UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA

NICOLAS TOLEDO DE CAMARGO

Estudo comparativo de modelos de linguagem e seus ajustes para análise de sentimentos em notícias financeiras

Campinas
22/11/2024

NICOLAS TOLEDO DE CAMARGO

Estudo comparativo de modelos de linguagem e seus ajustes para análise de sentimentos em notícias financeiras

Monografia apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos para obtenção de créditos na disciplina Projeto Supervisionado, sob a orientação do(a) Prof. João B. Florindo.

Resumo

Neste relatório, investigamos diferentes modelos de linguagem aplicados à análise de sentimentos em manchetes financeiras, explorando comparativamente aspectos teóricos e práticos. Avaliamos a qualidade de modelagem e a eficiência dos modelos BERT, Pythia, Mamba e LLaMa3.1 na classificação de manchetes financeiras em rótulos "positivo", "neutro"ou "negativo". Os modelos foram testados em suas versões base e, posteriormente, ajustados utilizando a técnica LoRA.

Utilizando um conjunto de dados elaborado por especialistas, a nossa avaliação incluiu a medição do F1-score, além da análise dos tempos de treinamento e inferência.

Os resultados mostram que todos os modelos ajustados com LoRA superaram significativamente suas versões base, produzindo resultados satisfatórios. O LLaMa3.1 demonstrou o melhor desempenho geral, embora sua grande quantidade de parâmetros tenha impactado negativamente sua eficiência em comparação com modelos menores. O BERT destacou-se por sua qualidade superior entre os modelos de tamanho similar, enquanto o Mamba, apesar de alcançar uma qualidade competitiva, não se mostrou eficiente para nossa tarefa.

Por fim, sugerimos que a escolha do modelo pode depender das prioridades da aplicação, sendo que para nossos experimentos concluímos que: o LLaMa3.1 é a melhor opção para qualidade, o Pythia para maior velocidade de treinamento, e o BERT para menor custo de inferência.

Abstract

In this report, we investigate different language models applied to sentiment analysis in financial headlines, comparatively exploring theoretical and practical aspects. We evaluated the modeling quality and efficiency of the BERT, Pythia, Mamba, and LLaMa3.1 models in classifying financial headlines into "positive", "neutral" or "negative" labels. The models were tested in their base versions and subsequently fine-tuned using the LoRA technique.

Using a dataset developed by experts, our evaluation included measuring the F1-score, as well as analyzing training and inference times.

The results show that all models fine-tuned with LoRA significantly outperformed their base versions, producing satisfactory results. LLaMa3.1 demonstrated the best overall performance, although its large number of parameters negatively impacted its efficiency compared to smaller models. BERT stood out for its superior quality among models of similar size, while Mamba, despite achieving competitive quality, did not prove efficient for our task.

Finally, we suggest that the choice of model may depend on the application's priorities. For our experiments, we concluded that LLaMa3.1 is the best option for quality, Pythia for faster training speed, and BERT for lower inference cost.

Conteúdo

1 Introdução

A análise de sentimentos tornou-se uma ferramenta cada vez mais essencial para investidores nos mercados financeiros, onde a identificação do tom emocional de notícias, manchetes ou relatórios oferece percepções valiosas para a tomada de decisões. A capacidade de determinar com precisão o sentimento em manchetes financeiras pode impactar significativamente a avaliação de tendências de mercado, gestão de riscos e alocação de portfólios. E com os avanços recentes nos modelos de processamento de linguagem natural, o uso de modelos pré-treinados tem demonstrado grande potencial para melhorar a eficácia desta tarefa.

Em nosso trabalho anterior (Nicolas T. de Camargo, 2024), focamos no ajuste fino de um modelo de linguagem pré-treinado específico, o LLaMa2, dando ênfase à exploração da sensibilidade dos hiperparâmetros do modelo para o ajuste à tarefa. Essa abordagem proporcionou *insights* sobre a sensibilidade do ajuste fino e a validação do processo para essa tarefa, resultando em uma melhoria de aproximadamente 2,6 vezes em relação ao modelo base, atingindo um F1-score de 0,867. Como sugestão para trabalhos futuros, propusemos explorar diferentes modelos para avaliar também seus desempenhos, e esse será o foco deste projeto.

Dessa forma, investigaremos o ajuste fino de um conjunto diversificado de modelos de linguagem pré-treinados. Os modelos selecionados para este estudo foram escolhidos para representar uma ampla gama de arquiteturas, tarefas de treinamento e número de parâmetros. Nosso objetivo é fornecer uma análise comparativa, tanto teórica quanto prática, desses modelos, destacando suas características relativas, quando tais modelos são aplicados à tarefa especializada de análise de sentimentos financeiros.

