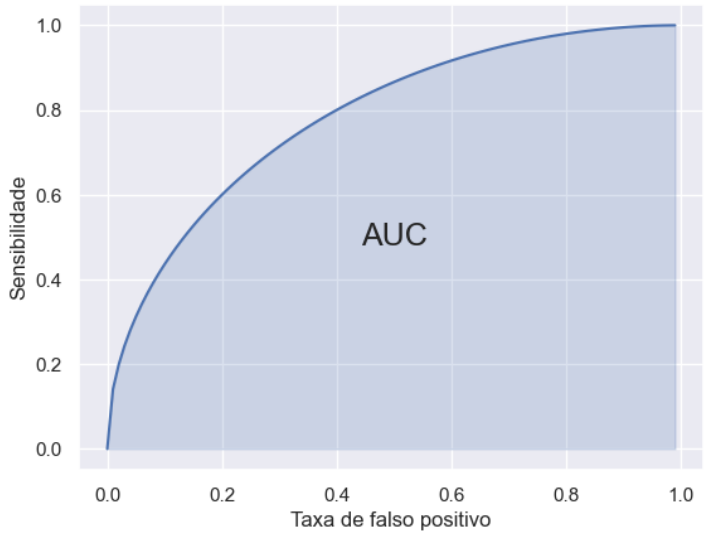
O F1-score é construído a partir de duas métricas distintas, a saber, precisão e sensibilidade. Tal relação é matematicamente expressa na Equação 2.5, é a média harmônica entre a precisão e a sensibilidade, e seu valor varia entre 0 e 1, sendo 1 o valor para um classificador perfeito (SELIYA; KHOSHGOFTAAR; HULSE, 2009). Essa métrica proporciona uma visão da capacidade do modelo em equilibrar precisão e sensibilidade na tarefa de classificação.

$$\text{F1-score} = \frac{2 \times \text{Precisão} \times \text{Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}} \tag{2.5}$$

2.6.6 AUC ROC

A área sob a curva característica de operação do receptor (ou curva ROC) também é feito a partir de duas métricas: a taxa de verdadeiros positivos, também conhecida como *recall* ou sensibilidade, e a taxa de falsos positivos dada pela Equação 2.6, onde FP significa falsos positivos e VN são verdadeiros negativos (DAVIS; GOADRICH, 2006). Calculada graficamente e ilustrada na Figura 4, a curva ROC representa a taxa de verdadeiros positivos em função da taxa de falsos positivos em vários pontos de corte para as probabilidades de classificação. A área sob essa curva, conhecida como AUC ROC, varia de 0 a 1, sendo 1 indicativo de um desempenho perfeito do modelo. Quanto mais próxima de 1, melhor é a capacidade do modelo em distinguir entre classes. Ela representa a probabilidade do modelo classificar corretamente uma instância positiva aleatória em relação a uma instância negativa aleatória.

Figura 4 – Gráfico do AUC ROC



Fonte: Autor (2023)

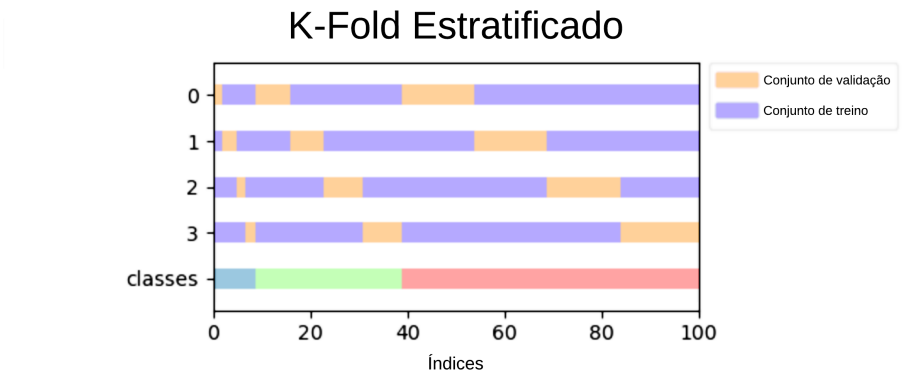
$$\text{Taxa de falsos positivos} = \frac{\text{FP}}{\text{FP} + \text{VN}} \tag{2.6}$$

2.7 Validação

A avaliação dos resultados de um modelo é frequentemente conduzida mediante a apresentação ao modelo de um conjunto de dados que inclui resultados conhecidos para fins de treinamento, assim como um conjunto de dados com resultados desconhecidos para a validação. Essa abordagem visa testar a generalização do modelo, ou seja, a eficácia do modelo ao prever dados que não foram previamente utilizados em seu treinamento. Uma das técnicas empregadas para este propósito é a validação cruzada, um método que envolve

a divisão do conjunto de dados em subconjuntos distintos para treinamento e validação, permitindo a avaliação iterativa do modelo em diferentes partições e, consequentemente, fornecendo uma avaliação mais generalizada de seu desempenho (YADAV; SHUKLA, 2016).

Figura 5 – Validação Cruzada K-Fold Estratificada



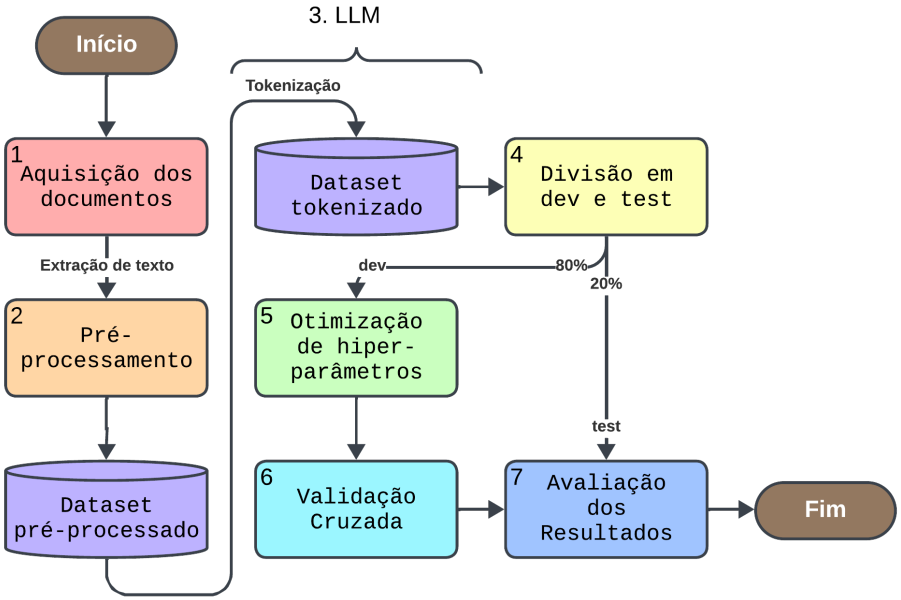
Fonte: Adaptado de Pedregosa et al. (2011)

Uma abordagem amplamente empregada na aplicação da técnica de validação cruzada é conhecida como Validação Cruzada K-Fold Estratificada, ilustrada na Figura 5. Nessa metodologia, os dados são divididos em K partes equivalentes chamados *folds*, e o treinamento do modelo ocorre iterativamente em $K - 1$ dessas partes, enquanto a parte restante é reservada para fins de validação. Este processo é repetido K vezes chamadas *splits*, onde a parte destinada à validação é alterada sequencialmente, uma a uma, até que todas as partes do conjunto de dados tenham sido utilizadas como conjunto de validação em algum ponto da validação. Essa abordagem estratificada visa garantir uma distribuição representativa das classes em cada subconjunto de treinamento e validação (YADAV; SHUKLA, 2016).

3 Metodologia

Neste capítulo, é delineada a metodologia adotada para conduzir as avaliações de desempenho das LLMs abordadas neste estudo. A visualização do fluxo da metodologia empregada é resumida na Figura 6 a seguir, sendo que cada etapa é explorada ao longo deste capítulo. A explicação aprofundada de cada passo busca proporcionar uma melhor compreensão do processo empregado.

Figura 6 – Fluxo da metodologia



Fonte: Autor (2023)

3.1 Aquisição dos Documentos

Os documentos utilizados neste projeto foram disponibilizados pela base de dados do Tribunal de Contas do Estado do Maranhão por intermédio do seu sistema eletrônico e-PCA. Um total de 19.853 foram coletados. Esses documentos, que originalmente se apresentavam no formato PDF, foram submetidos a um procedimento de extração de texto, viabilizando assim a sua utilização e análise no contexto deste projeto, para isso utilizou-se a biblioteca *pypdfium2*.

