**Análise Comparativa de Desempenho entre Support Vector Machine e GPT-4o mini na Classificação de Tópicos de Notícias**

**Autores:** Lucas Rego Da Silva

**Afiliação:** *Centro Universitário Euro-americano*

**E-mail:** *Lucas60899@unieuro.edu.com*

Resumo: A classificação automática de textos é uma tarefa fundamental no Processamento de Linguagem Natural (PLN), essencial para a organização do crescente volume de dados digitais. Com a ascensão dos Modelos de Linguagem Grandes (LLMs), surge a questão de seu desempenho em comparação com algoritmos de aprendizado de máquina tradicionais, consolidados e de menor custo computacional. Este trabalho apresenta uma análise comparativa quantitativa entre um modelo Support Vector Machine (SVM) (Cortes & Vapnik, 1995), utilizando features TF-IDF (Salton, Wong & Yang, 1975), e a API do LLM GPT-4o mini, operando em modo zero-shot, para a tarefa de classificação de notícias do dataset AG News (Zhang et al., 2015). A metodologia envolveu o treinamento do SVM em 120.000 amostras e a avaliação de ambos os modelos em uma amostra estratificada de 1.000 notícias do conjunto de teste. Os resultados, medidos por Acurácia e F1-Score (Macro), foram contra-intuitivos: o modelo SVM (Acurácia de 92%, F1 de 0.92) superou significativamente o GPT-4o mini (Acurácia de 85%, F1 de 0.86). A análise sugere que a capacidade do SVM de capturar sinais estatísticos fortes em um grande volume de dados de treino foi mais eficaz do que a compreensão semântica generalista do LLM em modo zero-shot. Os resultados foram submetidos a uma análise crítica detalhada, considerando custo computacional, generalização e significância prática. Conclui-se que algoritmos clássicos, quando bem aplicados, permanecem uma solução de alta performance e excelente custo-benefício para tarefas de classificação bem definidas.

**Palavras-chave:** Classificação de Texto, Aprendizado de Máquina, Support Vector Machine, Large Language Models, GPT, Análise Comparativa.

**1. Introdução**

Na atual era da informação, a produção de dados textuais cresce de forma exponencial. Notícias, artigos, e-mails e documentos jurídicos são gerados em um volume que torna a organização manual impraticável. Neste contexto, a Classificação Automática de Textos, uma tarefa central do PLN, emerge como uma solução crucial, permitindo a categorização automática e eficiente desses documentos.

Historicamente, a resolução desta tarefa tem sido dominada por algoritmos de Aprendizado de Máquina (AM) tradicionais. Abordagens como o Support Vector Machine (SVM) (Cortes & Vapnik, 1995) , combinadas com técnicas de engenharia de características como o TF-IDF (Salton, Wong & Yang, 1975) (Term Frequency-Inverse Document Frequency), consolidaram-se como o estado da arte por muitos anos.

Recentemente, o campo do PLN foi revolucionado pela ascensão dos LLMs, baseados na arquitetura Transformer (Vaswani et al., 2017), introduzida por Vaswani et al.. Modelos como a família GPT (Generative Pre-trained Transformer) da OpenAI demonstram uma capacidade de compreensão semântica sem precedentes, permitindo a classificação com pouca ou nenhuma customização (zero-shot).

Esta disrupção tecnológica levanta uma questão de pesquisa fundamental e de grande relevância prática: em uma tarefa de classificação bem definida, um LLM de ponta, operando sem treinamento específico, oferece um desempenho superior a um modelo clássico, otimizado e treinado em um grande volume de dados? A potencial performance de um LLM justifica seu custo computacional e financeiro inerente?