2 Trabalhos relacionados

Ajustes em modelos de linguagem. Modelos de linguagem em sua forma pré-treinada podem ser inadequados para tarefas específicas; no entanto, é possível torná-los mais eficientes através do ajuste fino. Cheonsu Jeong (2024) discute a contínua produção de modelos pré-treinados sem um estudo aprofundado sobre seus ajustes. O artigo enfatiza a importância de adaptar modelos de linguagem a domínios específicos,

apresentando exemplos práticos de como realizar ajustes finos, incluindo coleta de *dataset*, preprocessamento, escolha de modelos e outras considerações. O estudo oferece *insights* para futuros trabalhos nessa área.

Ajustes no Mamba. Trabalhos relacionados ao Mamba ainda são escassos (até a data deste estudo), por ser uma arquitetura recente, especialmente no domínio da análise de sentimentos financeiros. No entanto, alguns estudos iniciais sobre ajustes no Mamba são relevantes. Em Lucas Brennan-Almaraz & Daniel Guo & Sarvesh Babu (2024), os autores exploram o modelo em sua fase inicial, corrigindo *bugs* e desenvolvendo *scripts* próprios para ajustá-lo, comparando-o com modelos baseados em atenção em um *dataset* de perguntas e respostas. Posteriormente, Zhichao Xu (2024) utilizou a versão disponível no HuggingFace para ajuste em classificação de documentos, comparando-o com modelos do estado da arte. Ambos os estudos demonstram boa qualidade de modelagem, com ressalvas sobre eficiência e estabilidade, abrindo caminho para futuras explorações dessa arquitetura, incluindo sua aplicação em nosso domínio.

Modelos de linguagem adaptados para finanças. O domínio financeiro é amplamente explorado nas áreas de aprendizado de máquina e aprendizado profundo, sendo a análise de sentimentos financeiros um tópico de destaque, que ganhou novas abordagens com o advento de modelos de linguagem. O trabalho de Dogu Araci (2019) foi o primeiro a aplicar o BERT à análise de sentimentos financeiros, criando o FinBERT. O estudo envolveu o treinamento adicional do BERT em textos financeiros e ajuste fino para a tarefa de classificação de sentimentos, alcançando resultados superiores a estudos anteriores. Posteriormente, Pau Rodriguez Inserte et al. (2024) adaptou modelos menores (menos de 1,5B parâmetros), com uma estratégia semelhante, demonstrando que modelos pequenos ajustados superaram até modelos maiores de uso geral, ressaltando o ganho de eficiência proporcionado por essa troca. Já em Thanos Konstantinidis et al. (2024), os autores ajustaram o LLaMa2 para classificar sentimentos financeiros, além de medir a intensidade dos sentimentos, criando o FinLLaMa. O estudo também introduziu uma nova forma de avaliar os ajustes, utilizando métricas práticas de retorno em simulações de *trading*, proporcionando a discussão sobre as diferenças entre avaliações acadêmicas e aplicações práticas. Em nosso trabalho anterior, exploramos o ajuste fino no LLaMa2, focando na estabilidade com a variação de hiperparâmetros como número de épocas, taxas

de aprendizado e temperatura.

Esses estudos mostram que o tópico continua relevante e há ainda espaço para mais contribuições, o que motiva a continuidade do nosso trabalho com modelos de linguagem, e em específico, no domínio de análise de sentimentos financeiros.

3 Revisão teórica

Esta seção apresentará os principais mecanismos de computação utilizados pelos modelos, introdução às arquiteturas *encoder* e *decoder*, o método de ajuste fino e as métricas adotadas.

3.1 Mecanismos principais

Para os cálculos de complexidades, considere: B como o tamanho do *batch*, L o tamanho da sequência, D o tamanho do *embedding* e N o tamanho do estado escondido.

Auto-atenção: Apresentada inicialmente na arquitetura Transformer (Ashish Vaswani et al., 2017), a auto-atenção baseia-se no processo em que cada palavra possui vetores a representam (*query*, *key* e *value*), obtidos pelo produto dos *embeddings* por matrizes de pesos. Com o produto interno destes vetores, obtemos uma medida da relação entre cada palavra. Esse mecanismo é atualmente o mais utilizado em modelos de processamento de linguagem natural devido à sua capacidade de captar o contexto de forma eficaz.

$$Q = XW_q \qquad K = XW_k \qquad V = XW_v \tag{1}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{2}$$

Onde Q, K e V são as matrizes que contêm os *queries*, *keys* e *values* de todas as palavras da sequência de *embeddings* X.