Adicionalmente, foi conduzida uma fase de saneamento destes documentos, visando a eliminação de arquivos que se encontravam corrompidos ou que, por uma variedade de razões, não possibilitaram a extração apropriada de seus conteúdos textuais. Além disso, foram identificados e excluídos documentos duplicados, contribuindo assim para a qualidade do conjunto de dados utilizado pelo presente estudo.

3.2 Pré-processamento

Após a conclusão da etapa inicial de extração de texto a partir dos documentos em formato PDF, o procedimento seguinte consiste em uma fase dedicada ao pré-processamento textual. Este estágio é crucial para a eliminação de uma parcela do ruído decorrente da conversão do arquivo PDF para o formato textual, visando, assim, estabilizar e aprimorar o processo de criação dos tokens, fundamentais para as fases subsequentes de análise e processamento (WU et al., 2016; ZHAO et al., 2022; CAMACHO-COLLADOS; PILEHVAR, 2017).

Este pré-processamento pode ser visto no Algoritmo 1. Os valores atribuídos aos tamanhos máximos das sequências de caracteres repetidos, espaços em branco e caracteres especiais foram determinados de maneira empírica, adotando uma abordagem que visou eliminar sequências de ruído do texto. Foram escolhidos os valores: no máximo três para caracteres repetidos, sequências de caracteres em branco encurtadas para um único caractere e no máximo três caracteres especiais em sequência. Foram testados diversos valores e optou-se por três caracteres para evitar a remoção excessiva de informações textuais que poderiam ser relevantes e úteis para a classificação efetiva do documento em questão, buscando encontrar um equilíbrio entre a redução do ruído e a preservação das informações para a tarefa de classificação documental.

Algoritmo 1: Pré-processamento do texto.

Input: Texto a ser pré-processado.
Output: Texto pré-processado.

1 Remover tokens especiais sem relevância ao texto;
2 Remover caracteres repetidos;
3 Encurtar sequências de caracteres em branco;
4 Encurtar sequências de caracteres em especiais;

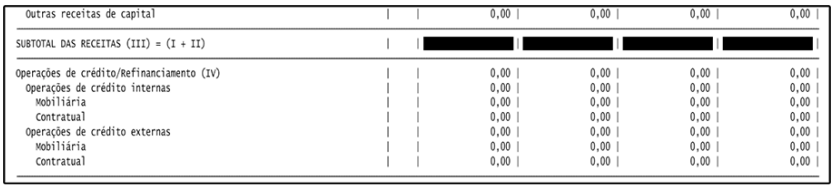
A Figura 7 é um exemplo de ruído. Neste exemplo, a linha da coluna no documento em PDF é transformada em um conjunto de traços no texto extraído. Quando o texto extraído sofre o processo de tokenização, cada traço corresponde a um token e ocupa espaço da entrada do modelo de maneira desnecessária. Logo, é preciso remover estes ruídos para o modelo poder receber informações relevantes suficientes ao documento.

Ao final do processo, os tipos de documentos escolhidos para realizar a classificação de documentos desta pesquisa estão listados a seguir:

- (DCASP) Balanço orçamentário
- (DCASP) Balanço financeiro
- (DCASP) Balanço patrimonial

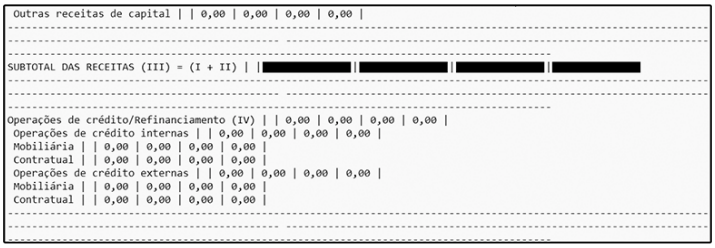
Figura 7 – Documento de balanço orçamentário. (a) Documento em PDF. (b) Texto extraído. (c) Texto extraído pré-processado.

(a) Documento em PDF



Outras receitas de capital			0,00	0,00	0,00	0,00
SUBTOTAL DAS RECEITAS (III) = (I + II)			[redacted]			
Operações de crédito/Refinanciamento (IV)			0,00	0,00	0,00	0,00
Operações de crédito internas			0,00	0,00	0,00	0,00
Mobiliária			0,00	0,00	0,00	0,00
Contratual			0,00	0,00	0,00	0,00
Operações de crédito externas			0,00	0,00	0,00	0,00
Mobiliária			0,00	0,00	0,00	0,00
Contratual			0,00	0,00	0,00	0,00

(b) Texto extraído

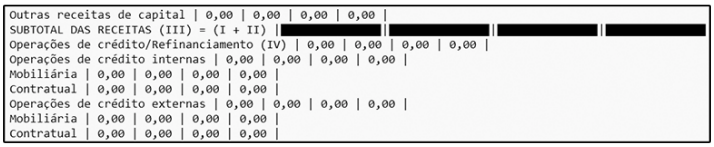


Outras receitas de capital | | 0,00 | 0,00 | 0,00 | 0,00 |

SUBTOTAL DAS RECEITAS (III) = (I + II) | | [redacted]

Operações de crédito/Refinanciamento (IV) | | 0,00 | 0,00 | 0,00 | 0,00 |
Operações de crédito internas | | 0,00 | 0,00 | 0,00 | 0,00 |
Mobiliária | | 0,00 | 0,00 | 0,00 | 0,00 |
Contratual | | 0,00 | 0,00 | 0,00 | 0,00 |
Operações de crédito externas | | 0,00 | 0,00 | 0,00 | 0,00 |
Mobiliária | | 0,00 | 0,00 | 0,00 | 0,00 |
Contratual | | 0,00 | 0,00 | 0,00 | 0,00 |

(c) Texto extraído pré-processado



Outras receitas de capital | 0,00 | 0,00 | 0,00 | 0,00 |
SUBTOTAL DAS RECEITAS (III) = (I + II) | [redacted]
Operações de crédito/Refinanciamento (IV) | 0,00 | 0,00 | 0,00 | 0,00 |
Operações de crédito internas | 0,00 | 0,00 | 0,00 | 0,00 |
Mobiliária | 0,00 | 0,00 | 0,00 | 0,00 |
Contratual | 0,00 | 0,00 | 0,00 | 0,00 |
Operações de crédito externas | 0,00 | 0,00 | 0,00 | 0,00 |
Mobiliária | 0,00 | 0,00 | 0,00 | 0,00 |
Contratual | 0,00 | 0,00 | 0,00 | 0,00 |

Fonte: Autor (2023)

- (DCASP) Demonstração das mutações do patrimônio líquido
- (DCASP) Demonstração das variações patrimoniais
- (DCASP) Demonstração dos fluxos de caixa
- (DCASP) Notas explicativas
- Relatório e certificado de auditoria, com parecer do dirigente do órgão de controle interno
- Exposição circunstanciada da gestão
- Extratos e conciliações bancárias
- Ofício de encaminhamento ao TCE/MA

3.3 LLMs

Após o pré-processamento dos textos, antes de encaminhar o texto para a LLM, é necessário realizar o procedimento de tokenização, como descrito na Seção 2.4.1. Cada LLM tem seu próprio processo de tokenização que depende da maneira como foi realizada seu pré-treinamento.

Neste trabalho, foram utilizados os seguintes modelos:

- mBERT, com sua tokenização baseada no algoritmo *WordPiece*;
- XLM-RoBERTa, com sua tokenização baseada no algoritmo *BPE*;
- mT5, com sua tokenização baseada no algoritmo *SentencePiece*.

Neste trabalho, optou-se por pré-processar e realizar o procedimento de tokenização de todos os documentos antes do treinamento. Os documentos pré-processados e tokenizados são salvos em disco, otimizando assim o processo de treinamento e validação, visto que desta forma não é necessário executar a tarefa de tokenização durante o treinamento.

Na Figura 8, proporciona-se uma visão geral do funcionamento do processo de classificação efetuado neste estudo. Inicialmente, destaca-se a fase em que o texto é extraído dos documentos em formato PDF, passando, em seguida, por um processo de pré-processamento, conforme previamente explicado. Posteriormente, o texto submete-se a uma etapa de tokenização, segmentando-o em tokens para a LLM.

Subsequentemente, os primeiros 512 tokens resultantes desse processo de tokenização são encaminhados ao LLM. Este último, por sua vez, responsável pela extração de informações latentes, entrega uma saída que reflete o espaço vetorial característico do documento em análise.