Este trabalho busca responder a essa questão através de uma análise comparativa rigorosa. O objetivo geral é comparar quantitativamente o desempenho, custo e complexidade entre um classificador SVM e a API do LLM GPT-4o mini na tarefa de classificação de tópicos do dataset AG News (Zhang et al., 2015). Para tal, os objetivos específicos são: (i) implementar e treinar um pipeline otimizado com TF-IDF (Salton, Wong & Yang, 1975) e o classificador LinearSVC; (ii) desenvolver e executar um processo de classificação zero-shot via API para o GPT-4o mini de forma paralela e eficiente; avaliar ambos os modelos utilizando as métricas de Acurácia e F1-Score (Macro); e (iv) analisar e discutir os resultados comparativos.

Este artigo está estruturado da seguinte forma: a Seção 2 apresenta a fundamentação teórica e trabalhos correlatos. A Seção 3 detalha a metodologia experimental. A Seção 4 apresenta os resultados obtidos, que são analisados na Seção 5. Por fim, a Seção 6 apresenta a conclusão, as limitações do estudo e sugestões para trabalhos futuros.

**2. Fundamentação Teórica e Trabalhos Relacionados**

**2.1. Classificação de Texto com Aprendizado de Máquina Tradicional**

A abordagem tradicional para classificação de texto envolve um pipeline de múltiplas etapas, que transforma o texto não estruturado em um formato que os algoritmos possam processar.

A primeira etapa crucial é a engenharia de características, onde o texto é convertido em vetores numéricos. Uma das técnicas mais eficazes e difundidas para tal é a TF-IDF (Salton, Wong & Yang, 1975). O TF-IDF (Salton, Wong & Yang, 1975) calcula um peso para cada palavra em um documento, que é proporcional à sua frequência no documento e inversamente proporcional à sua frequência em todo o conjunto de documentos (corpus), atribuindo assim maior importância a termos que são frequentes em um documento, mas raros no geral.

Uma vez que os textos são representados como vetores, um algoritmo classificador é treinado. O Support Vector Machine (SVM) (Cortes & Vapnik, 1995) é um algoritmo de aprendizado supervisionado particularmente poderoso para essa tarefa. O objetivo do SVM é encontrar um hiperplano em um espaço de alta dimensionalidade que separe de forma ótima as amostras de diferentes classes, maximizando a margem entre elas. Sua eficácia em lidar com dados de alta dimensionalidade, como os gerados pelo TF-IDF (Salton, Wong & Yang, 1975), e sua robustez contra o sobre ajuste (overfitting) o tornaram uma linha de base (baseline) forte em muitos problemas de classificação de texto.

**2.2. Classificação de Texto com Large Language Models (LLMs)**

Os LLMs representam uma mudança de paradigma. Baseados na arquitetura Transformer (Vaswani et al., 2017), introduzida por Vaswani et al., esses modelos abandonaram as estruturas recorrentes em favor de um mecanismo de autoatenção (self-attention). Isso permite que o modelo pese a importância de diferentes palavras em uma sequência de entrada, capturando relações complexas e de longa distância , e possibilitando um grau de paralelização no treinamento que era inviável anteriormente.

A principal força dos LLMs modernos, como o GPT-4o mini, reside em seu pré-treinamento em vastos volumes de texto da internet. Isso lhes confere um profundo conhecimento semântico e de mundo, que pode ser aplicado a novas tarefas com mínima customização. Uma dessas aplicações é o aprendizado zero-shot (Brown et al., 2020). Nessa abordagem, o modelo recebe uma instrução em linguagem natural (um prompt) descrevendo a tarefa e os dados de entrada, e realiza a classificação sem ter sido treinado em nenhum exemplo específico daquele problema.

**3. Metodologia**

**3.1. Conjunto de Dados**

O estudo utilizou o dataset público **AG News**, como construído por Zhang et al.. Este conjunto de dados consiste em títulos e descrições de notícias em inglês, classificadas em quatro categorias: **World (Mundo)**, **Sports (Esportes)**, **Business (Negócios)** e **Sci/Tech (Ciência/Tecnologia)**. O dataset é dividido em um conjunto de treinamento com 120.000 amostras e um conjunto de teste com 7.600 amostras, balanceado entre as quatro classes.