Na equação (1), temos $X \in \mathbb{R}^{L\times D}$ e $W \in \mathbb{R}^{D\times N}$, resultando em uma complexidade de $\mathcal{O}(LND)$ FLOPs. Na equação (2), as dimensões de Q, K e V são $\in \mathbb{R}^{L\times N}$, o que gera uma complexidade de $\mathcal{O}(L^2N)$ FLOPs. Ao combinar (1) e (2), temos uma complexidade total de $\mathcal{O}[B(LND + L^2N)]$ FLOPs. Em termos de L, a complexidade é $\mathcal{O}(L^2)$ tanto para o treinamento quanto para a inferência. No entanto, ao utilizar o *cache* KV (como no LLaMa), é possível reduzir a complexidade para $\mathcal{O}(L)$ FLOPs por *token* gerado durante a inferência. Além disso, como todas as operações envolvem multiplicações matriciais, o processo de treinamento é altamente paralelizável.

Structured State-Space: Os Modelos de Espaços de Estado Estruturados (Albert Gu et al., 2021), se baseiam nas equações de recorrência que derivam da dis-

cretização de um modelo de espaço estado. Estes SSMs (*State Space Models*) foram desenvolvidos para atingir uma qualidade comparável ao mecanismo de atenção, mas com menor complexidade computacional. Contudo, na prática, eles não alcançaram resultados tão satisfatórios devido à menor capacidade de capturar o contexto em comparação à atenção.

$$\begin{cases} h_t = Ah_{t-1} + Bx_t \\ y(t) = Ch_t \end{cases} \tag{3}$$

Onde A, B e C são pesos treináveis, e x_t, h_t e $y(t)$ representam a entrada, o estado escondido e a saída no passo t, respectivamente.

As dimensões das variáveis em (3) são: $x \in \mathbb{R}^{D\times 1}$, $h \in \mathbb{R}^{N\times 1}$, $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times D}$ e $C \in \mathbb{R}^{D\times N}$, resultando em uma complexidade de $\mathcal{O}(N^2 + ND)$ FLOPs por passo de inferência e $\mathcal{O}[BL(N^2 + ND)]$ FLOPs para o treinamento. Em termos de L, temos uma complexidade de $\mathcal{O}(L)$ para o treinamento e $\mathcal{O}(1)$ para a inferência por *token*. Embora a recorrência não permita um treinamento paralelizável a princípio, ao expandir a relação, é possível formular um *kernel* de convolução, tornando o processo paralelizável no treinamento.

$$K = (CB, CAB, \ldots, CA^{n-1}B) \qquad y = x * K \tag{4}$$

Mamba: Introduzido em (Albert Gu; Tri Dao, 2023), a principal modificação em relação aos SSMs Estruturados é a introdução de parâmetros treináveis variantes no tempo nas equações de recorrência, permitindo que o modelo selecione os *inputs* mais relevantes, criando então um modelo SSM Estruturado e Seletivo.

$$\begin{cases} h_t = A_t h_{t-1} + B_t x_t \\ y(t) = C_t h_t \end{cases} \tag{5}$$

Inicialmente, essa modificação causa a perda da propriedade de paralelização via convolução. No entanto, os autores mostraram que é possível aproveitar a associatividade das operações e utilizar um algoritmo de *scan* (Figura 5), mantendo o treinamento paralelizável e com complexidade de $\mathcal{O}(L)$.

3.2 Encoder-only

Nessas arquiteturas, a informação do *input* é processada de forma bidirecional, com o objetivo de extrair o máximo de informações relevantes e transformá-las em uma representação compacta. Estes espaços latentes podem então ser utilizados em uma variedade de tarefas, como por exemplo em camadas classificadoras, treinando um modelo para prever uma palavra faltante no texto (*Masked Language Modeling*) ou classificar sentenças (*Next Sentence Prediction*), como ocorre no BERT.

3.3 Decoder-only

Nessas arquiteturas, a informação é processada de forma unidirecional, com o objetivo de gerar uma saída baseada apenas no contexto anterior. Esses modelos, também chamados de modelos causais (*Causal Language Modeling*), são projetados para prever a próxima palavra em uma sentença e, de forma auto-regressiva, alimentar o *decoder* com essa nova palavra, continuando a geração de texto até a sinalização do fim da sentença (*token* [EOS]).

