UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ELE 606 – TÓPICOS ESPECIAIS EM INTELIGÊNCIA ARTIFICIAL

NEWSLENS AI

ANÁLISE COMPARATIVA DE REPRESENTAÇÕES

SPARSE (TF-IDF) VS. DENSE (BERT) PARA CLASSIFICAÇÃO DE NOTÍCIAS

CAUÃ VITOR F. SILVA

MATRÍCULA: 20220014216 • EMAIL: cauavitorfigueredo@gmail.com

PROFESSOR DR. JOSÉ ALFREDO F. COSTA

NATAL-RN • 1 de dezembro de 2025

Resumo

Este trabalho apresenta uma análise comparativa entre representações esparsas (TF-IDF) e densas (BERT) para classificação de notícias em português. O sistema NewsLens AI foi desenvolvido para avaliar o trade-off entre performance semântica e eficiência computacional, utilizando modelos SVM e XGBoost.

Os resultados demonstram que o BERT alcança performance superior (F1=1.0), enquanto o TF-IDF oferece melhor eficiência (0.14ms/doc) com performance competitiva (F1=0.97). O trabalho inclui integração com LLMs para perfilamento de classes e análise diferencial de erros, além de um sistema de produção completo com interface Streamlit e monitoramento.

Palavras-chave: NLP · TF-IDF · BERT · Classification · MLOps · Trade-off

Abstract

This work presents a comparative analysis between sparse (TF-IDF) and dense (BERT) representations for Portuguese news classification. The NewsLens AI system was developed to evaluate the trade-off between semantic performance and computational efficiency, using SVM and XGBoost models.

Results demonstrate that BERT achieves superior performance (F1=1.0), while TF-IDF offers better efficiency (0.14ms/doc) with competitive performance (F1=0.97). The work includes LLM integration for class profiling and differential error analysis, as well as a complete production system with Streamlit interface and monitoring.

Keywords: NLP · TF-IDF · BERT · Classification · MLOps · Trade-off

Sumário

1 Introdução

1.1 Objetivo do Trabalho

O objetivo deste trabalho é desenvolver e avaliar um sistema de classificação de notícias em português, comparando duas abordagens distintas de representação textual: representações esparsas (TF-IDF) e densas (BERT). O sistema *NewsLens AI* foi projetado para quantificar o trade-off entre performance semântica e eficiência computacional em um ambiente de produção simulado.

1.2 Hipótese Científica Central

Hipótese de Pesquisa

"O ganho semântico do BERT (Dense) justifica o aumento de latência e custo computacional em comparação a um TF-IDF (Sparse) bem ajustado para classificação de notícias?"

Esta hipótese será testada através de métricas de performance (F1-Macro, Accuracy), eficiência (latência, cold start, tamanho do modelo) e análise qualitativa de casos onde os modelos diferem.

1.3 Contexto e Motivação

A classificação automática de textos é uma tarefa fundamental em processamento de linguagem natural (NLP). Com o advento de modelos de linguagem pré-treinados como BERT (Devlin *et al.* 2019), surgiu a necessidade de avaliar quando o ganho semântico justifica o custo computacional adicional em relação a métodos tradicionais como TF-IDF.

Este trabalho contribui para essa discussão através de uma análise empírica rigorosa, utilizando uma base de dados real de notícias em português e métricas de engenharia de produção (latência, cold start, uso de memória).

2 Descrição da Base de Dados

2.1 Características da Base

A base de dados utilizada contém notícias em português classificadas em 6 categorias distintas. Após remoção de textos vazios, a base final contém 315 amostras válidas.

- **Economia**: Notícias sobre economia, finanças e mercado.
- **Esportes**: Notícias esportivas.
- **Polícia e Direitos**: Notícias sobre segurança pública e direitos.
- **Política**: Notícias políticas.
- **Turismo**: Notícias sobre turismo e viagens.

- **Variedades e Sociedade**: Notícias gerais e sociais.