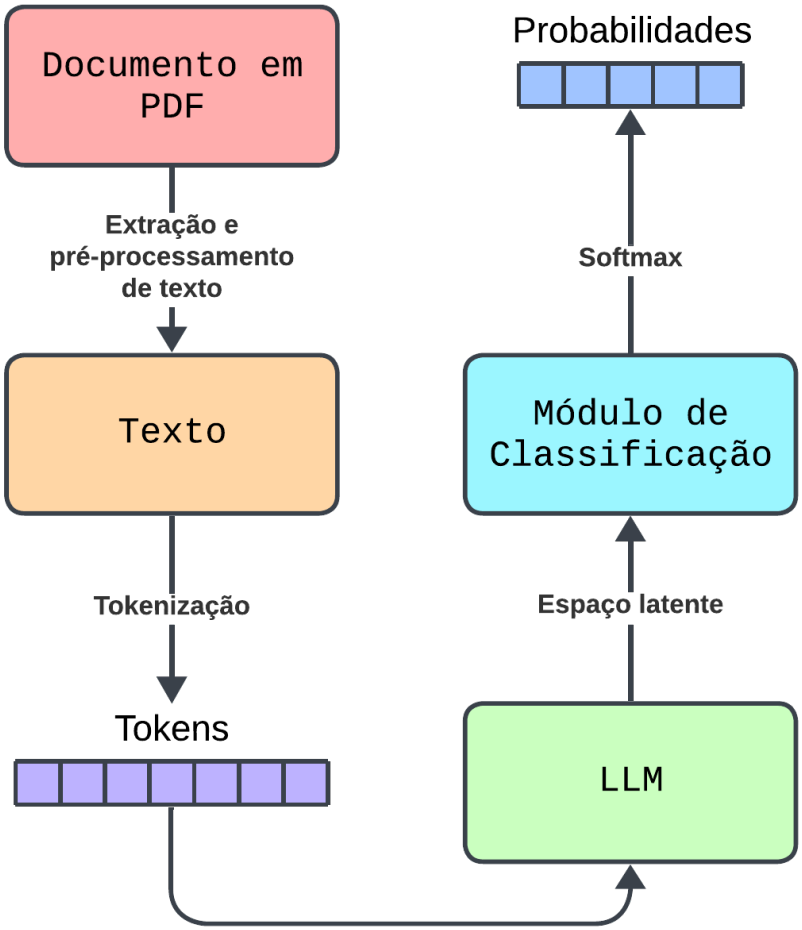
Posteriormente, a saída do LLM é direcionada ao módulo de classificação, uma instância composta por uma Multilayer Perceptron (MLP) que, por sua vez, após passar pela função *softmax*, gera um conjunto de 11 probabilidades, cada uma associada às respectivas classes finais. Este processo de etapas concatenadas constitui o método empregado no contexto desta pesquisa para o processo de classificação dos documentos.

3.4 Divisão dos Dados

O conjunto de documentos tokenizados é dividido de maneira aleatória em desenvolvimento e teste na proporção de 80% para desenvolvimento e 20% para teste de forma estratificada, ou seja, mantendo a proporção das classes. O conjunto de desenvolvimento é então utilizado para as etapas de otimização de hiperparâmetros e validação cruzada, enquanto o conjunto de teste é utilizado apenas para a etapa de validação final do modelo, cujos resultados são apresentados no Capítulo 4.

Uma visão geral do processo de divisão dos dados pode ser vista na Figura 9 a seguir.

Figura 8 – Execução do modelo



Fonte: Autor (2023)

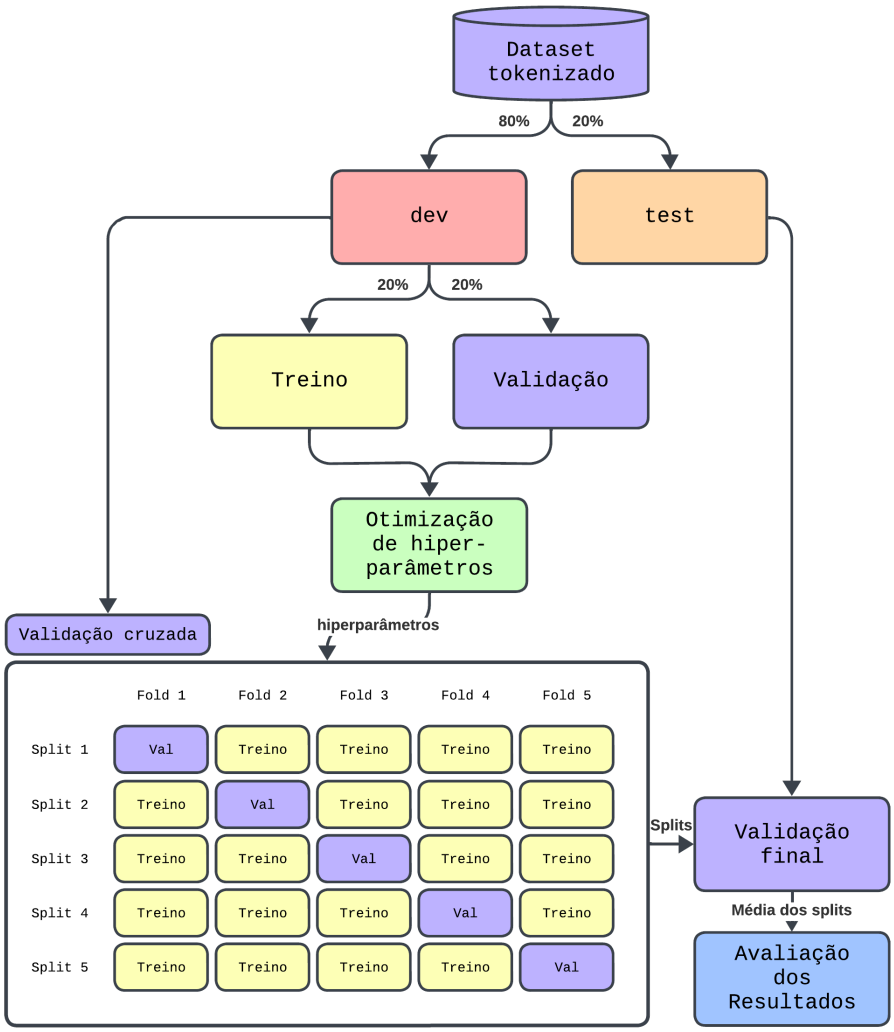
3.5 Otimização de Hiperparâmetros

Utilizando o conjunto de dados de desenvolvimento, composto de 80% dos dados totais, foi feito uma divisão estratificada de 20% para treinamento e 20% para validação na etapa de otimização para reduzir o tempo necessário de execução desta etapa mantendo uma quantidade razoável de tentativas.

Desta maneira, foi utilizado o algoritmo Tree-structured Parzen Estimator (TPE) (BERGSTRA et al., 2011) e a biblioteca Optuna (AKIBA et al., 2019) tendo como objetivo a otimização do valor de *macro F1-score* em 5 épocas por tentativa, por um total de 10 tentativas (ou *trials*) otimizando, para isto, os seguintes valores:

1. Taxa de aprendizado (*learning rate*), entre os valores 1×10^{-5} e 1×10^{-4};
2. Regularização (*weight decay*), entre os valores 0,0 e 0,1;

Figura 9 – Divisão dos dados



Fonte: Autor (2023)

3. Taxa de aquecimento (*warmup ratio*), também entre os valores 0,0 e 0,1;

Por conta do classificação ser *multi-class*, optou-se pela otimização da métrica *F1-score*, de acordo com o trabalho de (FORMAN; SCHOLZ, 2010).

Testes também foram realizados com os hiperparâmetros β_1 e β_2 do otimizador AdamW, porém obteve-se resultados insatisfatórios para a maior parte das tentativas, logo, para estes valores foi preferível utilizar os mesmos valores de Loshchilov e Hutter (2017).

3.6 Validação Cruzada

Utilizando os hiperparâmetros encontrados pelo otimizador e o conjunto de dados de desenvolvimento correspondente a 80% dos dados, foi realizado a técnica de validação cruzada estratificada chamada *K-Fold* com $K = 5$.

Ou seja, do conjunto de desenvolvimento, cada *split* tem uma divisão de 80% para treinamento e 20% para validação, isto é, 4 *folds* para treinamento e 1 *fold* para validação. Na Figura 9, cada quadrado no conjunto de validação cruzada corresponde a 1 *fold* ou, em outras palavras, 20% dos dados.

Os treinos de cada *split* foram realizados por no máximo 15 épocas cada, configurados com *Early Stopping* de 5 épocas para a métrica *macro F1-score* no conjunto de dados de validação do *split*, isto é, finalizando o treino caso não houvesse melhora na métrica por 5 épocas consecutivas ou chegasse ao final da 15ª época. Ao final de cada treino, os pesos da época de melhor *F1-score* são escolhidos para a próxima etapa.

3.7 Avaliação dos Resultados

Cada *split* da validação cruzada produz pesos diferentes para o modelo, gerando desta maneira 5 configurações diferentes do modelo que serão testadas com o conjunto de dados de teste. Por fim, é calculado a média e desvio padrão das métricas resultantes desse processo de validação no conjunto de teste.