3.4 LoRA

O ajuste fino com eficiência de parâmetros (*Parameter-Efficient Fine-Tuning*) é um método que permite congelar parte dos parâmetros originais de um modelo, treinando apenas algumas camadas, o que torna o processo mais acessível computacionalmente.

O LoRA (*Low-Rank Adaptation*) (Edward J. Hu et al., 2021) é uma abordagem PEFT. A técnica consiste em congelar todos os parâmetros originais do modelo e adicionar novos parâmetros em camadas específicas. Esses parâmetros adicionados têm menor dimensão com relação aos originais, necessitando de um menor custo computacional para serem treinados, pois são representados por matrizes de posto baixo. Em vez de aprender a matriz original completa $W_0 \in \mathbb{R}^{d\times k}$, o método treina duas matrizes menores: $A \in \mathbb{R}^{d\times r}$ e $B \in \mathbb{R}^{r\times k}$, onde $r << \min(d, k)$. Então essas matrizes são multiplicadas e adicionadas à camada.

$$W_0 + \delta W = W_0 + AB \tag{6}$$

Ao ter $d \times k >> d \times r + r \times k$, o custo computacional é reduzido (basta escolher $r < \frac{d \times k}{d+k}$). Por exemplo, se $d = k = 512$ e $r = 1$, temos $d \times k = 262144 >> d \times r + r \times k = 1024$.

3.5 Métricas

Precision: corresponde à fração de valores preditos que verdadeiramente pertencem a uma classe, em relação a todos os valores que foram preditos como pertencentes a essa classe. Interpreta o quanto os valores preditos como positivos estão corretos.

$$Precision = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos positivos}} \tag{7}$$

Recall: corresponde à fração de valores preditos que verdadeiramente pertencem a uma classe, em relação a todos os valores que verdadeiramente pertencem a essa classe. Interpreta a capacidade do modelo de identificar corretamente os exemplos da classe positiva.

$$Recall = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos negativos}} \tag{8}$$

F1-score: é útil por considerar tanto os falsos positivos quanto os falsos negativos. É calculada como a média harmônica entre *precision* e *recall*.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

Onde todas estas medidas variam de zero a um.

4 Metodologia

4.1 Modelos base

	BERT	**Pythia**	**Mamba**	**LLaMa3.1**
Mecanismo	Auto-Atenção	Auto-Atenção	SSM Seletivo	Auto-Atenção
Arquitetura	Encoder	Decoder	Decoder	Decoder
Tarefa	Classificação	Causal	Causal	Causal
Parâmetros	335M	410M	370M	8B

Tabela 1: Principais características dos modelos selecionados.

Os modelos apresentados foram escolhidos por serem *open-source* e por sua diversidade em número de parâmetros, arquiteturas, mecanismos de processamento e tarefas principais.

O BERT (Bidirectional Encoder Representations from Transformers) (Jacob Devlin et al., 2018) é um modelo de linguagem baseado em camadas de *encoder* do Transformer, utilizado para extração de *features*. Esse modelo se destacou por empregar o mecanismo de atenção para capturar o contexto de forma bidirecional, permitindo um alto nível de processamento de frases. Isso o torna ideal para especialização em tarefas específicas por meio de ajuste fino ao adicionar uma camada linear classificadora de saída.

O Pythia (Stella Biderman et al., 2023) faz parte de uma família de modelos de linguagem causais com diferentes números de parâmetros, baseado em camadas de *decoder* do Transformer. O objetivo principal dessa família de modelos é fomentar a pesquisa científica em modelos de linguagem, em vez de buscar a melhor performance absoluta.

O LLaMa3.1 (Abhimanyu Dubey et al., 2024) é um *decoder-only transformer*, com uma arquitetura semelhante às versões anteriores da família LLaMa, mas com ajustes em hiperparâmetros e melhorias na qualidade dos dados de treinamento. A série LLaMa tem se tornado uma referência de qualidade entre os modelos *open-source* no estado da arte.

O Mamba (Albert Gu; Tri Dao, 2023) também é um modelo causal, porém baseado no mecanismo de SSM Seletivo, uma nova arquitetura derivada do S4. O Mamba foi introduzido com a expectativa de se tornar uma arquitetura rival aos *transformers*,

onde o *paper* apresenta resultados comparáveis e com menor complexidade computacional, especialmente em contextos e geração de texto de sequências muito longas.

4.2 Experimentos

Objetivo: Cada modelo classificará as notícias financeiras do *dataset* nos rótulos "positivo", "negativo" ou "neutro".