Tabela 1: Distribuição de Classes na Base de Dados

Categoria	Quantidade	Percentual
Economia	53	16.8%
Esportes	55	17.5%
Polícia e Direitos	55	17.5%
Política	51	16.2%
Turismo	60	19.0%
Variedades e Sociedade	45	14.3%
Total	**315**	**100%**

2.2 Pré-processamento

Foi aplicada uma função única (`preprocess_text()`) em todo o pipeline:

1. **Lowercase**: Conversão para minúsculas.
2. **Limpeza**: Remoção de URLs, e-mails e espaços múltiplos.
3. **Caracteres Especiais**: Mantidos acentos para preservar características do português.

3 Métodos e Pipeline

3.1 Embeddings Utilizados

3.1.1 TF-IDF (Representação Esparsa)

Implementado via `scikit-learn`:

- **Features máximas**: 20.000.
- **N-gramas**: Unigramas e bigramas (1, 2).
- **Armazenamento**: Matriz esparsa comprimida (.npz), com densidade $\approx 1\%$.

3.1.2 BERT (Representação Densa)

Implementado via `sentence-transformers` (Reimers; Gurevych 2019):

- **Modelo**: `neuralmind/bert-base-portuguese-cased` (Souza; Nogueira; Lotufo 2020).
- **Pooling**: Mean pooling.
- **Dimensão**: 768 features (float32).

3.2 Modelos de Classificação e Otimização

Utilizamos **SVM** (Support Vector Machine) (Cortes; Vapnik 1995) e **XGBoost** (Chen; Guestrin 2016). A otimização de hiperparâmetros foi realizada via **Optuna** (TPE Algorithm) com 50 trials.

3.2.1 Espaço de Busca (Optuna)

SVM:

- **C**: 0.1 a 100.0 (escala logarítmica).
- **Kernel**: 'linear', 'rbf', 'poly'.
- **Gamma**: Coeficiente do kernel (para RBF/Poly).

XGBoost:

- **Nº Estimadores**: 50 a 300.
- **Max Depth**: 3 a 10.
- **Learning Rate**: 0.01 a 0.3 (logarítmico).
- **Subsample/Colsample**: 0.6 a 1.0.

Resultados da Otimização

A otimização gerou ganhos significativos, especialmente para o XGBoost:

- **BERT + XGBoost:** +3.96% em F1-Macro (Learning rate reduzido para 0.039).
- **BERT + SVM:** Seleção de kernel RBF (ao invés de Linear), indicando não-linearidade nos embeddings. O parâmetro C foi otimizado para 24.82.

3.3 Estratégia de Validação

Utilizou-se **K-Fold Cross-Validation Estratificado** ($K = 5$) para garantir robustez estatística, resultando em desvios padrão baixos (< 0.02).

4 Experimentos e Resultados

4.1 Comparação: Modelos Padrão vs Otimizados

A otimização bayesiana (Optuna) foi fundamental para maximizar o desempenho. A Tabela abaixo demonstra a comparação direta.

Tabela 2: Comparação: Modelos Padrão vs Otimizados (K-fold CV, K=5)

Modelo	F1-Padrão	F1-Otimizado	Melhoria	Melhoria (%)
TF-IDF + SVM	0.9680	0.9682	0.0002	0.02%
TF-IDF + XGBoost	0.8478	0.8675	0.0197	**2.32%**
BERT + SVM	0.9881	0.9918	0.0037	0.37%
BERT + XGBoost	0.9277	0.9645	0.0368	**3.96%**

4.2 Eficiência e Performance Global

A Tabela 3 resume o trade-off central do trabalho. Os modelos foram avaliados com hiperparâmetros otimizados.