As métricas escolhidas para apresentar os resultados da metodologia de cada LLM foram *loss*, acurácia, *F1-score*, ROC AUC, precisão e sensibilidade (*recall*), citadas na Seção 2.6, visto que são as métricas mais comumente encontradas em trabalhos de classificação. Todas as métricas, exceto *loss* e acurácia, utilizam a média macro (*macro average*) visto que essas versões são mais apropriadas quando as classes estão desbalanceadas no conjunto de dados (OPITZ, 2022).

3.8 Ambiente de Experimentação

O ambiente utilizado para o processamento dos dados e treinamento dos modelos foi o Google Colab, um serviço hospedado de Jupyter Notebook que não requer configuração para ser usado e fornece acesso gratuito a recursos de computação, incluindo unidades de processamento gráfico (GPUs, do inglês Graphics Processing Unit), voltado principalmente à ciência de dados. A GPU utilizada foi a Tesla T4 de 16GB de memória GDDR6.

Foram utilizadas diversas bibliotecas durante a implementação do projeto, tais como scikit-learn (PEDREGOSA et al., 2011), *pypdfium2* (PYPDFIUM2, 2021), Huggingface Transformer (WOLF et al., 2020), Optuna (AKIBA et al., 2019), PyTorch (PASZKE et al., 2019) e SciPy (VIRTANEN et al., 2020).

4 Resultados

Neste capítulo, serão apresentados e analisados os resultados obtidos por meio dos experimentos conduzidos, com ênfase na avaliação dos modelos contemplados nesta pesquisa. Os gráficos resultantes, fruto do processo experimental, são explorados proporcionando uma visão abrangente das conclusões derivadas da análise. A apresentação dos resultados busca oferecer uma compreensão detalhada do desempenho dos modelos.

4.1 Resultados Relacionados ao Conjunto de Dados

Os documentos recuperados da base de dados do Tribunal de Contas do Estado do Maranhão são organizados em 182 *tipos-documentos* distintos, entre balanços, balancetes, comparativos, demonstrativos, relatórios, extratos, notas, declarações, dentre outros.

Após a eliminação de documentos problemáticos, extração dos textos e da aplicação do pré-processamento nos textos extraídos, foram mantidos apenas os 11 tipos que apresentavam a maior quantidade de documentos, excluindo-se os demais registros do conjunto de dados empregado neste projeto, visto que o restante dos documentos não tinham quantidades suficientes para o aprendizado do classificador.

Como resultado desse processo, restou um total de 11.747 documentos dos 19.853 originais, distribuídos em 11 categorias distintas, conforme ilustrado na Tabela 1.

Tabela 1 – Classes dos documentos

Nº	**Nome**	**Quant.**	**%**
1	(DCASP) Balanço orçamentário	1225	10,43%
2	(DCASP) Balanço financeiro	1218	10,37%
3	(DCASP) Balanço patrimonial	1215	10,34%
4	(DCASP) Demonstração das variações patrimoniais	1206	10,27%
5	(DCASP) Demonstração dos fluxos de caixa	1218	10,37%
6	(DCASP) Demonstração das mutações do patrimônio líquido	963	8,20%
7	(DCASP) Notas explicativas	694	5,91%
8	Relatório e certificado de auditoria, com parecer do dirigente do órgão de controle interno	798	6,79%
9	Exposição circunstanciada da gestão	943	8,03%
10	Extratos e conciliações bancárias	1135	9,66%
11	Ofício de encaminhamento ao TCE/MA	1132	9,64%
-	**Total**	**11747**	**100%**

Foi calculado para o conjunto de dados o coeficiente de assimetria (ou *skew*) de Fisher-Pearson visto em Kokoska e Zwillinger (2000, Seção 2.2.24.1) para verificar o

desbalanceamento das classes, chegando-se ao valor de 0,023. Ou seja, estatisticamente foi possível afirmar que o desbalanceamento das classes é leve.

No total, há 1.295.529 diferentes sequências de texto de um total de 86.010.107 sequências. Na Tabela 2 há mais informações sobre o conjunto de dados utilizado neste trabalho em relação a quantidade de sequências, tamanho das sequências e quantidade de páginas.

Tabela 2 – Informação sobre os documentos

Informação	**Quantidade de sequências por documento**	**Tamanho de uma sequência**	**Quantidade de páginas por documento**
Menor	14	1	1
Mediana	398	7	3
Maior	1.218.778	128	9554
Moda	424	5	1
Média	7321,88	7,76	67,40
Desvio padrão	36.412,81	3,38	337,91

A Tabela 2 apresenta métricas para a análise de um conjunto de dados textual, fornecendo percepções acerca da estrutura e da distribuição das informações contidas nos documentos. Na linha *Menor*, destaca-se o valor mínimo observado de cada métrica, representando os extremos inferiores do conjunto. A linha *Mediana* revela a medida central, sendo útil para a compreensão da distribuição de dados. Já a linha *Maior* indica os valores máximos, oferecendo uma perspectiva sobre os extremos superiores dos dados. As estatísticas descritivas como *Moda*, *Média* e *Desvio padrão* fornecem informações sobre tendências centrais, variabilidade e concentração dos dados. Essas informações proporcionam uma visão abrangente das características textuais e estruturais do conjunto de dados em consideração.

Tabela 3 – Informação sobre as sequências nos documentos de tamanho ≥ 3

Sequência	**Quantidade**	**Sequência**	**Quantidade**
saldo	1.912.656	enviada	472.248
conta	1.053.841	aplicação	431.267
valor	768.908	atual	429.039
com	741.795	municipal	416.304
extrato	569.844	mês	366.361
ted	569.396	cota	346.847
banco	563.153	por	326.218
transferência	548.584	transf	318.534
para	492.926	referente	307.035
anterior	472.681	ano	305.332

Na Tabela 3, demonstram-se as sequências mais frequentes presentes no conjunto de dados utilizado neste trabalho. É perceptível que as sequências de maior recorrência

demonstram uma clara conexão com o conceito de prestação de contas. A visualização destes dados é fornecida na Figura 10, que apresenta uma representação gráfica na forma de uma nuvem de palavras, destacando assim as sequências mais proeminentes contidas no conjunto de dados examinado.

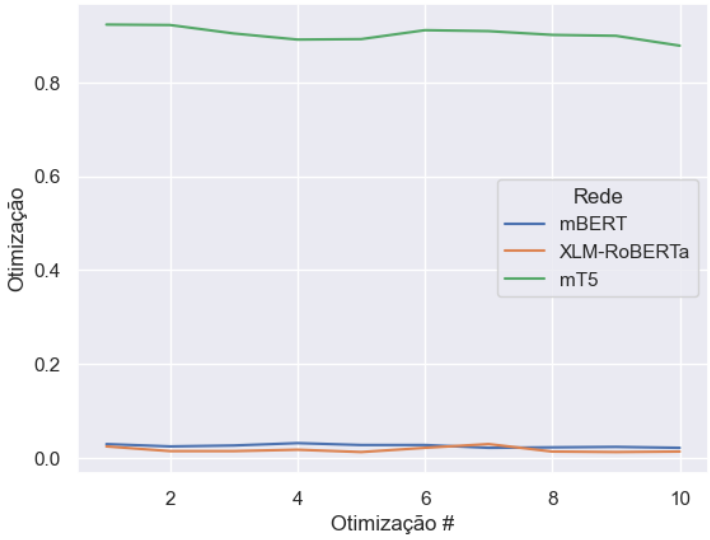
Figura 10 – Nuvem de palavras

Fonte: Autor (2023)

4.2 Resultados Relacionados à Classificação

Em um primeiro momento, conduziu-se uma fase inicial de otimização de hiperparâmetros, ilustrada na Figura 11 a seguir, com o intuito de selecionar os valores mais apropriados para cada modelo empregado.

Figura 11 – Otimização dos hiperparâmetros



Fonte: Autor (2023)

Como função objetivo da otimização dos hiperparâmetros foi utilizado o valor resultante da diferença entre 1 e o F1-score. Este processo foi realizado de forma individualizada para cada modelo e, como resultado dessa abordagem, foram identificados valores singulares para os hiperparâmetros de cada um, os quais estão apresentados na Tabela 4, evidenciando a distinção nas configurações ótimas obtidas em relação a cada um dos modelos analisados.

