UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: Luis Sávio Gomes Rosa

1. Proposta do Estudo

O presente estudo de caso propõe uma comparação entre os métodos de extração de features de NLP com modelos LLM (Large Language Models). Neste trabalho vamos aplicar dois modelos de LLM (Um modelo pré treinado da biblioteca Hugging Faces e outro modelo feito do zero), comparar os resultados, e posteriormente comparar seus resultados com métodos de extração de features.

2. Introdução

O dataset utilizado neste trabalho são sobre notícias que foram classificadas como fake news e não fake news. Inicialmente, foram dados 2 datasets, ambos com 300 linhas cada, um com apenas fake news, e outro sem fake news. Os datasets possuíam a mesma estrutura, caracterizadas pelas seguintes colunas:

- ***id**: Um identificador único para cada registro.*
- ***news_url**: O URL da notícia.*
- ***title**: O título da notícia.*
- ***tweet_ids**: Identificadores de tweets associados à notícia.*

- ***Observações:***

Inicialmente foram dois datasets, logo, como o objetivo se trata de uma classificação, foi-se necessário a atribuição de uma nova coluna "fake_news", coluna essa do tipo booleana, onde 1 representa que a notícia é fake news e 0 significa que não. A partir disso foi-se necessário a concatenação dos dois datasets.

O objetivo principal deste relatório é a aplicação de modelos baseados em transformers com a aplicação de um modelo pré treinado e outro e um modelo próprio desenvolvido do zero com uma arquitetura de Transformer Encoder. As etapas deste estudo incluem*:*

3. Estudo Anterior

É importante ressaltar o que foi feito no estudo anterior, a seguir, será apresentado o fluxograma com os processos do estudo anterior e seus resultados.

Fluxograma do primeiro processo:

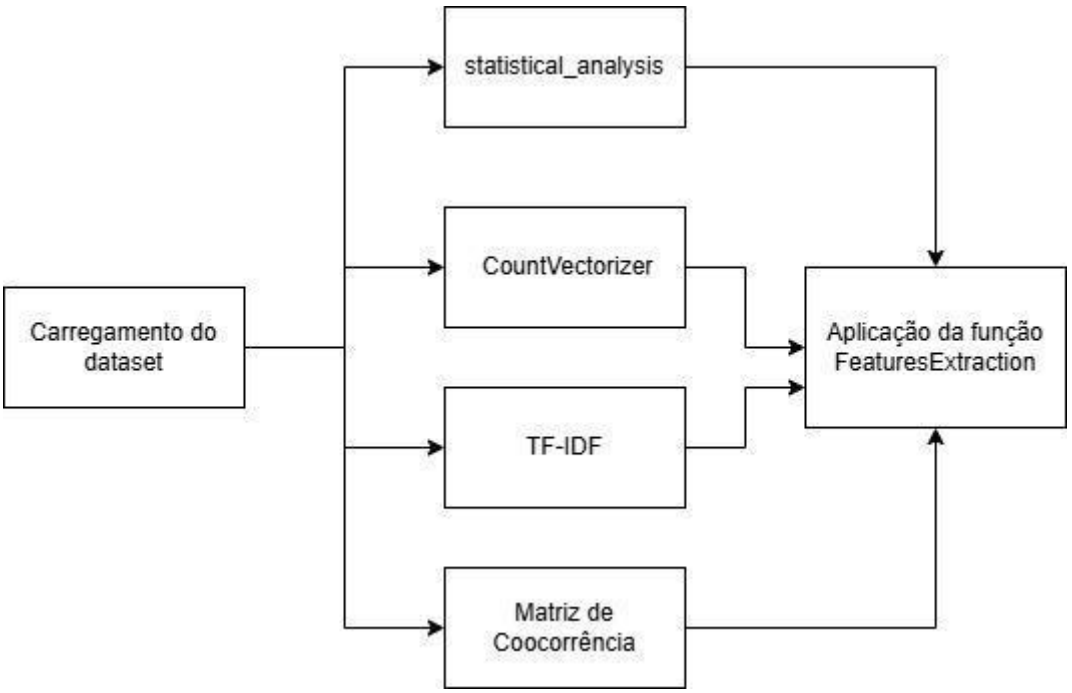


Tabela com os resultados das acurácias do primeiro estudo:

Modelo	**acurácia**
cooccurrence_matrix_no_pre	**0.950000**
cooccurrence_matrix_pre	**0.950000**
tfidf_vectorizer_no_pre	**0.933333**
count_vectorizer_no_pre	**0.927778**
count_vectorizer_pre	**0.922222**
tfidf_vectorizer_pre	**0.922222**

statistical_analysis_no_pre	**0.561111**
statistical_analysis_pre	**0.538889**
word2vec_no_pre	**0.455556**
word2vec_pre	**0.455556**

Uma breve discussão dos resultados:

A análise realizada permitiu avaliar o desempenho de diferentes métodos de extração de atributos e técnicas de pré-processamento em um modelo de classificação para detecção de fake news. Os resultados demonstraram que a ***Matriz de Coocorrência****, tanto com quanto sem pré-processamento, foi a técnica mais eficaz, alcançando uma acurácia de* ***95%****. Isso evidencia sua capacidade de capturar relações semânticas entre palavras, especialmente após a redução de dimensionalidade com PCA. Os métodos* ***TF-IDF*** *e* ***CountVectorizer*** *também apresentaram desempenhos próximos, destacando-se como alternativas robustas para análise textual, embora tenham registrado ligeira queda de desempenho com a aplicação do pré-processamento. Por outro lado, a* ***Análise Estatística*** *e o* ***Word2Vec*** *não foram capazes de fornecer informações suficientemente discriminantes para obter bons resultados, mostrando limitações significativas no contexto deste dataset.* Isto posto, seguimos com o estudo atual de LLM´s

4. Etapas do estudo:

Transformação dos dados em encoder: *Transformação do dataset em encoders*

Pré-Processamento Textual: *Preparação do dataset para que os modelos para formato adequado que serão utilizados nos modelos nos modelos posteriormente.*

Aplicação de um modelo pré-treinado*: Aplicação do modelo pré- treinado do HuggingFace*

Aplicação do Fine-Tuning: Aplicação do fine-tuning no modelo do *HuggingFace em busca dos melhores parâmetros.*

Treinamento de um modelo do zero: *Treinamento de um Transformer do zero.*

Avaliação de Modelos: *Comparar o desempenho dos LLM´s.*

Comparação de resultados final: *Comparação Final dos modelos com os resultados dos métodos anteriores.*

Assim, este trabalho busca contribuir para a compreensão e desenvolvimento de soluções eficazes no campo da detecção de fake news.

5. Detalhamento das etapas e discussões do projeto

a. Pré-processamento e encoder

Como o objetivo do nosso estudo é comparar resultados, o dataset que foi utilizado foi o mesmo que foi passado pelo pré-processamento do estudo anterior. Logo, não foi necessário nenhum pré-processamento. Isto posto, o dataset passou pelo processo de encoder, onde ocorreu a codificação das palavras em vetores numéricos. No arquivo de pré processamento foi feito:

- Implementado o pré-processamento para adequar os textos ao formato exigido pelos modelos.
- Divisão do dataset em treino e teste, garantindo a mesma divisão para todos os modelos.
- Conversão das labels para formato numérico.

b. Aplicação do Hugging Face sem fine-tuning:

Utilizando os modelos pré-treinados que estão na huggingFace, sem realizar nenhum fine tuning, foi feito uma predição no conjunto de teste, e Gerado o classification report e a matriz confusão.

- Escolha do modelo base: cardiffnlp/twitter-xlm-roberta-base.
- Tokenização: Utilização do tokenizer do modelo base.
- Criação do Dataset: Transformação do CSV para um dataset PyTorch.
- Avaliação com conjunto de teste.
- Salvamento do modelo treinado (fake_news_model).

c. Aplicação do Fine-Tuning

Nesta etapa foi gerado um modelo via fine tuning, e escolhido o modelo apropriado como base e especifique para a aplicação. As etapas foram:

- O modelo da Hugging Face foi ajustado ao conjunto de dados através de fine-tuning.
- Esse processo permitiu a adaptação do modelo às particularidades do dataset utilizado.
- O modelo treinado foi salvo para uso na etapa de predição.

d. Treinamento de um Modelo Próprio

Nesta etapa foi criada a arquitetura de encoder para classificar as informações textuais. O modelo foi treinado com os dados gerados no arquivo 2. Os parâmetros do modelo foram:

- `MAX_LEN = 128`
- `EMBED_DIM = 128`
- `VOCAB_SIZE = 4000`
- `NUM_HEADS = 4`
- `NUM_LAYERS = 3`
- `HIDDEN_DIM = 256`
- `DROPOUT = 0.4`
- `BATCH_SIZE = 16`
- `EPOCHS = 80`
- `LEARNING_RATE = 3e-4`

e. Avaliação e Comparação dos Modelos LLM

Nessa etapa foram realizadas as predições dos modelos treinados. Foi feito uma clasification report e matriz confusão para cada modelo.

