

ESPECIALIZAÇÃO EM DEEP LEARNING 2023.2

Processamento de Linguagem Natural

Projeto Final

**Douglas Contente Pimentel Barbosa
José Paulo Cauás Tenório**

Sumário

Introdução

Base de Dados

Modelo I: SVM + Bag of Words

Modelo II: SVM + Embeddings

Modelo III: BERT

Conclusões

Introdução

- **OBJETIVO:** Desenvolver e testar o desempenho de três modelos de Processamento de Linguagem Natural para análise de sentimento das avaliações de um produto da Internet

Introdução

- **MODELOS AVALIADOS:**
- **SVM + BoW (Bag of Words)**
- **SVM + Embeddings**
- **BERT (Bidirectional Encoder Representations from Transformers)**

Base de Dados



- PRODUTO: Aplicativo Chordify*
- FONTE: Google Play Store
- AMOSTRA: 1000 avaliações (texto e nota)
- IDIOMA: inglês
- NOTA MÉDIA: 4,4

**https://www.google.com/url?q=https%3A%2F%2Fplay.google.com%2Fstore%2Fapps%2Fdetails%3Fid%3Dnet.chordify.chordify%26hl%3Dpt_BR*

Base de Dados



- Script de coleta dos dados disponível em [1]
- Formato Excel (.xlsx)
- Notas de 1 a 3 → Avaliação Negativa
- Notas de 4 a 5 → Avaliação Positiva
- Dataset desbalanceado com 70.2% de avaliações Positivas e 29.8% Negativas.

[1]

https://colab.research.google.com/drive/1-mtsSErLojsq3X4o0ieZgelNDnEBHFV_?usp=sharing

Modelo I

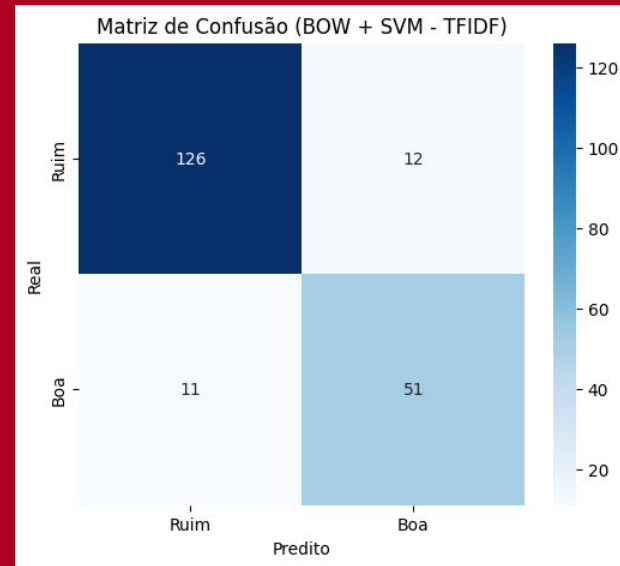
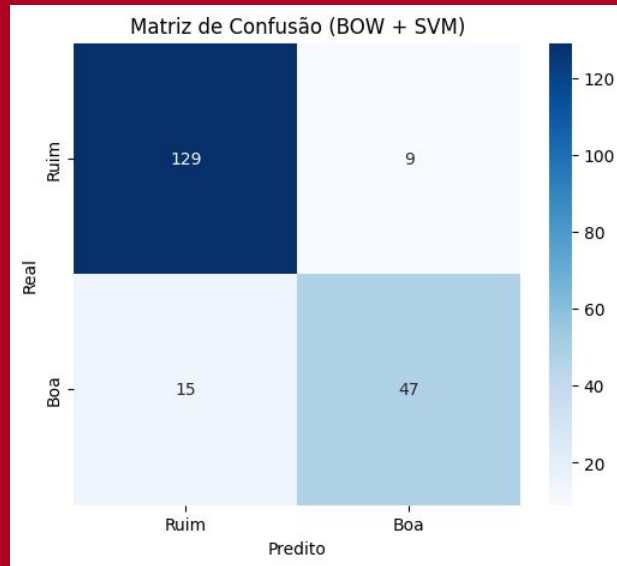
SVM + BoW

Etapas

1. Pré-processamento:
 - 1.1. Ajuste de casefold.
 - 1.2. Remoção de caracteres especiais.
 - 1.3. Remoção de Stopwords com *nltk*.
 - 1.4. Lemmatização com modelo spaCy.
2. Treinamento do modelo (20% para teste).
3. Vetorização com *CountVectorizer* e *TfidfVectorizer*
4. Classificação com Máquinas de Vetor de Suporte (SVM) - pesos balanceados.