Avaliações: Os modelos serão avaliados em sua versão base e, posteriormente, em suas versões ajustadas. As avaliações abrangerão os aspectos de **qualidade** e **eficiência**.

1) Qualidade: A qualidade dos modelos será medida pelo F1-score no conjunto de teste, utilizando a versão ajustada que alcançou o maior F1-score no conjunto de validação.

2) Eficiência: A eficiência será avaliada com base nos tempos de treinamento e inferência dos modelos.

4.3 Fonte de dados

Os dados utilizados para treino e teste estão estruturados em uma tabela que contém as notícias e seus respectivos sentimentos (positivo, neutro ou negativo). O conjunto de dados, denominado *FinancialPhraseBank*, consiste em 4.837 classificações de notícias financeiras em inglês. Trata-se de um conjunto desbalanceado, com aproximadamente 59% de notícias neutras, 28% de positivas e 12% de negativas. As classificações foram realizadas por especialistas em finanças, incluindo pesquisadores e estudantes de mestrado da *Aalto University School of Business*.

4.4 Configuração Experimental

O *dataset* foi balanceado por *downsampling* no menor número de exemplos (rótulo "negativo") e dividido mantendo a mesma quantidade de cada sentimento. Os conjuntos de divisão são: 70% dos dados destinados ao treino, 10% à validação e 20% ao teste. Para o BERT, os textos e rótulos podem ser diretamente utilizados, já que o utilizamos com uma camada classificadora na saída. No entanto, para os modelos causais,

é necessário ajustar os dados para o formato de *prompt*, induzindo o modelo a continuar a frase.

Padrão de prompt de treinamento para modelos causais

Analyze the sentiment of the news headline enclosed in square brackets, determine if it is positive, neutral, or negative, and return the answer as the corresponding sentiment label "positive," "neutral," or "negative".
$[HEADLINE] = SENTIMENT$

Padrão de prompt de teste para modelos causais

Analyze the sentiment of the news headline enclosed in square brackets, determine if it is positive, neutral, or negative, and return the answer as the corresponding sentiment label "positive," "neutral," or "negative".
$[HEADLINE] =$

Os ajustes dos modelos serão realizados utilizando a técnica LoRA, com os mesmos hiperparâmetros aplicados a todos. A única exceção será a quantização do modelo LLaMa, devido às limitações de memória da GPU disponível. Os principais hiperparâmetros compartilhados incluem: número máximo de épocas, tamanho do *batch*, posto do LoRA, camadas alvo para ajuste, taxa de aprendizado e o tamanho máximo do contexto.

Hiperparâmetros	**Valor**
Número máximo de épocas	10
Tamanho do batch	8
LoRA Rank	64
Módulos alvo	Camadas lineares
Taxa de aprendizado	0.0002
Tamanho de contexto	512

Tabela 2: Hiperparâmetros compartilhados nos ajustes.

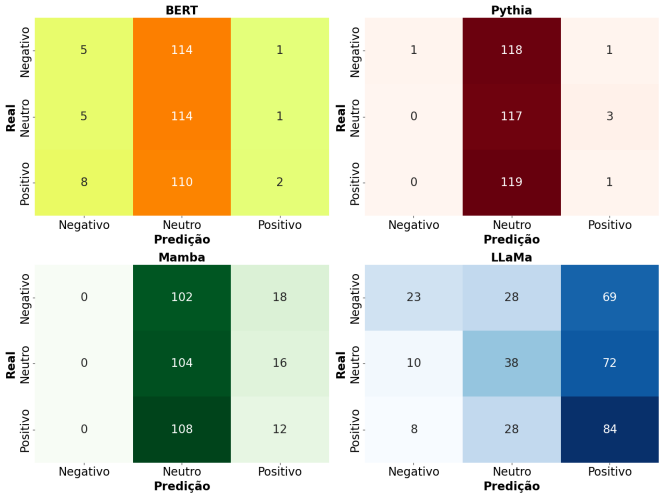
Os experimentos serão executados na plataforma Kaggle utilizando uma GPU T4, com os modelos pré-treinados carregados do HuggingFace. Os *notebooks* com os testes estão disponíveis em https://www.kaggle.com/nicolastdc/code.