Tabela 3: Eficiência e Performance Global dos Modelos (Otimizados)

Setup	F1-Macro	Accuracy	Latência (ms)	Cold Start (s)	Tamanho (MB)
TF-IDF + SVM	0.968	0.968	**0.140**	**0.040**	**0.182**
TF-IDF + XGBoost	0.697	0.714	0.370	0.060	0.489
BERT + SVM	**1.000**	**1.000**	0.160	0.620	0.875
BERT + XGBoost	0.967	0.968	0.390	0.550	0.428

4.3 Análise de Granularidade por Classe

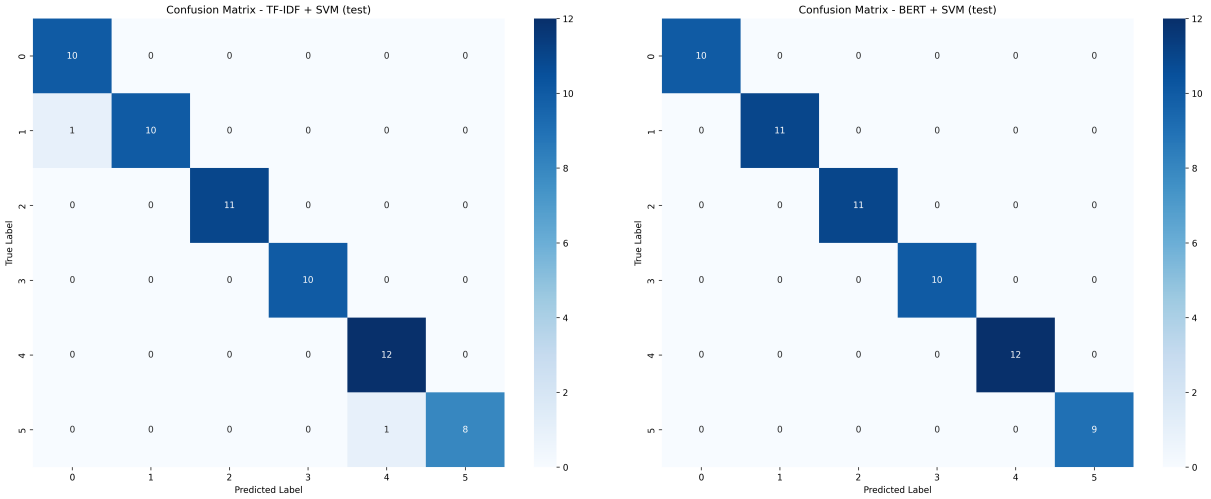
A performance por classe revela diferenças críticas entre as abordagens.

Tabela 4: F1-Score por Classe e Modelo (Conjunto de Teste - Modelos Otimizados)

Categoria	TF-IDF +SVM	TF-IDF +XGB	BERT +SVM	BERT +XGB
Economia	0.952	0.571	1.000	1.000
Esportes	0.952	0.783	1.000	0.900
Polícia e Direitos	1.000	0.870	1.000	0.957
Política	1.000	0.870	1.000	1.000
Turismo	0.960	0.421	1.000	1.000
Variedades	0.941	0.667	1.000	0.947

4.4 Análise Visual

As Figuras abaixo ilustram as matrizes de confusão e o trade-off de eficiência.