Modelo I

SVM + BoW



Modelo I

SVM + BoW

- Resultados SVM + BOW

	precision	recall	f1-score	support
Boa	0.90	0.93	0.91	138
Ruim	0.84	0.76	0.80	62
accuracy			0.88	200
macro avg	0.87	0.85	0.86	200
weighted avg	0.88	0.88	0.88	200

- Resultados SVM + BOW-TFIDF

	precision	recall	f1-score	support
Boa	0.92	0.91	0.92	138
Ruim	0.81	0.82	0.82	62
accuracy			0.89	200
macro avg	0.86	0.87	0.87	200
weighted avg	0.89	0.89	0.89	200

Modelo II

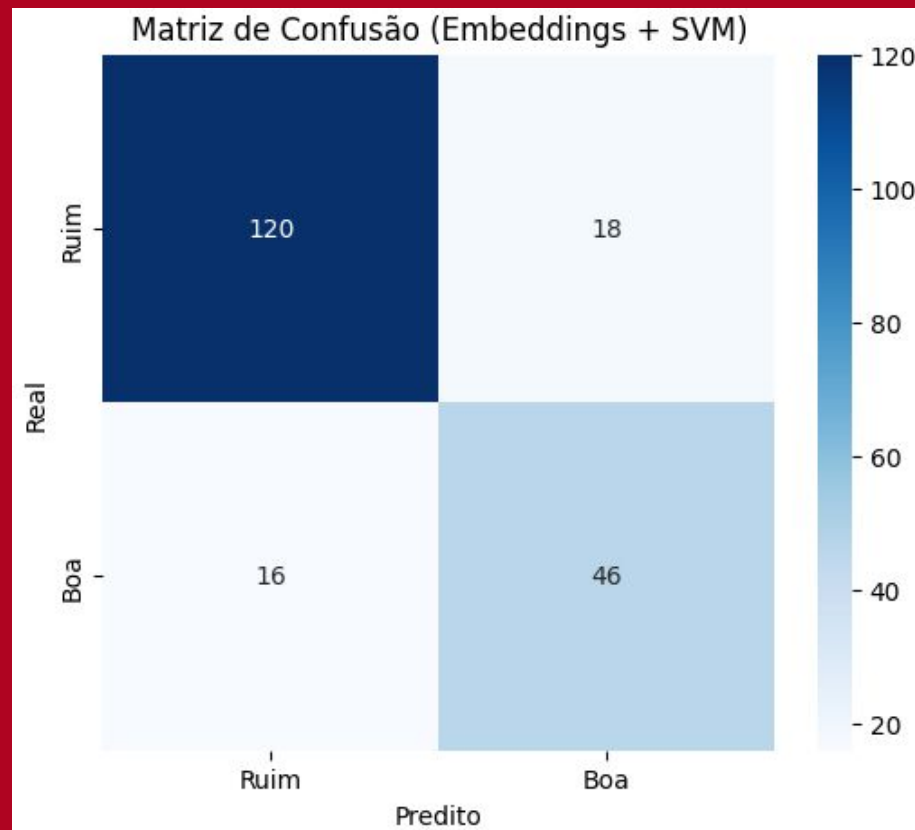
SVM + Embeddings

Etapas

1. Instalação e carregamento da biblioteca média do spaCy (`en_core_we_md`).
2. Pré-processamento:
 - 2.1. Ajuste de casefold.
 - 2.2. Remoção de caracteres especiais.
 - 2.3. Remoção de Stopwords e lematização com spaCy.
3. Treinamento do modelo (20% para teste).
4. Vetorização usando embeddings do spaCy e Word2Vec.
5. Classificação com Máquinas de Vetor de Suporte (SVM) - pesos balanceados.