5 Resultados

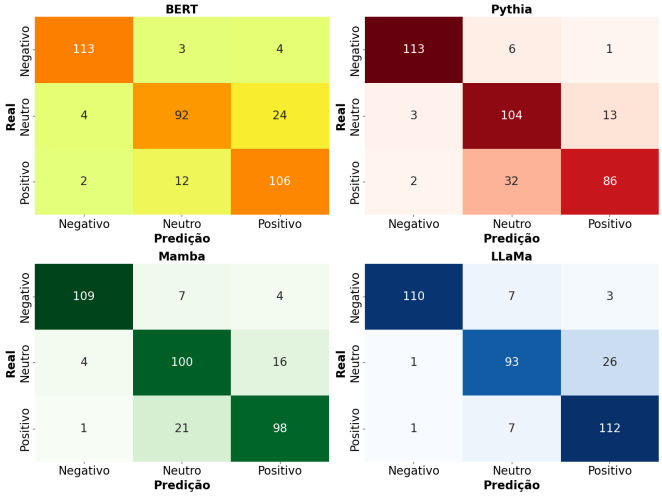
Resultados

Modelo	Base	Ajustado
BERT	0.200	<u>0.864</u>
Pythia	0.175	0.842
Mamba	<u>0.208</u>	0.853
LLaMa	**0.376**	**0.875**

Tabela 3: F1-score no conjunto de teste para os modelos base e ajustados. Os melhores resultados estão em negrito, e os segundo melhores resultados estão sublinhados.

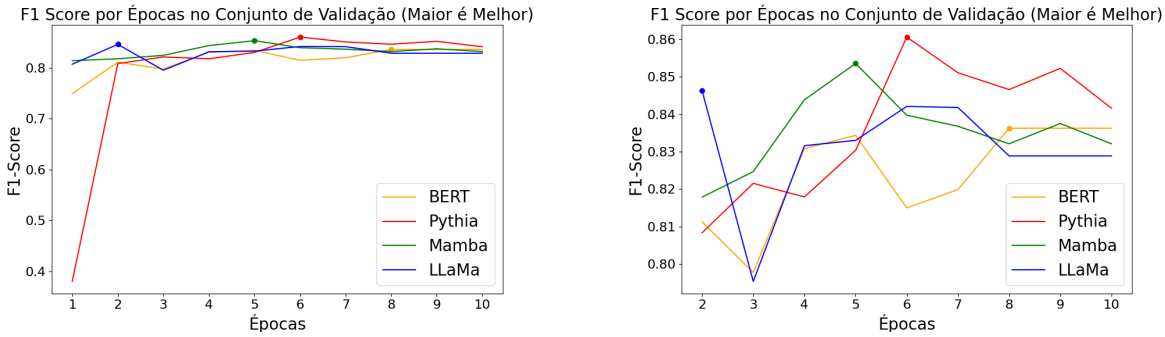


(a) Modelos base.



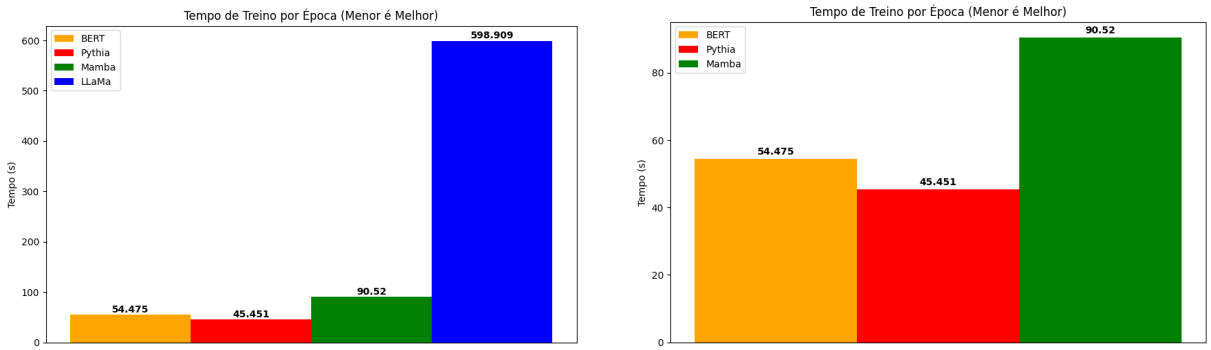
(b) Modelos ajustados.

Figura 1: Matrizes de confusão no conjunto de teste.