**3.2. Amostragem de Teste**

Para garantir uma comparação financeiramente viável com a API do GPT-4o mini, foi criada uma amostra estratificada de **1.000 documentos** do conjunto de teste original. A amostragem garantiu a mesma proporção de classes do conjunto original, resultando em 250 notícias de cada uma das quatro categorias. Esta mesma amostra foi utilizada para a avaliação final de ambos os modelos, garantindo uma comparação direta e justa.

**3.3. Implementação do Modelo SVM**

O modelo SVM foi implementado utilizando a biblioteca **Scikit-learn** em Python. Para otimizar o processo de vetorização e classificação, foi construído um Pipeline contendo duas etapas:

1. **Vetorização:** Foi utilizado o TfidfVectorizer com os parâmetros stop\_words='english' para remover palavras comuns da língua inglesa e ngram\_range=(1, 2) para considerar tanto palavras individuais quanto pares de palavras consecutivas como features.
2. **Classificação:** Foi empregado o classificador LinearSVC. Esta implementação é otimizada para problemas lineares com grande número de amostras e features, oferecendo uma aceleração significativa em relação à implementação padrão SVC(kernel='linear').

O pipeline completo foi treinado (.fit()) no conjunto de treino integral de 120.000 notícias. Após o treinamento, o modelo foi salvo em um arquivo usando a biblioteca joblib para persistência e reutilização.

**3.4. Implementação do Modelo GPT-4o mini**

A avaliação do GPT-4o mini foi realizada em modo *zero-shot* através da API oficial da OpenAI. O processo foi implementado em Python utilizando as seguintes técnicas:

* **Cliente Assíncrono:** Foi utilizado o AsyncOpenAI para permitir chamadas concorrentes à API.
* **Execução Paralela:** A biblioteca asyncio foi usada para orquestrar o envio de múltiplas requisições em paralelo, visando a eficiência de tempo. Um asyncio.Semaphore foi configurado para limitar a concorrência e evitar erros de *rate limit*.
* **Engenharia de Prompt:** Para cada notícia, foi gerado um prompt claro e estruturado, instruindo o modelo a retornar a classificação em um formato JSON para facilitar o processamento automático da resposta. O prompt utilizado foi:

Sua tarefa é classificar a notícia abaixo em uma das quatro categorias a seguir:

- 0: World (Notícias sobre eventos globais e política internacional)

- 1: Sports (Notícias sobre todos os tipos de esportes)

- 2: Business (Notícias sobre finanças, empresas e economia)

- 3: Sci/Tech (Notícias sobre ciência e tecnologia)

Analise a notícia e retorne APENAS um objeto JSON com a chave "categoria\_id".

Notícia:

"{texto\_da\_noticia}"

**3.5. Métricas de Avaliação**

O desempenho dos dois modelos foi avaliado e comparado com base nas seguintes métricas, calculadas a partir da matriz de confusão:

* **Acurácia:** A proporção de predições corretas sobre o total de predições.
* **Precisão, Recall e F1-Score:** O F1-Score é a média harmônica da precisão e do recall. Para a comparação geral, foi utilizado o **F1-Score (Macro Average)**, que calcula a métrica para cada classe independentemente e depois tira a média, tratando todas as classes com igual importância.

**4. Resultados**

Os experimentos foram executados conforme a metodologia descrita. A seguir, são apresentados os resultados obtidos para cada modelo na amostra de teste de 1.000 notícias.

**4.1. Desempenho do Modelo SVM**

O modelo SVM, após treinamento no conjunto de 120.000 amostras, alcançou os resultados apresentados na Tabela 1 quando avaliado no conjunto de teste.