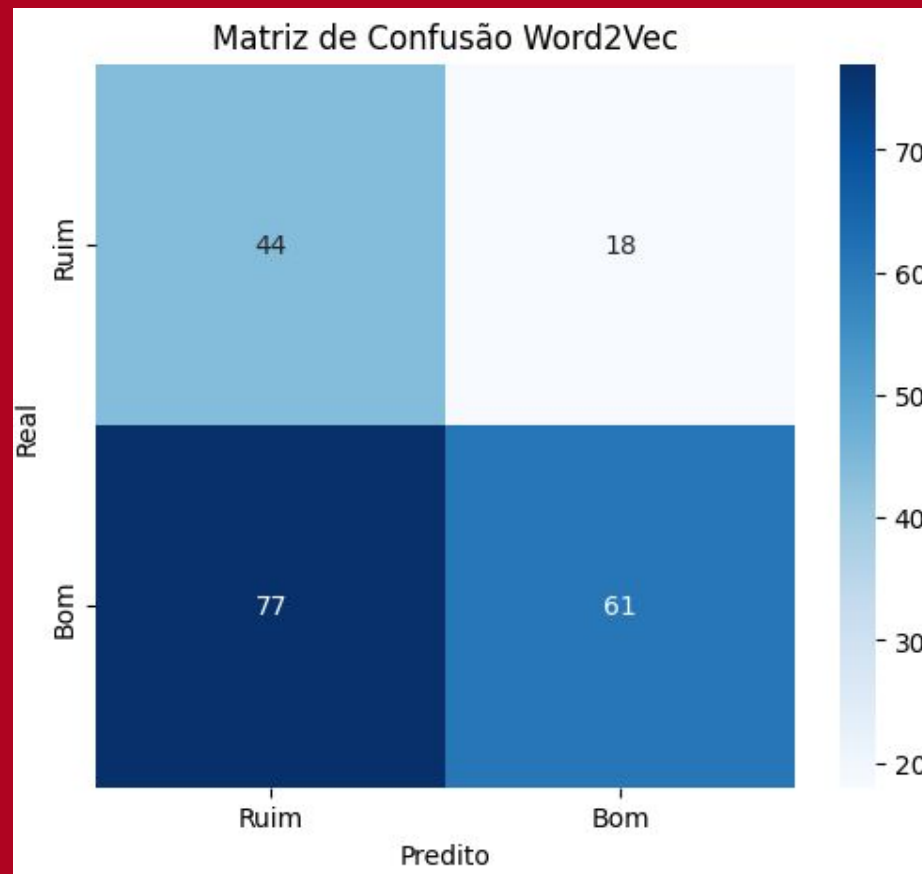
Modelo II

SVM + Embeddings



Modelo II

SVM + Embeddings



Modelo II

SVM + Embeddings

- Resultados SVM + Embeddings (pré-treinados - spaCy)

	precision	recall	f1-score	support
Boa	0.92	0.86	0.89	138
Ruim	0.73	0.84	0.78	62
accuracy			0.85	200
macro avg	0.83	0.85	0.84	200
weighted avg	0.86	0.85	0.86	200

- Resultados SVM + Word2Vec (puro)

	precision	recall	f1-score	support
Ruim	0.36	0.71	0.48	62
Bom	0.77	0.44	0.56	138
accuracy			0.53	200
macro avg	0.57	0.58	0.52	200
weighted avg	0.65	0.53	0.54	200