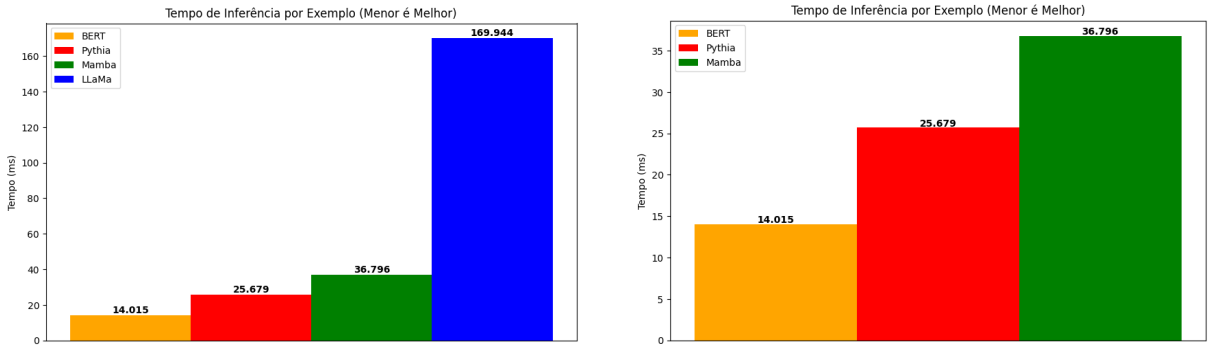
(a) Todas as épocas. (b) Ampliado a partir da segunda época.

Figura 2: Gráficos do F1-score por época no conjunto de validação, com destaque para os melhores valores.



(a) Todos os modelos. (b) Apenas os modelos menores.

Figura 3: Gráficos do tempo de treino por época.



(a) Todos os modelos. (b) Apenas os modelos menores.

Figura 4: Gráficos do tempo de inferência por exemplo.

6 Discussão

Ao avaliar a Tabela 3 e as Matrizes 1a, é evidente que os modelos base não são capazes de produzir bons resultados para nossa tarefa. Observa-se também que o BERT, Pythia e Mamba tendem a classificar as notícias como neutras, enquanto o LLaMa apresenta um viés em classificá-las como positivas.

A classificação do modelo base BERT foi feita com uma camada classificadora inicializada aleatoriamente, portanto, não podemos tirar conclusões relevantes a partir dos seus resultados. No entanto, o viés observado nos outros modelos provavelmente se deve ao conjunto de dados de pré-treinamento. O Pythia e o Mamba foram treinados no mesmo *dataset* (*The Pile*), enquanto o LLaMa3.1 foi treinado com *datasets* mais variados e modificados.

Durante o ajuste fino, conforme mostrado na Figura 2, o Pythia, Mamba e LLaMa atingiram seus picos de qualidade no conjunto de validação dentro do limite de épocas pré-estabelecido. O LLaMa atingiu seu melhor desempenho com menos épocas, enquanto o Pythia precisou de mais. Já a curva do BERT sugere que mais épocas de treinamento poderiam ter melhorado seu desempenho, visto que não apresentou o comportamento de pico observado nos outros modelos.

Quanto à qualidade dos resultados no conjunto de teste, a Tabela 3 mostra que o LLaMa foi o modelo que melhor se ajustou à tarefa. Entre os modelos de tamanho semelhante, o BERT apresentou o melhor desempenho, enquanto o Pythia foi o menos adequado, embora seus resultados ainda sejam competitivos.

As Matrizes 1b revelam que o maior erro comum entre os modelos foi a dificuldade em distinguir entre notícias neutras e positivas. Por outro lado, as notícias negativas foram classificadas com alta precisão.

Em termos de eficiência, analisando as Figuras 3 e 4, o LLaMa apresentou o maior tempo de treino e inferência, com uma disparidade significativa em comparação com os modelos menores. Entre os modelos de tamanho similar, o Pythia foi o mais eficiente no treinamento, enquanto o Mamba foi o menos eficiente. No tempo de inferência, o BERT foi o mais rápido, e o Mamba continuou sendo o mais lento.

Vale destacar os resultados do Mamba, que teoricamente é uma nova arquite-

tura projetada para ter a mesma qualidade dos Transformers, mas com maior eficiência. No entanto, em nossos testes, embora o Mamba tenha alcançado uma qualidade comparável, ele se mostrou inferior em termos de eficiência em todos os cenários. Alguns fatores que podem ter contribuído para isso são: 1) Estamos trabalhando com sequências relativamente curtas (nosso maior contexto tem pouco mais de cem palavras, e estamos gerando apenas uma), enquanto a motivação por trás do Mamba é lidar com sequências muito longas, como sequências de DNA ou documentos inteiros, onde o comprimento das sequências impacta mais a eficiência. 2) A implementação inicial do Mamba disponível no HuggingFace pode não ser a mais otimizada em comparação com as implementações eficientes de mecanismos de atenção (até o momento dos nossos testes), especialmente no manejo da memória durante a computação.