**Tabela 1: Relatório de Classificação do Modelo SVM (LinearSVC)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **f1-score** | **support** |
| **World** | 0.95 | 0.90 | 0.92 | 250 |
| **Sports** | 0.97 | 0.98 | 0.97 | 250 |
| **Business** | 0.86 | 0.92 | 0.89 | 250 |
| **Sci/Tech** | 0.91 | 0.90 | 0.91 | 250 |
| **accuracy** |  |  | 0.92 | 1000 |
| **macro avg** | 0.92 | 0.92 | 0.92 | 1000 |
| **weighted avg** | 0.92 | 0.92 | 0.92 | 1000 |

**4.2. Desempenho do Modelo GPT-4o mini**

O modelo GPT-4o mini (OpenAI, 2024), operando em modo zero-shot, produziu os resultados apresentados na Tabela 2 para o mesmo conjunto de teste.

**Tabela 2: Relatório de Classificação do Modelo GPT-4o mini**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **f1-score** | **support** |
| **World** | 0.90 | 0.86 | 0.88 | 250 |
| **Sports** | 0.97 | 0.97 | 0.97 | 250 |
| **Business** | 0.71 | 0.89 | 0.79 | 250 |
| **Sci/Tech** | 0.89 | 0.70 | 0.78 | 250 |
| **accuracy** |  |  | 0.85 | 1000 |
| **macro avg** | 0.87 | 0.85 | 0.86 | 1000 |
| **weighted avg** | 0.87 | 0.85 | 0.86 | 1000 |

**4.3. Sumário Comparativo**

A Tabela 3 apresenta uma comparação direta das métricas de desempenho globais entre os dois modelos.

**Tabela 3: Comparativo de Desempenho Global**

|  |  |  |
| --- | --- | --- |
| **Modelo** | **Acurácia** | **F1-Score (Macro Avg)** |
| **SVM (LinearSVC)** | **92.0%** | **0.92** |
| GPT-4o mini | 85.0% | 0.86 |

5. Discussão e Análise Crítica

Para validar a significância estatística entre os modelos, foi aplicado o teste de McNemar sobre as classificações corretas e incorretas dos dois sistemas. Os resultados indicaram uma diferença estatisticamente significativa (p < 0.01) em favor do SVM. Isso reforça que a superioridade observada não é fruto do acaso, mas sim de um desempenho consistentemente superior em múltiplas classes.

A análise comparativa dos resultados revela um cenário contra-intuitivo e rico em implicações. O modelo SVM treinado não apenas se mostrou competitivo, como superou o LLM de ponta GPT-4o mini em performance global por uma margem significativa.

A análise por categoria aprofunda o entendimento. O empate técnico na categoria "Sports" (F1 de 0.97 para ambos) sugere que seu vocabulário altamente distintivo é um sinal claro tanto para a análise estatística do TF-IDF (Salton, Wong & Yang, 1975) quanto para a compreensão semântica do GPT. Contudo, nas categorias "Business" e "Sci/Tech", onde o vocabulário pode ter maior sobreposição, o SVM demonstrou uma superioridade expressiva. O F1-score do SVM foi 10 pontos superior em "Business" e 13 pontos em "Sci/Tech". Isso indica que, para este dataset, a capacidade do TF-IDF (Salton, Wong & Yang, 1975) de identificar e ponderar palavras-chave e n-gramas com forte poder discriminativo, aprendida a partir de 120.000 exemplos, foi uma estratégia mais eficaz do que a habilidade semântica generalista do LLM.

Este resultado levanta uma discussão fundamental sobre o paradigma do "Especialista vs. Generalista". O SVM foi treinado extensivamente para se tornar um especialista na tarefa, enquanto o GPT-4o mini atuou como um generalista brilhante, mas sem o contexto focado que o treinamento supervisionado proporciona. Fica evidente que, quando dados rotulados são abundantes, treinar um especialista pode produzir resultados superiores aos de um generalista em modo *zero-shot*.

Adicionalmente, as implicações práticas são notáveis. O treinamento do SVM, após otimização para LinearSVC, foi executado em poucos minutos em um computador pessoal, com custo zero de inferência. A avaliação com o GPT-4o mini, embora de baixo custo monetário (~R$ 0,40), exigiu gerenciamento de API e tempo de execução condicionado à rede. Para aplicações em larga escala, o modelo treinado localmente apresenta vantagens evidentes em custo e velocidade de inferência.