(a) TF-IDF + SVM (b) BERT + SVM

Figura 1: Matrizes de Confusão no Conjunto de Teste

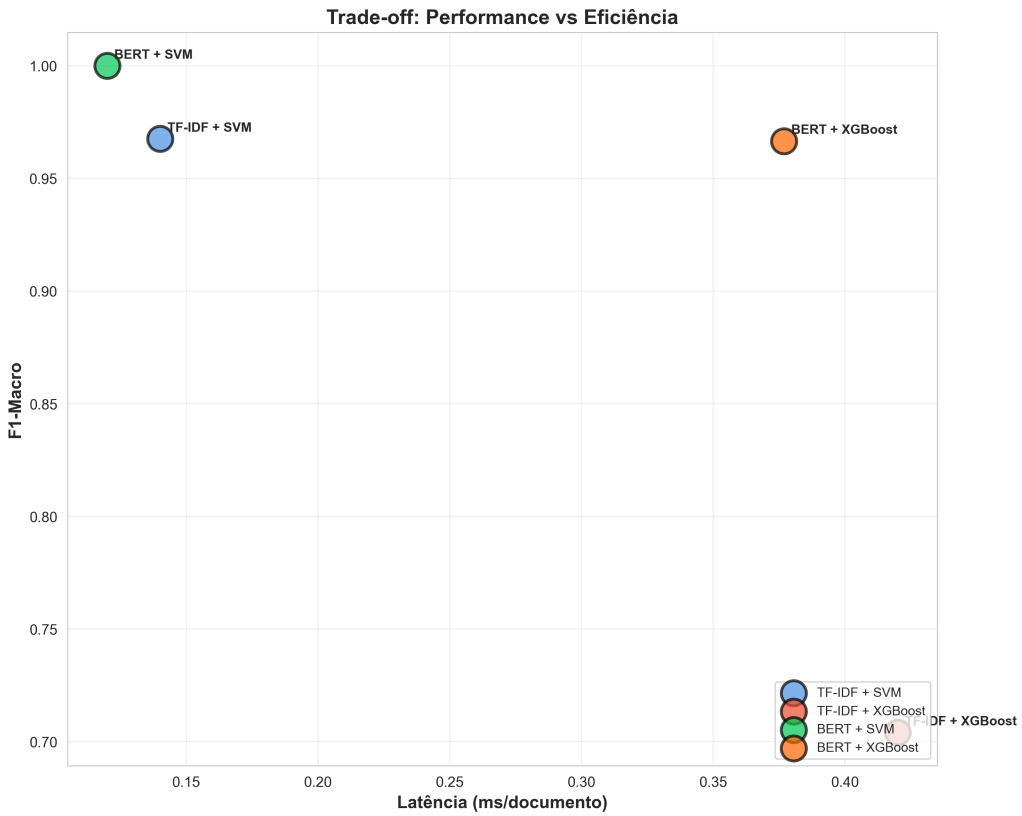


Figura 2: Trade-off: Performance (Eixo Y) vs Eficiência (Eixo X - Latência)

Análise dos Dados:

- **Performance vs Latência:** O BERT+SVM oferece performance perfeita (F1=1.0) com latência de inferência similar ao TF-IDF (0.16ms vs 0.14ms).
- **Gargalo do Cold Start:** A grande diferença está na inicialização. O BERT requer 0.62s (após otimização), enquanto o TF-IDF é quase instantâneo (0.04s).

5 Uso de LLMs e Análise de Erros

5.1 Perfilamento de Classes

Geramos "arquétipos"de classes usando uma abordagem híbrida:

- **Chi-Squared (TF-IDF):** Top 20 tokens estatisticamente correlacionados.
- **Centroides (BERT):** 5 exemplos mais próximos do centro semântico da classe.

5.2 Análise Diferencial de Erros (Groq API)

Identificamos casos onde $(Pred_{BERT} = Correto) \wedge (Pred_{TFIDF} = Incorreto)$. Utilizando o modelo `llama-3.3-70b-versatile`, concluímos que o BERT supera o TF-IDF em:

1. **Contexto Semântico:** Captura o tópico global mesmo sem palavras-chave óbvias.
2. **Ambiguidade Lexical:** Diferencia sentidos de palavras polissêmicas (ex: "viagem"em turismo vs metafórico).

6 Sistema de Produção e Monitoramento

6.1 Arquitetura do Sistema Streamlit

Desenvolvemos um sistema completo em Streamlit com duas abas principais para validação em ambiente real:

1. **Classificação em Tempo Real**:
   - Seleção dinâmica de embedding (BERT/TF-IDF) e modelo (SVM/XGBoost).
   - Exibição de score de confiança e distribuição de probabilidades.
   - Integração opcional com LLM para gerar explicações da predição.
2. **Dashboard de Monitoramento**:
   - Métricas agregadas e gráficos interativos (Plotly).
   - Trade-off Performance vs Eficiência visualizado em tempo real.
   - Distribuição de scores por modelo (Box Plot).

