Treinamento de modelos NLP para análise das avaliações dadas pelos consumidores a um produto “Tablet”

Processamento de Linguagem Natual

ALUNA:

Maria Eduarda Neves

Especialização Deep Learning

Centro de Informática UFPE

Universidade Federal de Pernambuco

ESCOPO:

1. Introdução
2. Objetivo do trabalho
3. Metodologia
4. Resultados
5. Conclusão

1. Introdução
2. Objetivo do trabalho
3. Metodologia
4. Resultados
5. Conclusão

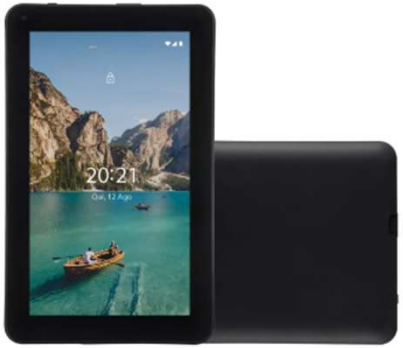
Problema Inicial

- Com o avanço da tecnologia e maior acesso de pessoas à internet torna-se importante compreender melhor o processamento de linguagem natural;
- Visa-se, com isso, compreender quando ocorrem discursos de ódio na internet, para tentar evita-los

Banco de Dados

Especialização Deep Learning — Centro de Informática UFPE — Universidade Federal de Pernambuco

- A coleta do banco de dados se deu através da API do mercado livre.
  - Através dela, coletou-se o código ID do produto e foi possível extrair um banco de dados em formato .csv referente às avaliações;
- Produto de Interesse:
  - Tablet Mirage 7 Pol 64gb 4gb ram Quad core wi-fi cor Preto;
  - ID produto: MLB4241052396



Banco de Dados

- 26 colunas com variáveis referentes às informações fornecidas pelos consumidores sobre o produto
- Eliminação das demais colunas, ficando apenas com as de “content” e “rate”
- DataFrame final: 1255 linhas e 2 colunas;

Banco de Dados

Demonstração do DataFrame :

content	rate
Comprei pro meu filho de 5 anos. Baixei netfli...	4
Um bom tablet. Comprei para vídeos e leitura. ...	5
Gostei muito, fácil de manusear, serve para o ...	5
O produto e bom sim, tem suas limitações, mas ...	5
Muito bom produto recomendo vou comprar outro.	5
...	...
O produto é bom,,, o problema é a tela q deixa...	3
Imagem e auto falante não é bom.	3
Ótimo.	1
Fica desligado toda hora sozinho não gostei.	1
Desculpe equipamento é ruim.	1

Quantitativo por “rate”

rate	count
5	595
1	234
4	181
3	160
2	85

- Total: 1255 avaliações
- Banco de dados desbalanceado

1. Introdução
2. **Objetivo do trabalho**
3. Metodologia
4. Resultados
5. Conclusão

Objetivos

- Fazer a análise de NLP para um determinado produto
  - Coletar informações de texto e nota;
- Compreender como se comportam diferentes classificadores quando testados em um mesmo conjunto de dados

1. Introdução
2. Objetivo do trabalho
3. **Metodologia**
4. Resultados
5. Conclusão

Metodologias Aplicadas

- Processamento de texto a fim de deixar o DataFrame melhor adaptado para os classificadores.
- Utilização de três classificadores de NLP diferentes para o processamento de texto e nota do produto escolhido

1. SVM + Bag Of Words (BOW)
2. SVM + Embeddings
3. Bert

Processamento de Texto

1. Normalização
   - Utilizando Spacy
   - Transformação de maiúsculas e minúsculas
   - Remoção de caracteres especiais (pontuação)
   - Remoção de algumas StopWords (‘a’, ‘o’, ‘de’, ‘e’)
     - Se encontravam entre os termos de maior frequência
2. Tokenização
   - Utilizando NLTK Tokenizer
3. Lematização
   - Foi feito um teste, fazendo a lematização das frases, entretanto o resultado encontrado não foi satisfatório, optou-se então por permanecer sem esta etapa do processamento

Quantitativo pós processamento:

- Maior vetor de palavras:
  - 108
- Total de Tokens:
  - 14552
- Tamanho do vocabulário:
  - 1832