7 Conclusão

Neste relatório, exploramos teoricamente e na prática diferentes modelos de linguagem, analisando suas vantagens e desvantagens para a tarefa de análise de sentimentos em manchetes financeiras.

Os experimentos demonstraram que os modelos base, sem ajuste fino, não são adequados para a tarefa. No entanto, após o ajuste fino, todos os modelos apresentaram bons resultados dentro dos hiperparâmetros predefinidos.

As matrizes de confusão sugerem que o viés classificatório dos modelos gerais é fruto dos *datasets* de pré-treinamento, e os resultados finais mostram que há espaço para melhorias no conjunto de dados de ajuste, especialmente em relação às notícias positivas e neutras, uma vez que a confusão entre essas classes foi o erro mais frequente.

Nossas avaliações de qualidade e eficiência proporcionaram *insights* sobre a escolha de modelos em uma aplicação prática. Por exemplo, se a prioridade for qualidade, o LLaMa3.1 é a melhor opção. Se a prioridade for um treinamento mais factível, o Pythia demonstrou maior velocidade. Já em termos de custo de inferência, o BERT foi o mais eficiente.

Embora todos os modelos tenham obtido sucesso em nossos experimentos, alguns pontos merecem destaque. **(1)** A reafirmação do LLaMa3.1 como um modelo de referência em termos de qualidade entre os *open-source*, superando até os resultados obtidos em nosso projeto anterior com o LLaMa2, apesar de que sua grande quantidade de parâmetros afetou negativamente sua eficiência em comparação com modelos menores. **(2)** A relevância contínua do BERT, que, apesar de ser o modelo mais antigo e de menor tamanho, apresentou melhor qualidade em relação aos modelos mais novos de tamanho similar, demonstrando que a arquitetura *encoder-only* pode ser ainda melhor que a *decoder-only* em tarefas de classificação. **(3)** O potencial da nova arquitetura Mamba, que, embora tenha alcançado qualidade competitiva em comparação aos modelos baseados em atenção, não se destacou em termos de eficiência para nossa tarefa específica.

Em suma, nosso trabalho reforça a importância da adaptação de modelos de linguagem aos domínios específicos e mostra a diferença de aplicabilidade ao variar as especificações dos modelos, abrindo portas para mais investigações sobre ajustes eficientes

e aplicações práticas em diferentes contextos e abordagens.

A Apêndices

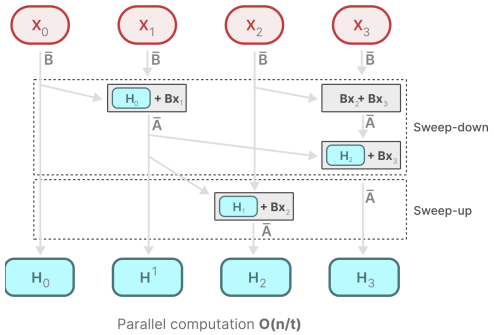


Figura 5: Ilustração do algoritmo de *scan* paralelo (n = tamanho da sequência; t = número de *threads*) (Maarten Grootendorst, 2024).

Referências

Cheonsu Jeong, 2024. - *Fine-tuning and Utilization Methods of Domain-specific LLMs*

Lucas Brennan-Almaraz & Daniel Guo & Sarvesh Babu - *Wrestling Mamba: Exploring Early Fine-Tuning Dynamics on Mamba and Transformer Architectures*

Zhichao Xu, 2024. - *RankMamba: Benchmarking Mamba's Document Ranking Performance in the Era of Transformers*

Dogu Araci, 2019. - *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*

Pau Rodriguez Inserte et al., 2024. - *Large Language Model Adaptation for Financial Sentiment Analysis*

Thanos Konstantinidis et al., 2024. - *FinLlama: Financial Sentiment Classification for Algorithmic Trading Applications*

Nicolas T. de Camargo, 2024. - *Ajuste fino de um modelo de linguagem para análise de sentimentos financeiros.*

Ashish Vaswani et al., 2017. - *Attention Is All You Need*

Albert Gu et al., 2021 - *Efficiently Modeling Long Sequences with Structured State Spaces*

Albert Gu; Tri Dao, 2023 - *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*

Edward J. Hu et al., 2021. - *LoRA: Low-Rank Adaptation of Large Language Models*

Jacob Devlin et al., 2018. - *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*

Stella Biderman et al., 2023. - *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*

Abhimanyu Dubey et al., 2017. - *The Llama 3 Herd of Models*

Maarten Grootendorst, 2024 - *A Visual Guide to Mamba and State Space Models*