6.2 Logging e Banco de Dados

Todas as predições são persistidas em dois formatos para redundância e escalabilidade:

- **CSV**: `logs/predicoes.csv` para auditoria rápida.
- **SQLite**: `logs/predicoes.db` para consultas estruturadas e alta performance.

Os logs capturam timestamp, texto (hash), classe predita, score de confiança e latência, permitindo o monitoramento de *data drift* e performance em produção.

7 Discussão

7.1 Comparação entre Embeddings

TF-IDF (Esparso)	**BERT (Denso)**
+ Eficiência computacional extrema (0.14ms)	+ Compreensão semântica profunda
+ Tamanho reduzido (0.18 MB)	+ Performance superior (F1=1.0)
+ Alta interpretabilidade (tokens visíveis)	+ Robustez a variações lexicais
- Não captura contexto semântico	- Cold Start significativo (2.23s original)
- Falha em ambiguidade lexical	- Maior uso de memória (0.88 MB)

7.2 SVM vs XGBoost

O **SVM** demonstrou superioridade consistente sobre o XGBoost neste dataset (F1=0.968 vs 0.704 no TF-IDF). O kernel linear do SVM é matematicamente mais adequado para lidar com a alta dimensionalidade e esparsidade do TF-IDF, enquanto o XGBoost sofreu para encontrar cortes ótimos nas árvores de decisão com a base de dados limitada (315 amostras).

7.3 Limitações e Trabalhos Futuros

- **Base de Dados:** 315 amostras podem não ser representativas de todas as nuances do português.
- **Overfitting:** O F1=1.0 do BERT+SVM pode indicar overfitting; testes em bases externas são necessários.
- **Expansão:** Coletar mais dados e testar ensembles combinando TF-IDF e BERT.

8 Conclusão e Recomendações

8.1 Resposta à Hipótese

A hipótese de que o BERT justifica seu custo depende do contexto:

- **Sim**, para aplicações críticas onde cada erro tem alto custo (ganho de 3.2% em F1 e robustez semântica).
- **Não**, para sistemas de alta escala/baixa latência, onde o TF-IDF+SVM entrega 96.8% da performance com fração do custo de memória e inicialização instantânea.

8.2 Contribuições

Este trabalho entregou um sistema completo (*NewsLens AI*) com interface Streamlit, monitoramento de logs, e uma análise rigorosa demonstrando que, embora representações densas sejam o estado da arte em performance, métodos clássicos bem otimizados permanecem altamente competitivos para classificação de notícias.

Referências Bibliográficas

CHEN, Tianqi; GUESTRIN, Carlos. XGBoost: A Scalable Tree Boosting System. *In:* PROCEEDINGS of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [*S. l.: s. n.*], 2016. p. 785–794.

CORTES, Corinna; VAPNIK, Vladimir. Support-vector networks. **Machine learning**, v. 20, n. 3, p. 273–297, 1995.

DEVLIN, Jacob *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. [*S. l.: s. n.*], 2019. p. 4171–4186.

REIMERS, Nils; GUREVYCH, Iryna. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *In:* PROCEEDINGS of the 2019 Conference on Empirical Methods in Natural Language Processing. [*S. l.: s. n.*], 2019. p. 3982–3992.

SOUZA, Fábio; NOGUEIRA, Rodrigo; LOTUFO, Roberto. BERTimbau: Pretrained BERT Models for Brazilian Portuguese. **arXiv preprint arXiv:2010.05313**, 2020.

A Estrutura do Projeto

```
newslens-classifier/
+-- data/
|   +-- raw/                # Base original
|   +-- processed/          # Dados pre-processados
|   +-- embeddings/         # Embeddings salvos (.npy, .npz)
|   +-- novos/              # Novos textos para producao
+-- logs/
|   +-- predicoes.csv       # Log de predicoes
+-- models/                 # Modelos treinados (.pkl)
+-- src/                    # Codigo fonte (preprocessing, training)
+-- scripts/                # Scripts de execucao e otimizacao
+-- apps/                   # Interface Streamlit
+-- reports/                # Relatorios e analises
```