SVM + BOW

- É uma representação baseada em pesos
- Não guarda a ordem dos elementos
- O dataframe va ter em quantidade de colunas, o tamanho do vocabulário extraído
- Modelo utilizado:
  - SVC → Support Vector Classifier
- Parâmetros utilizados com o auxílio do GridSearch:
  - C: 100; gamma: 0.1; kernel: rbf
  - Conjunto de teste representou 20%

SVM + EMBEDDINGS

- É uma representação distribuída das palavras
- Cada unidade linguística vai ser representada por um vetor denso
  - Tamanho do vetor escolhido: 75
- Utilizado o Word2Vec para gerar os vetores das palavras
- Foi feito uma padronização das frases, com zero padding, para que todas tivessem a mesma dimensão que aquela de maior dimensão (108)
- Modelo utilizado:
  - SVC → Support Vector Classifier
- Parâmetros utilizados com o auxílio do GridSearch:
  - C: 100; gamma: scale; kernel: linear
  - Conjunto de teste representou 20%

BERT

- Tem a função de mapear a representação contextual bidirecional de toda a frase
  - Ou seja, ele prevê palavras com base nas anteriores e nas seguintes
- Conjunto de dados foi dividido em três
  - Treinamento (0.70); Validação (0.20); Teste (0.10)
- Foi importado através do transformers
  - BertTokenizer → Realiza a tokenização das palavras
  - Modelo pré-treinado:
    - BertForSequenceClassification.from_pretrained('neuralmind/bert-base-portuguese-cased')

1. Introdução
2. Objetivo do trabalho
3. Metodologia
4. **Resultados**
5. Conclusão

Resultados

Classification Report

Rating - Class	BOW				Embeddings				BERT			
	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support
1	0.62	0.87	0.72	39	0.51	0.69	0.59	39	0.71	0.77	0.74	31
2	0.60	0.60	0.60	10	0.36	0.80	0.50	10	0.00	0.00	0.00	10
3	0.56	0.64	0.60	28	0.62	0.54	0.58	28	0.27	0.30	0.29	10
4	0.38	0.13	0.20	38	0.39	0.24	0.30	38	0.38	0.33	0.36	15
5	0.82	0.85	0.83	136	0.82	0.78	0.80	136	0.76	0.83	0.79	60
accuracy			0.71	251			0.66	251			0.65	126
macro avg	0.60	0.62	0.59	251	0.54	0.61	0.55	251	0.42	0.45	0.43	126
weighted avg	0.68	0.71	0.68	251	0.67	0.66	0.65	251	0.60	0.65	0.62	126

Resumo Final

CLASSIFICADOR	ACC	F1-MACRO	F1-AVG
BOW	0.709163	0.589962	0.682820
EMBEDDINGS	0.657371	0.551792	0.653619
BERT	0.650794	0.434994	0.624807

- O F1-score leva em consideração tanto a precisão quanto o recall, enquanto a acuraria leva em consideração somente o número de amostras corretamente classificadas com relação ao total
- O f1-macro vai levar em consideração a média aritmética enquanto o f1-avg a média ponderada
- Devido ao desbalanceamento das classes, uma boa forma de analisarmos como o classificador se saiu é através da F1-AVG

1. Introdução
2. Objetivo do trabalho
3. Metodologia
4. Resultados
5. **Conclusão**

Conclusão

- Devido ao desbalanceamento das classes apresentadas, todos os três classificadores encontraram dificuldades devido à pouca quantidade de amostras apresentadas;
  - Bert não conseguiu classificar elementos da classe “rate = 2”
- Uma base de dados mais balanceada poderia trazer melhores resultados;
- Nos três classificadores analisados, eles tiveram mais facilidade de identificar os casos extremos (maior e menor nota).
  - Isto pode ser devido ao fato deles terem maiores quantidades de amostras ou por possuírem palavras fortes que ajudam em suas classificações;
- **Por fim, classificador que apresentou melhores resultados entre os três analisados foi o SVM + BOW**

Obrigado!

ALUNA:

Maria Eduarda Neves

Especialização Deep Learning

Centro de Informática UFPE

Universidade Federal de Pernambuco