Tabela 4 – Hiperparâmetros escolhidos

Modelo	**Taxa de aprendizado**	**Regularização**	**Taxa de aquecimento**
mBERT	$4{,}95 \times 10^{-5}$	0,063	0,045
XLM-RoBERTa	$5{,}42 \times 10^{-5}$	0,097	0,081
mT5	$7{,}46 \times 10^{-5}$	0,034	0,028

A fim de aprimorar a confiabilidade dos resultados obtidos, optou-se pela aplicação da técnica de validação cruzada durante a realização dos experimentos. Essa estratégia visa mitigar a influência da variabilidade nos conjuntos de treinamento, validação e teste, proporcionando uma avaliação mais consistente do desempenho do modelo. Neste trabalho foi determinado o valor de $k = 5$, dividindo o conjunto de dados de desenvolvimento em 5 grupos de treino e validação.

Nas figuras que seguem, é possível observar os intervalos de valores associados aos cinco *splits* de treinamento realizados por cada modelo para o conjunto de validação do respectivo *split*, enquanto a reta central visível nos gráficos representa a média desses intervalos de valores, proporcionando uma visão mais abrangente das variações e tendências nos dados resultantes dos treinamentos executados pelos modelos nos diferentes *splits*.

Para cada *split*, os pesos que obtiveram o melhor F1-score são utilizados para calcular as métricas finais de cada modelo, resultando nas médias mencionadas nos parágrafos seguintes.

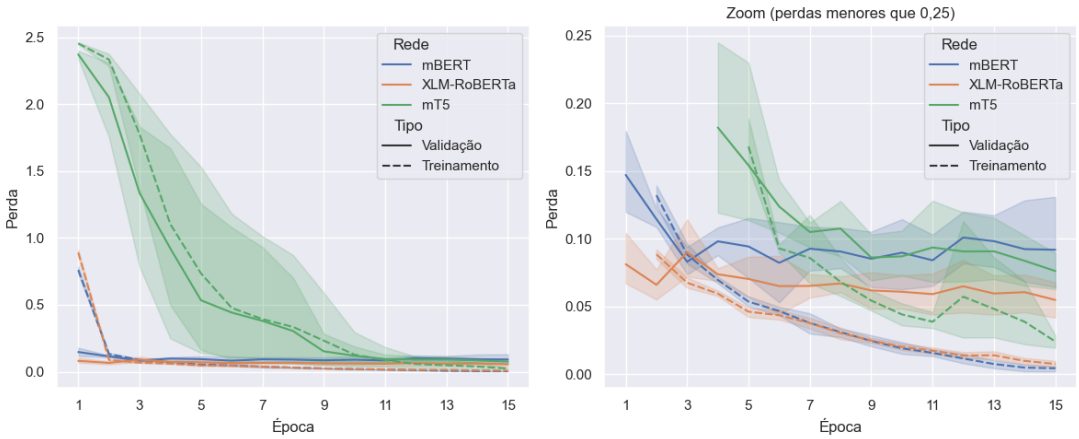
A Figura 12 mostra os valores das perdas (ou *loss*) das redes durante o treinamento.

Percebe-se, na Figura 12, que a perda para o treinamento é menor para o mBERT, mas é a maior para a validação. Isso sugere que o modelo aprendeu bem para o conjunto de treinamento, mas não generaliza tão bem quanto os outros modelos. Já o modelo mT5 apresentou uma grande variabilidade na perda, além de apenas conseguir competir com os outros modelos a partir da décima época, apresentando a pior perda para o treinamento e a segunda pior para a validação ao final da 15ª época.

A Figura 13 mostra os valores dos F1-scores dos modelos ao longo dos treinamentos em cada *split* da validação cruzada.

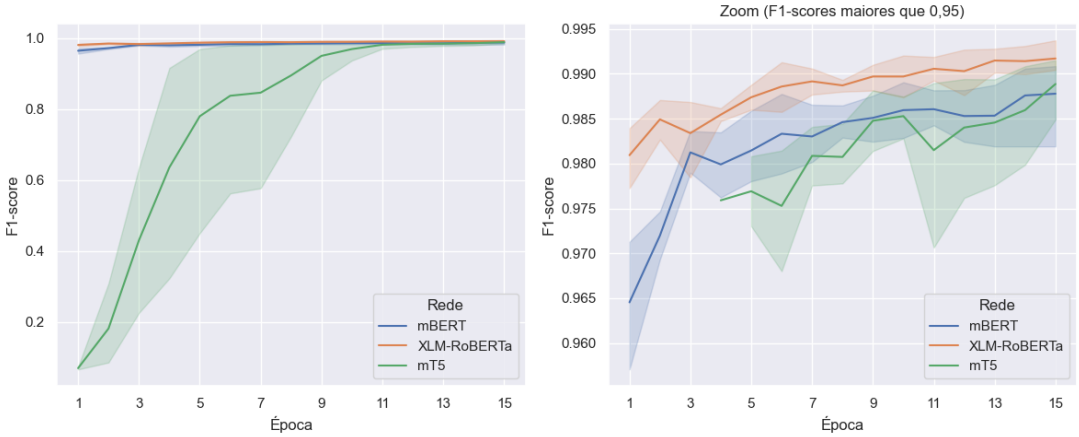
O F1-score foi a métrica utilizada no trabalho como objetivo principal e escolha do modelo final. Percebe-se pela Figura 13 uma clara vantagem do modelo XLM-RoBERTa,

Figura 12 – Gráficos das *losses* dos modelos ao longo das Épocas. À esquerda: gráfico completo. À direita: apenas perdas menores que 0,25



Fonte: Autor (2023)

Figura 13 – Gráficos dos F1-scores dos modelos ao longo das Épocas. À esquerda: gráfico completo. À direita: apenas valores maiores que 0,95



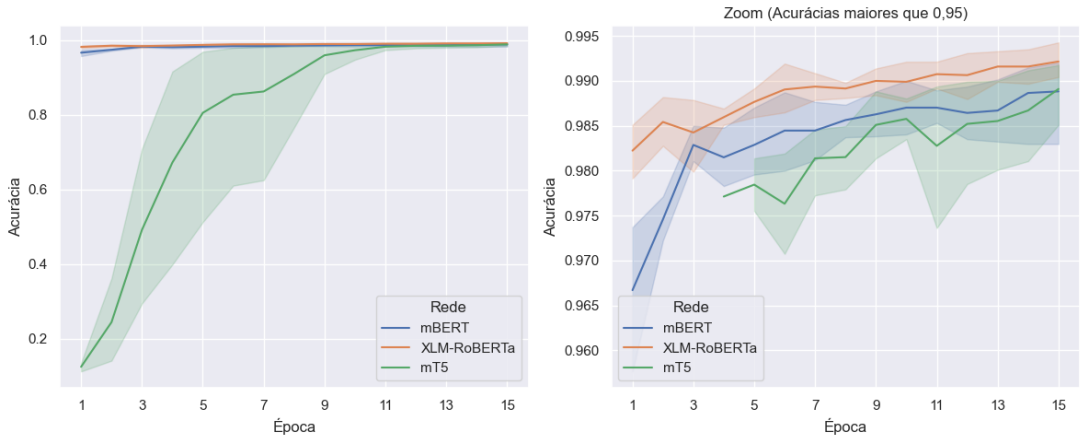
Fonte: Autor (2023)

que atinge o F1-score de $99{,}21\% \pm 0{,}16\%$ na validação, enquanto o mBERT alcançou $98{,}81\% \pm 0{,}35\%$, e o mT5 $98{,}63\% \pm 0{,}72\%$. Percebe-se, tanto pelo gráfico quanto pelo desvio padrão, que o modelo mT5 tem uma grande variação quando comparado aos outros modelos, enquanto o modelo XLM-RoBERTa alcança a melhor pontuação desde a primeira época.

Na Figura 14, o gráfico das acurácias dos modelos ao longo do treinamento mostra, novamente, uma vantagem do modelo XLM-RoBERTa em relação aos demais, obtendo uma pontuação de $99{,}22\% \pm 0{,}20\%$ de acurácia, enquanto o mBERT obteve $98{,}89\% \pm 0{,}35\%$, e o mT5 atingiu $98{,}71\% \pm 0{,}61\%$.