Em termos de custo computacional, o SVM foi treinado em aproximadamente 11 segundos utilizando um notebook com 16GB de RAM e processador ryzen 7 5700g. A inferência do GPT-4o mini via API levou em média 0,09 segundos por notícia, totalizando cerca de 1 minutos e 30 segundos para 1.000 amostras, com custo de aproximadamente R$ 0,40. Esses dados reforçam que, além da precisão superior, o SVM apresenta vantagens operacionais relevantes em contextos produtivos.

6. Conclusão e Trabalhos Futuros

Este trabalho comparou duas abordagens distintas para a classificação de texto: um modelo SVM clássico treinado em um grande corpus e um LLM de ponta em modo zero-shot. Conclui-se que, para o dataset AG News (Zhang et al., 2015), a abordagem supervisionada com SVM e features TF-IDF (Salton, Wong & Yang, 1975) alcançou um desempenho superior (Acurácia de 92%) em comparação com o GPT-4o mini (Acurácia de 85%). O estudo evidencia que, em cenários com tarefas bem definidas e disponibilidade de dados de treino, os algoritmos clássicos de aprendizado de máquina continuam sendo ferramentas extremamente relevantes, eficazes e de excelente custo-benefício.

Limitações do Estudo: Os resultados aqui apresentados são específicos para o dataset AG News (Zhang et al., 2015) e para a abordagem zero-shot do LLM. A performance pode variar em datasets com características diferentes ou com a aplicação de técnicas de engenharia de prompt mais avançadas.

**Trabalhos Futuros:** Como uma extensão natural desta pesquisa, sugere-se a avaliação do GPT-4o mini em modo **few-shot**, fornecendo alguns exemplos de cada categoria no prompt. Seria de grande valia investigar se essa técnica de aprendizado em contexto é capaz de diminuir ou eliminar a diferença de performance observada em relação ao modelo SVM especialista. Adicionalmente, a comparação poderia ser estendida para outros LLMs e outros datasets de benchmark.

**7. Referências**

[11] BROWN, T. et al. Language Models are Few-Shot Learners. In: Advances in Neural Information Processing Systems. 2020.

[12] RAFFEL, C. et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research, 2020.

[13] GAO, T. et al. Making Pretrained Language Models Better Few-shot Learners. In: Proceedings of ACL. 2021.

[14] WANG, A. et al. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In: ICLR 2019.

[1] ZHANG, X.; ZHAO, J.; LECUN, Y. Character-level Convolutional Networks for Text Classification. In: *Advances in Neural Information Processing Systems 28*. 2015.

[2] CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, v. 20, n. 3, p. 273-297, 1995.

[3] VASWANI, A. et al. Attention is all you need. In: *Advances in Neural Information Processing Systems 30*. 2017.

[4] DEVLIN, J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 2019.

[5] JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing (3rd ed. draft)*. 2023. Disponível em: <https://web.stanford.edu/~jurafsky/slp3/>.

[6] SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. *Communications of the ACM*, v. 18, n. 11, p. 613-620, 1975.

[7] SOUSA, R. N. de. MINERJUS: solução de apoio à classificação processual com uso de Inteligência Artificial. Tese (Doutorado) — Universidade Federal do Tocantins, 2019.

[8] MARINATO, M. S. et al. Classificação Automática de Petições Iniciais Usando Classificadores Combinados. In: *Anais do XVI Brazilian e-Science Workshop*. SBC, 2022.

[9] OpenAI. *GPT-4o*. 2024. Disponível em: <https://openai.com/index/hello-gpt-4o/>.

[10] SILVA JÚNIOR, P. C. S. da. *Classificação de petições iniciais no Conselho Nacional do Ministério Público*. Dissertação (Mestrado Profissional em Computação Aplicada) — Universidade de Brasília, Brasília, 2024.