Modelo III

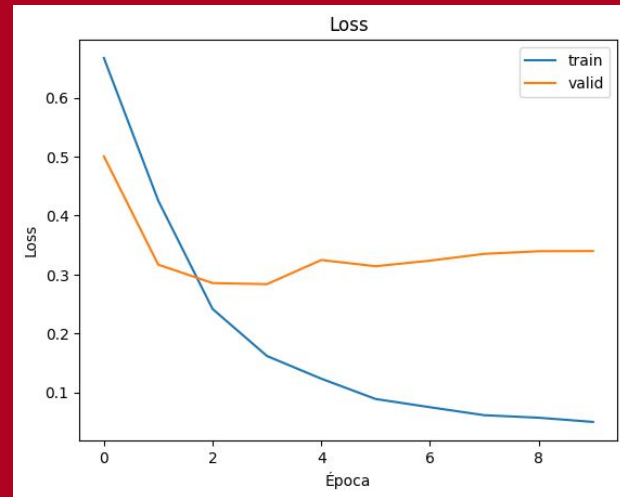
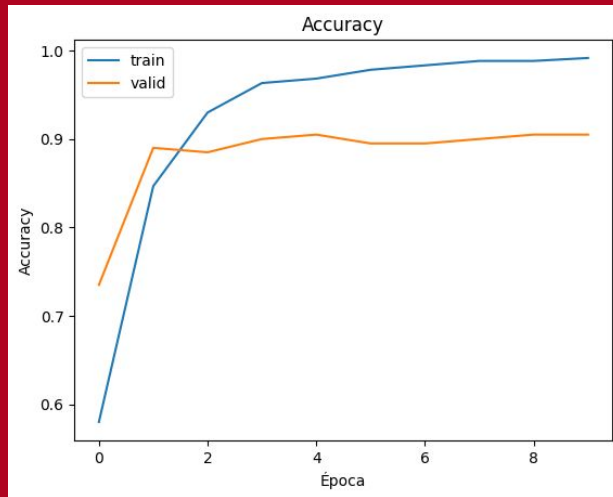
BERT

Etapas

1. Pré-processamento:
 - 1.1. Converter os dados para uma classificação binária e criar os dicionários (Train, Val e Test)
 - 1.2. Tokenizar as sentenças.
 - 1.3. Inserir os tokens [CLS] e [SEP].
 - 1.4. Nivelar o tamanho das sentenças (Padding)
 - 1.5. Criar as máscaras (interesse do modelo)
2. Carregar modelo BERT pré-treinado (uncased_L-12_H-768_A-12)
3. Configurar classificação com Head do HuggingFace para classificação de sentenças.
4. Configurar otimizador (épocas, batch, etc)
5. Treinar modelo (60% para Treino e 20% para validação).
6. Avaliar desempenho com dados de Teste (20%).

Modelo III

BERT



- 600 amostras de treino
- 200 amostras de validação
- 200 amostras de teste
- Aproximadamente 1100 segundos / época

Modelo III

BERT

- RESULTADOS

	precision	recall	f1-score	support
0	0.85	0.85	0.85	47
1	0.95	0.95	0.95	153
accuracy			0.93	200
macro avg	0.90	0.90	0.90	200
weighted avg	0.93	0.93	0.93	200

Comparação

Modelo	Precisão	Recall	Accuracy	F1
SVM+BOW	0,88	0,88	0,88	0,88
SVM+BOW-TFIDF	0,89	0,89	0,89	0,89
SVM+EMB	0,86	0,85	0,85	0,86
SVM+W2V	0,65	0,53	0,53	0,54
BERT	0,93	0,93	0,93	0,93

Conclusões

- As 03 estratégias obtiveram resultados úteis a finalidade aqui tratada (exceto W2V).
- O modelo W2V puro apresentou baixo desempenho. Indicativo de que mais dados seriam necessários para treinar o modelo nessa condição.
- O dataset é composto por frases curtas e de caráter similar. Isso explica o bom resultado dos modelos de BoW (melhor modelo do BoW foi com vetorização TF-IDF).

Conclusões

- O BERT levou um longo tempo para realizar o fine tuning, mesmo com apenas 10 épocas e 600 amostras (aproximadamente 1.100 seg/época)
- Os desempenhos dos modelos podem ter sido afetados por
 - a) existência de avaliações em idiomas distintos do inglês. Ex: “Bahut acche Hain”.
 - b) existência de avaliações 3 estrelas com conteúdo positivo. Ex: “Very nice app”.

Trabalhos Futuros

- Avaliar os modelos para uma base de avaliações em português utilizando o BERTimbau*, por exemplo.
- Avaliar os impactos de:
 - a) considerar as avaliações 3 estrelas como positivas;
 - b) Excluir as avaliações 3 estrelas;
 - c) Tratar as avaliações 3 estrelas uma a uma.

**<https://huggingface.co/neuralmind/bert-base-portuguese-cased>*



**Muito
obrigado!**

dcpb@cin.ufpe.br

jpct@cin.ufpe.br

50
anos



Centro de
Informática
UFPE