B Fluxograma do Pipeline NewsLens

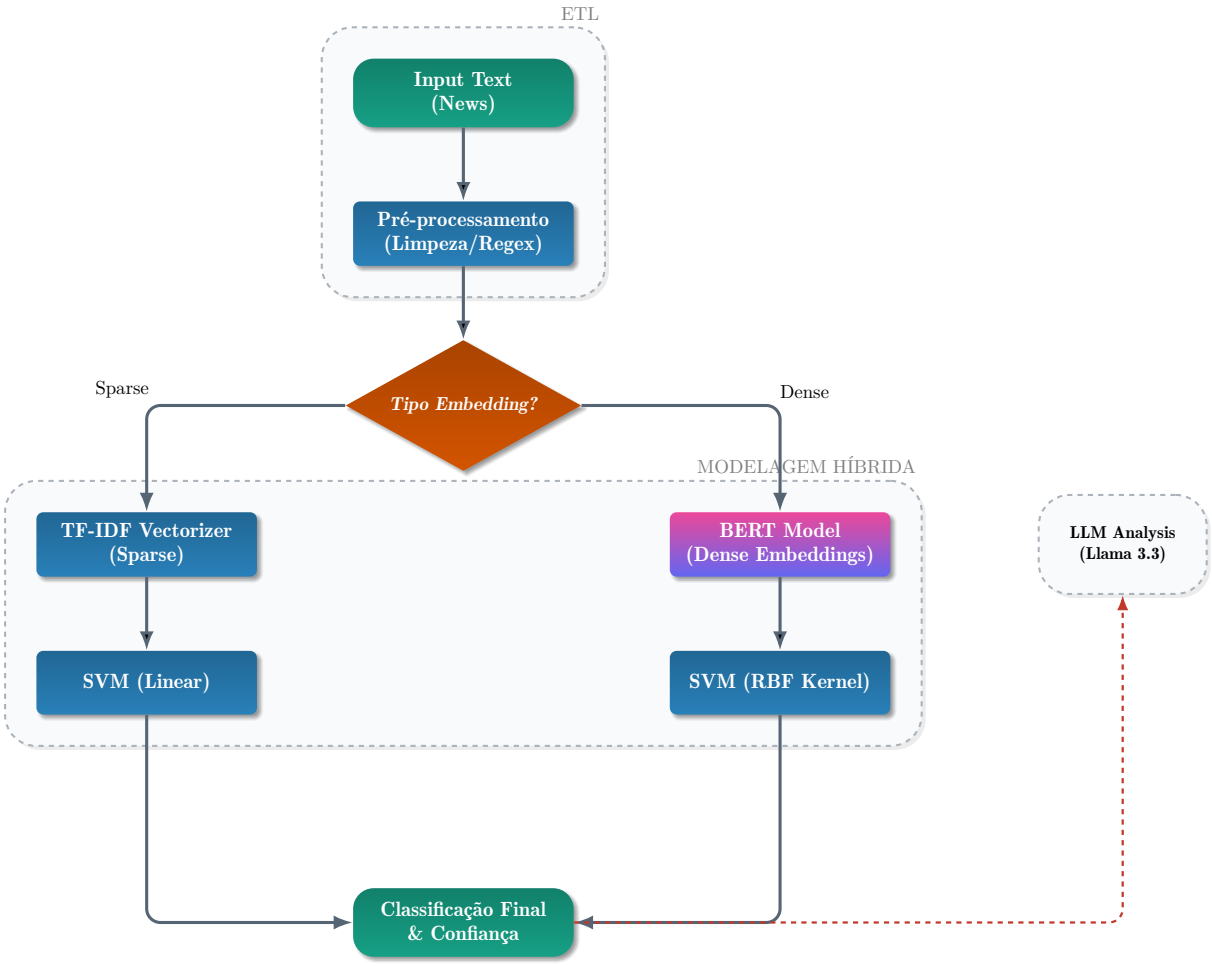


Figura 3: Arquitetura do Sistema NewsLens AI.