A Figura 15 apresenta informações relacionadas às métricas ROC AUC dos modelos analisados no estudo. É interessante notar que apesar do modelo mT5 exibir uma maior

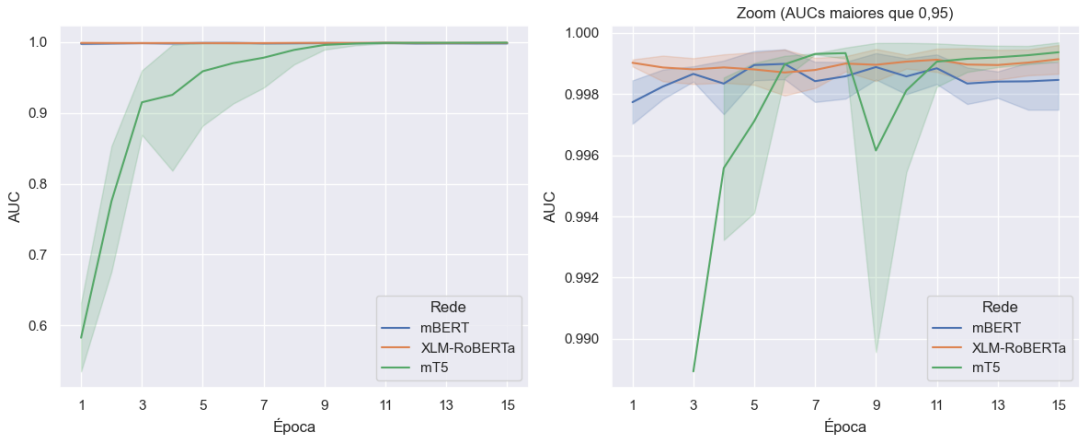
Figura 14 – Gráficos das acurácias dos modelos ao longo das Épocas. À esquerda: gráfico completo. À direita: apenas valores maiores que 0,95



Fonte: Autor (2023)

variabilidade nas primeiras épocas em comparação com os demais modelos, ao atingir o estágio final o modelo mT5 demonstrou um desempenho superior aos demais nesta métrica, atingindo o valor de $99{,}93\% \pm 0{,}04\%$, enquanto XLM-RoBERTa teve um resultado muito próximo, de $99{,}91\% \pm 0{,}07\%$, e o modelo mBERT conseguiu $99{,}88\% \pm 0{,}06\%$.

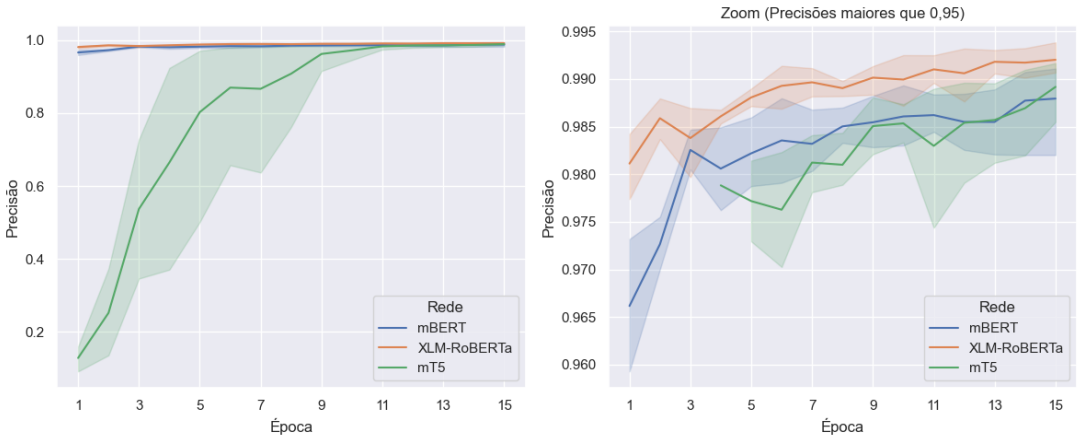
Figura 15 – Gráficos para ROC AUC dos modelos ao longo das Épocas. À esquerda: gráfico completo. À direita: apenas valores maiores que 0,95



Fonte: Autor (2023)

Na análise gráfica apresentada pela Figura 16, é possível observar uma vez mais uma superioridade do modelo XLM-RoBERTa em comparação com os demais, no que diz respeito à métrica de precisão. Este resultado sugere que o modelo XLM-RoBERTa não apenas mantém uma maior consistência, mas também destaca-se por sua capacidade de proporcionar resultados mais precisos e confiáveis em comparação com as alternativas consideradas no estudo. XLM-RoBERTa alcançou $99{,}24\% \pm 0{,}16\%$, enquanto mBERT obteve $98{,}83\% \pm 0{,}34\%$, e mT5 conseguiu $98{,}73\% \pm 0{,}56\%$.

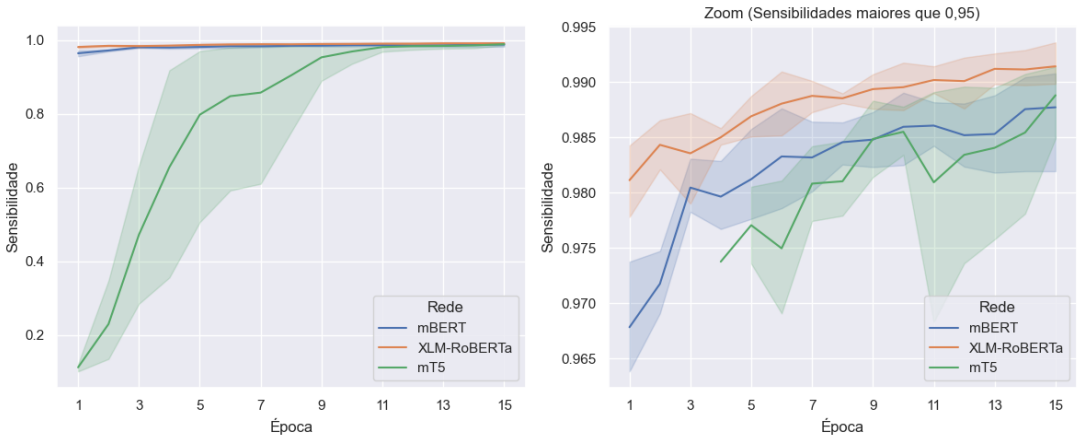
Figura 16 – Gráficos para as precisões dos modelos ao longo das Épocas. À esquerda: gráfico completo. À direita: apenas valores maiores que 0,95



Fonte: Autor (2023)

É possível constatar novamente um melhor desempenho do modelo XLM-RoBERTa na Figura 17, desta vez para a métrica de sensibilidade (ou *recall*). Este padrão sugere de maneira consistente que o modelo XLM-RoBERTa não apenas se destaca mas, também, mantém uma vantagem sobre os outros modelos em diversas métricas, atingindo o valor de 99,18% ± 0,17% para a sensibilidade. Já o modelo mBERT obteve 98,80% ± 0,36% e o modelo mT5 alcançou 98,58% ± 0,82%.

Figura 17 – Gráficos para as sensibilidades dos modelos ao longo das Épocas. À esquerda: gráfico completo. À direita: apenas valores maiores que 0,95



Fonte: Autor (2023)

Após a conclusão dos procedimentos de treinamento efetuados pelos modelos durante a etapa de validação cruzada, procedeu-se ao cálculo das métricas associadas aos pesos mais eficazes alcançados por cada modelo em cada iteração de *split* referente ao conjunto de teste. Dessa maneira, as médias resultantes para cada métrica estão registradas

e apresentadas nas Tabelas 5 e 6, que se seguem como uma representação consolidada dos desempenhos avaliados ao longo do processo.

Tabela 5 – Métricas do conjunto de validação

Modelo	Perda	F1-score	Acurácia	ROC AUC	Precisão	Sensibilidade
mBERT	7,31% ± 2,41%	98,81% ± 0,35%	98,89% ± 0,35%	99,88% ± 0,06%	98,83% ± 0,34%	98,80% ± 0,36%
mT5	8,51% ± 2,46%	98,63% ± 0,72%	98,71% ± 0,61%	**99,93% ± 0,04%**	98,73% ± 0,56%	98,58% ± 0,82%
XLM-RoBERTa	**5,48% ± 1,64%**	**99,21% ± 0,16%**	**99,22% ± 0,20%**	99,91% ± 0,07%	**99,24% ± 0,16%**	**99,18% ± 0,17%**

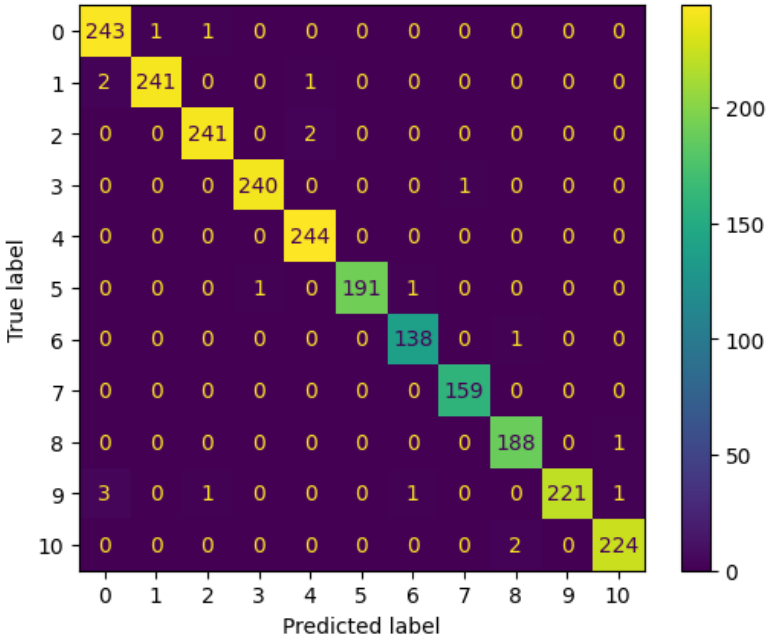
Tabela 6 – Métricas do conjunto de teste