6. Resultados dos LLM´s

Neste tópico discutiremos sobre os resultados dos modelos baseados em transformers.

a. Avaliação do Modelo HugginFace sem Fine-Turned:

	Precision	Recall	f1-score
0	0.51	0.92	0.65
1	0.61	0.12	0.20
accuracy			0.52

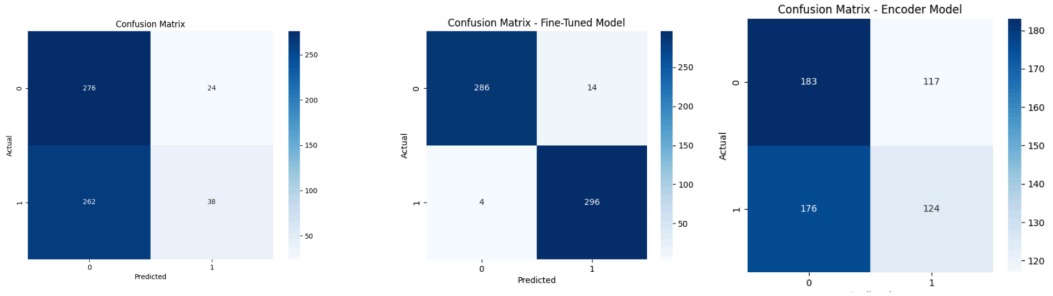
b. Avaliação do Modelo HugginFace Fine-Turned:

	Precision	Recall	f1-score
0	0.98	0.95	0.96
1	0.95	0.98	0.97
accuracy			0.97

c. Avaliação do modelo próprio:

	Precision	Recall	f1-score
0	0.50	0.61	0.55
1	0.51	0.41	0.45
accuracy			0.51

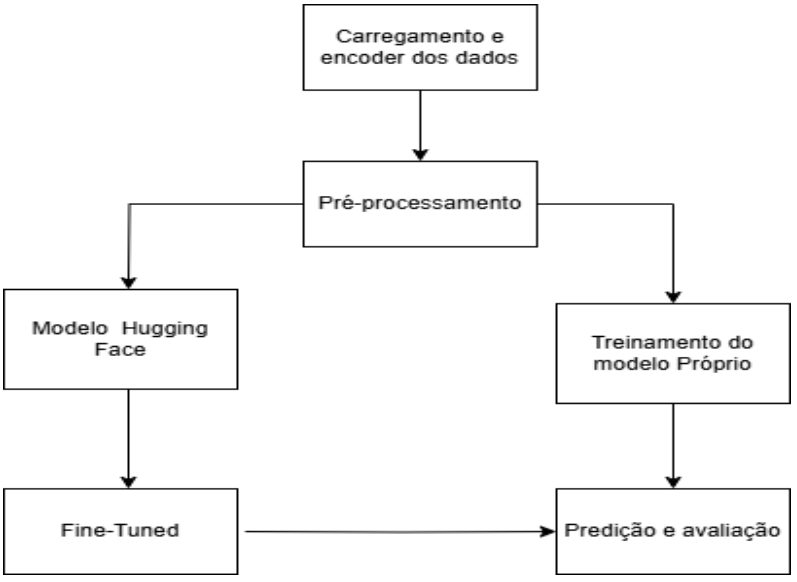
d. Matriz de confusão dos modelos:



e. Discussão do modelos llm

O estudo demonstrou a importância da escolha de um bom modelo base para a classificação de fake news. O uso de modelos pré-treinados, como os da Hugging Face, mostrou-se vantajoso, reduzindo a necessidade de um treinamento longo e alcançando alta precisão rapidamente. No entanto, o modelo próprio, apesar do desempenho inferior, oferece a possibilidade de personalização completa. Futuras melhorias poderiam incluir a expansão do dataset e a experimentação com arquiteturas mais profundas.