Modelo	Perda	F1-score	Acurácia	ROC AUC	Precisão	Sensibilidade
mBERT	9,08% ± 1,80%	98,65% ± 0,29%	98,69% ± 0,29%	99,90% ± 0,05%	98,67% ± 0,27%	98,65% ± 0,30%
mT5	8,12% ± 2,13%	98,71% ± 0,75%	98,75% ± 0,67%	99,90% ± 0,01%	98,75% ± 0,67%	98,70% ± 0,80%
XLM-RoBERTa	**6,53% ± 0,77%**	**98,99% ± 0,12%**	**98,99% ± 0,12%**	**99,94% ± 0,02%**	**98,98% ± 0,12%**	**99,01% ± 0,12%**

O modelo XLM-RoBERTa obteve os melhores resultados em praticamente todas as métricas abordadas no trabalho, excetuando-se apenas a métrica ROC AUC na validação, onde o modelo mT5 conseguiu superar o XLM-RoBERTa, embora por uma margem relativamente mínima de apenas 0,02% na média, ressaltando-se que essa diferença permaneceu dentro dos limites estabelecidos pelo desvio padrão, reforçando assim a estreita competição entre esses modelos na obtenção de seus desempenhos.

Na Figura 18, a matriz de confusão do modelo XLM-RoBERTa do primeiro *split* para o conjunto de dados de teste mostra que a classe 9 (Extratos e conciliações bancárias) teve a maior quantidade de erros, 6 erros de um total de 227 documentos no teste, o que equivale a 2,64%, o que evidencia a capacidade de acerto do modelo.

Figura 18 – Matriz de confusão do XLM-RoBERTa



Fonte: Autor (2023)

Na lista a seguir, os tipos de documentos que correspondem a cada número do rótulo na matriz de confusão:

0. (DCASP) Balanço orçamentário
1. (DCASP) Balanço financeiro
2. (DCASP) Balanço patrimonial
3. (DCASP) Demonstração das mutações do patrimônio líquido
4. (DCASP) Demonstração das variações patrimoniais
5. (DCASP) Demonstração dos fluxos de caixa
6. (DCASP) Notas explicativas
7. Relatório e certificado de auditoria, com parecer do dirigente do órgão de controle interno
8. Exposição circunstanciada da gestão
9. Extratos e conciliações bancárias
10. Ofício de encaminhamento ao TCE/MA

É relevante destacar que os modelos em questão demonstraram desempenhos altamente satisfatórios. O modelo mBERT registrou o menor valor de F1-score para o conjunto de teste, atingindo um ainda notável patamar de $98{,}65\% \pm 0{,}29\%$. Este resultado, embora represente o ponto mais baixo alcançado entre os modelos, ainda se configura como uma avaliação bastante positiva e robusta para a métrica em questão, evidenciando a eficácia do mBERT na classificação de documentos.

5 Conclusão

Diante do exposto, este estudo propôs uma análise aprofundada sobre a eficácia e o desempenho de modelos de linguagem avançados, conhecidos como Large Language Models (LLMs), na tarefa específica de classificação de documentos relacionados à Prestação de Contas de gestores vinculados ao Tribunal de Contas do Estado do Maranhão. A proposta buscou avaliar como esses modelos, que são treinados em larga escala para entender e gerar texto natural, podem otimizar e aprimorar a automação do processo de análise e categorização de documentos contábeis e financeiros submetidos ao tribunal.

Após a análise dos dados e a implementação das estratégias delineadas no projeto de pesquisa, o estudo identificou que o modelo XLM-RoBERTa é o mais indicado para a tarefa de classificação dos documentos, visto que este modelo alcançou um F1-score de $98{,}99\% \pm 0{,}12\%$ no conjunto de dados do teste.

Ao realizar as validações e testes com os modelos abordados nesta pesquisa, constatou-se que todos os modelos apresentaram resultados favoráveis nas métricas exploradas. Isto realça o quão bom são as LLMs para a tarefa de classificação de documentos.

Ao longo desta pesquisa, o primeiro objetivo específico da pesquisa, foi alcançado. A coleta e organização dessas informações proporcionaram uma base sólida para as análises subsequentes, permitindo uma abordagem sistemática na aplicação de LLMs para a classificação desses documentos. A disponibilidade de um conjunto de dados representativo se revelou crucial para atingir os objetivos propostos nesta pesquisa, conferindo validade e solidez aos processos de avaliação propostos.

Continuando com os demais objetivos específicos da pesquisa, o objetivo delineado de empregar esses modelos no processo de categorização de documentos foi atingido, evidenciando a capacidade dessas tecnologias avançadas em lidar com a complexidade inerente aos dados contidos nos documentos analisados.

A realização de experimentos para avaliar o desempenho de cada LLM na classificação de documentos obteve resultado significativos. A análise abrangeu diversas métricas, incluindo acurácia, F1-score, ROC AUC, precisão e sensibilidade, proporcionando uma avaliação ampla e precisa do rendimento de cada modelo.

A análise dos resultados obtidos, alvo central desta pesquisa, corrobora com a realização dos objetivos estabelecidos. Esta etapa crucial validou as abordagens adotadas, contribuindo para a compreensão do papel e potencial das LLMs na análise documental e categorização. Este processo analítico representa não apenas um encerramento conclusivo

desta investigação, mas também abre portas para investigações futuras e a aplicação prática desses conhecimentos no contexto mais amplo da gestão documental e da tecnologia de processamento de linguagem natural.

A pesquisa é crucial para compreender o potencial desses modelos na área de auditoria e fiscalização, contribuindo para a eficiência e eficácia dos procedimentos de avaliação das Prestações de Contas na gestão pública do Estado do Maranhão.

Para pesquisas subsequentes, este estudo sugere diversas abordagens que podem aprimorar a compreensão e aplicação de LLMs na classificação de documentos. Em primeiro lugar, recomenda-se a avaliação de LLMs mais robustas e recentes, dotadas de bilhões de parâmetros, a fim de explorar o potencial desses modelos em uma escala ainda mais ampla. Além disso, sugere-se a incorporação de técnicas contemporâneas, como Document Image Classification (DIC), que vai além da análise textual ao classificar as páginas dos documentos como imagens, ampliando assim as perspectivas de análise. Adicionalmente, a pesquisa sugere a consideração não apenas do conteúdo textual, mas também da estrutura do texto, incluindo elementos como localização e estilo, como critérios para a classificação de documentos. Essa abordagem mais abrangente busca enriquecer a compreensão do desempenho e das capacidades dos LLMs em contextos mais diversificados e desafiadores.

Referências

ADHIKARI, A. et al. *DocBERT: BERT for Document Classification*. 2019. Citado na página 17.

ADHIKARI, A. et al. Rethinking complex neural network architectures for document classification. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4046–4051. Disponível em: <https://aclanthology.org/N19-1408>. Citado na página 17.

ADOBE. *About Adobe PDF*. 2023. Disponível em: <https://www.adobe.com/acrobat/about-adobe-pdf.html>. Acesso em: 2 de novembro de 2023. Citado na página 19.

AKIBA, T. et al. Optuna: A next-generation hyperparameter optimization framework. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.: s.n.], 2019. Citado 2 vezes nas páginas 36 e 38.

ALMEIDA, F.; XEXéO, G. *Word Embeddings: A Survey*. 2019. Citado na página 22.

APTé, C.; DAMERAU, F.; WEISS, S. M. Automated learning of decision rules for text categorization. *ACM Trans. Inf. Syst.*, Association for Computing Machinery, New York, NY, USA, v. 12, n. 3, p. 233–251, jul 1994. ISSN 1046-8188. Disponível em: <https://doi.org/10.1145/183422.183423>. Citado na página 17.

BA, J. L.; KIROS, J. R.; HINTON, G. E. *Layer Normalization*. 2016. Citado na página 21.

BENGIO, Y. et al. A neural probabilistic language model. *J. Mach. Learn. Res.*, JMLR.org, v. 3, n. null, p. 1137–1155, mar 2003. ISSN 1532-4435. Citado na página 22.

BERGSTRA, J. et al. Algorithms for hyper-parameter optimization. In: SHAWE-TAYLOR, J. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2011. v. 24. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf>. Citado na página 36.