Fluxograma do estudo de LLM´s:



7. Comparação dos resultados dos modelos com o estudo anterior.

Modelo	acurácia
cooccurrence_matrix_no_pre	0.950000
cooccurrence_matrix_pre	0.950000
tfidf_vectorizer_no_pre	0.933333
count_vectorizer_no_pre	0.927778
count_vectorizer_pre	0.922222
tfidf_vectorizer_pre	0.922222
statistical_analysis_no_pre	0.561111
statistical_analysis_pre	0.538889
word2vec_no_pre	0.455556
word2vec_pre	0.455556

Modelos AV1

Modelos AV2

Modelo	Acurácia
Modelo Próprio	0.51
HagginFace Fine-Turned	0.97

Os resultados apresentados indicam uma clara diferença de desempenho entre as abordagens tradicionais de extração de atributos e os modelos baseados em aprendizado profundo. Os modelos da AV1, que utilizam técnicas como co-occurrence matrix, TF-IDF e CountVectorizer, apresentaram um desempenho sólido, com acurácias variando entre 91% e 95%. Em particular, os modelos baseados em co-ocorrência alcançaram 95% de acurácia, demonstrando que métodos tradicionais bem ajustados ainda são altamente eficazes para a classificação de fake news. No entanto, abordagens baseadas em análises estatísticas e Word2Vec apresentaram um desempenho inferior, com acurácias entre 45% e 56%, sugerindo que essas técnicas podem não capturar tão bem as nuances dos textos analisados.

Na AV2, que engloba modelos baseados em redes neurais, o modelo fine-tuned da Hugging Face se destacou, atingindo 97% de acurácia, o que o posiciona como a melhor abordagem testada. Isso reforça a importância do fine-tuning em modelos pré-treinados, que já possuem um entendimento contextual dos textos e podem ser ajustados com dados específicos da tarefa. Por outro lado, o modelo próprio desenvolvido do zero obteve 51% de acurácia, um desempenho abaixo do esperado e inferior até mesmo a algumas técnicas tradicionais de extração de atributos.

Com base nesses resultados, fica evidente que o fine-tuning de modelos pré-treinados é a abordagem mais eficiente para essa tarefa, superando as técnicas tradicionais e os modelos desenvolvidos do zero. O modelo próprio, apesar de apresentar um

desempenho inferior, pode ser aprimorado com ajustes na arquitetura, aumento da quantidade de dados e otimização dos hiperparâmetros. Além disso, técnicas como data augmentation podem ser exploradas para balancear melhor as classes e melhorar a generalização do modelo. Dessa forma, os resultados mostram que, embora métodos tradicionais sejam bastante eficazes, o uso de modelos pré-treinados com fine-tuning é a melhor estratégia para a classificação de fake news.

8. Próximos passos

1. Otimização do Modelo Próprio

- Ajuste da Arquitetura: Modificar a estrutura do modelo próprio, explorando diferentes números de camadas, cabeças de atenção e dimensões do embedding.
- Aprimoramento dos Hiperparâmetros: Testar outras configurações para taxa de aprendizado, batch size e dropout, utilizando técnicas como Grid Search ou Bayesian Optimization.
- Regularização: Implementar camadas de normalização e dropout para evitar overfitting.

2. Expansão e Aprimoramento do Dataset

- Aumento da Base de Dados: Incorporar mais amostras de textos, utilizando datasets externos ou técnicas de web scraping.
- Data Augmentation: Aplicar técnicas como sinonimização, tradução e substituição de palavras-chave para aumentar a diversidade dos textos de treino.
- Balanceamento das Classes: Se necessário, aplicar técnicas como oversampling (SMOTE) ou undersampling para corrigir possíveis desbalanceamentos.

3. Uso de Modelos Mais Avançados

- Testar Arquiteturas Mais Robustas: Explorar modelos mais modernos, como DeBERTa, BART ou T5, que podem trazer ganhos significativos de performance.
- Ajuste Fino em Modelos Maiores: Aplicar fine-tuning em modelos mais robustos da Hugging Face, como RoBERTa ou XLM-R, ajustando para o contexto de fake news.
- Ensemble de Modelos: Combinar diferentes modelos (exemplo: transformer + modelo estatístico) para capturar diferentes padrões no texto.