BRASIL. *Constituição da República Federativa do Brasil de 1988*. 1988. Disponível em: <https://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm>. Citado 2 vezes nas páginas 19 e 20.

CAMACHO-COLLADOS, J.; PILEHVAR, M. T. *On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis*. 2017. Citado na página 33.

CLARK, K. et al. *What Does BERT Look At? An Analysis of BERT's Attention*. 2019. Citado na página 25.

CONNEAU, A. et al. *Unsupervised Cross-lingual Representation Learning at Scale*. 2019. Citado 2 vezes nas páginas 26 e 27.

DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2006. (ICML '06), p. 233–240. ISBN 1595933832. Disponível em: <https://doi.org/10.1145/1143844.1143874>. Citado 2 vezes nas páginas 29 e 30.

DEVLIN, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. Citado 4 vezes nas páginas 23, 24, 25 e 26.

FEIJO, D. de V.; MOREIRA, V. P. *Mono vs Multilingual Transformer-based Models: a Comparison across Several Language Tasks*. 2020. Citado na página 17.

FORMAN, G.; SCHOLZ, M. Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *SIGKDD Explor.*, v. 12, n. 1, p. 49–57, 2010. Disponível em: <http://dblp.uni-trier.de/db/journals/sigkdd/sigkdd12.html#FormanS10>. Citado na página 37.

GAGE, P. A new algorithm for data compression. *C Users J.*, R & D Publications, Inc., USA, v. 12, n. 2, p. 23–38, feb 1994. ISSN 0898-9788. Citado na página 24.

GRANDINI, M.; BAGLI, E.; VISANI, G. *Metrics for Multi-Class Classification: an Overview*. 2020. Citado na página 27.

HINTON, G.; VINYALS, O.; DEAN, J. *Distilling the Knowledge in a Neural Network*. 2015. Citado na página 17.

HUANG, Y. et al. *Advancing Transformer Architecture in Long-Context Large Language Models: A Comprehensive Survey*. 2023. Citado na página 21.

JABBAR, H. *MorphPiece : Moving away from Statistical Language Representation*. 2023. Citado na página 22.

KADDOUR, J. et al. *Challenges and Applications of Large Language Models*. 2023. Citado na página 24.

KHURANA, D. et al. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, v. 82, n. 3, p. 3713–3744, Jan 2023. ISSN 1573-7721. Disponível em: <https://doi.org/10.1007/s11042-022-13428-4>. Citado 2 vezes nas páginas 15 e 22.

KOKOSKA, S.; ZWILLINGER, D. *CRC standard probability and statistics tables and formulae, student edition*. London, England: CRC Press, 2000. Citado na página 39.

KUDO, T.; RICHARDSON, J. *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing*. 2018. Citado 2 vezes nas páginas 23 e 24.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, May 2015. ISSN 1476-4687. Disponível em: <https://doi.org/10.1038/nature14539>. Citado na página 20.

LENZA, P. *Direito constitucional esquematizado*. 15. ed. rev. atual. ampl. ed. São Paulo: Saraiva, 2020. Citado na página 15.

LIU, Y. et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. Citado na página 26.

LOSHCHILOV, I.; HUTTER, F. *Decoupled Weight Decay Regularization*. 2017. Citado na página 37.

MIKOLOV, T. et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. Citado 2 vezes nas páginas 22 e 23.

MIKOLOV, T. et al. *Distributed Representations of Words and Phrases and their Compositionality*. 2013. Citado 2 vezes nas páginas 22 e 23.

NAVEED, H. et al. *A Comprehensive Overview of Large Language Models*. 2023. Citado na página 15.

OPITZ, J. From bias and prevalence to macro f1, kappa, and mcc: A structured overview of metrics for multi-class evaluation. In: . [s.n.], 2022. Disponível em: <https://api.semanticscholar.org/CorpusID:253270558>. Citado na página 38.

PASZKE, A. et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. Citado na página 38.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado 2 vezes nas páginas 31 e 38.

PEñA, A. et al. Leveraging large language models for topic classification in the domain of public affairs. In: _____. *Lecture Notes in Computer Science*. Springer Nature Switzerland, 2023. p. 20–33. ISBN 9783031414985. Disponível em: <http://dx.doi.org/10.1007/978-3-031-41498-5_2>. Citado na página 17.

PYPDFIUM2. *pypdfium2*. 2021. Disponível em: <https://github.com/pypdfium2-team/pypdfium2>. Citado na página 38.

RADFORD, A. et al. Improving language understanding by generative pre-training. 2018. Citado na página 24.

RADFORD, A. et al. Language models are unsupervised multitask learners. 2019. Citado na página 26.

RAFFEL, C. et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2019. Citado na página 27.

RICHARDSON, C.; HECK, L. *Commonsense Reasoning for Conversational AI: A Survey of the State of the Art*. 2023. Citado na página 24.

SCHUSTER, M.; NAKAJIMA, K. Japanese and korean voice search. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2012. p. 5149–5152. Citado na página 23.

SELIYA, N.; KHOSHGOFTAAR, T. M.; HULSE, J. V. A study on the relationships of classifier performance metrics. In: *2009 21st IEEE International Conference on Tools with Artificial Intelligence*. [S.l.: s.n.], 2009. p. 59–66. Citado 3 vezes nas páginas 27, 28 e 29.

SENNRICH, R.; HADDOW, B.; BIRCH, A. *Neural Machine Translation of Rare Words with Subword Units*. 2016. Citado 2 vezes nas páginas 24 e 26.

SONG, D. et al. Multi-label legal document classification: A deep learning-based approach with label-attention and domain-specific pre-training. *Inf. Syst.*, Elsevier Science Ltd., GBR, v. 106, n. C, may 2022. ISSN 0306-4379. Disponível em: <https://doi.org/10.1016/j.is.2021.101718>. Citado na página 17.

TCE/MA. *e-PCA - Sistema de Prestação de Contas Anual Eletrônica*. 2023. Disponível em: <https://app.tcema.tc.br/hotsites/epca>. Acesso em: 1 de dezembro de 2023. Citado na página 15.

TCE/MA. *INSTRUÇÃO NORMATIVA TCE/MA Nº 52, DE 25 DE OUTUBRO DE 2017*. 2023. Disponível em: <https://app.tcema.tc.br/publicacao/#/documentohtml/894>. Acesso em: 1 de dezembro de 2023. Citado na página 15.

TCE/MA. *Sistema de Prestação de Contas Anual eletrônica (ePCA) já está disponível aos usuários*. 2023. Disponível em: <https://www.tcema.tc.br/index.php/noticias/2521>. Acesso em: 1 de dezembro de 2023. Citado 2 vezes nas páginas 15 e 19.

TENNEY, I.; DAS, D.; PAVLICK, E. *BERT Rediscovers the Classical NLP Pipeline*. 2019. Citado na página 25.

TORAMAN, C. et al. Impact of tokenization on language models: An analysis for turkish. *ACM Transactions on Asian and Low-Resource Language Information Processing*, Association for Computing Machinery (ACM), v. 22, n. 4, p. 1–21, mar. 2023. ISSN 2375-4702. Disponível em: <http://dx.doi.org/10.1145/3578707>. Citado 2 vezes nas páginas 22 e 23.

VASWANI, A. et al. *Attention Is All You Need*. 2017. Citado 4 vezes nas páginas 20, 21, 22 e 24.

VIRTANEN, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, v. 17, p. 261–272, 2020. Citado na página 38.

WAN, L. et al. *Long-length Legal Document Classification*. 2019. Citado 2 vezes nas páginas 15 e 17.

WANG, Q. et al. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, v. 9, n. 2, p. 187–212, Apr 2022. ISSN 2198-5812. Disponível em: <https://doi.org/10.1007/s40745-020-00253-5>. Citado na página 28.

WOLF, T. et al. Transformers: State-of-the-art natural language processing. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, 2020. p. 38–45. Disponível em: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>. Citado na página 38.

WU, Y. et al. *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. 2016. Citado 3 vezes nas páginas 23, 25 e 33.

XUE, L. et al. *mT5: A massively multilingual pre-trained text-to-text transformer*. 2020. Citado na página 27.

YADAV, S.; SHUKLA, S. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In: *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. [S.l.: s.n.], 2016. p. 78–83. Citado na página 31.

YANG, P. et al. SGM: Sequence generation model for multi-label classification. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018. p. 3915–3926. Disponível em: <https://aclanthology.org/C18-1330>. Citado na página 17.

ZHAO, L. et al. *Classification of Natural Language Processing Techniques for Requirements Engineering*. 2022. Citado na página 33.

ZHENG, H. et al. *Learn From Model Beyond Fine-Tuning: A Survey*. 2023. Citado na página 20.