Estudo de Caso – Comparação entre vetorização de textos utilizando transferlearning DistilBERT e CountVectorizer/Tfidf

Grupo:

André Gomes Monteiro RM: 89168
Leonardo Aranha RM: 86919
Luara Maria Marino RM: 89375
Renato Kenji Yamashiro RM: 88847

Introdução

Natural language processing (NLP) é um ramo da inteligência artificial que foca em entender como os humanos escrevem e falam. Esta é uma tarefa é difícil pois envolve o tratamento de dados não estruturados. O modelo de aprendizado neste caso é supervisionado, isto é, para que a máquina possa "aprender o significado do texto", é necessário que cada texto tenha uma rótula contendo sentimento como também é necessário a vetorização das palavras.

A etapa de vetorização envolve a transformação de textos em números, na qual apenas a transformação de string em números não envolve apenas em enumera-los, é necessário também a contagem e analisar a frequência de cada termo para que tenha algum "sentido".

Objetivo

Este documento tem como objetivo a comparação do desempenho do modelo da regressão logística utilizando dois tipos de vetorização de texto: Transformer Distilbert e CountVectorizer/Tfidf.

Metodologia

O dataset vem do repositório do github como mostra o link abaixo:

https://raw.githubusercontent.com/prof-renato/data/main/humor_detection.csv

O objetivo do modelo supervisionado é prever o humor de cada corpo de texto, na qual está rotulada entre True e False na coluna humor.

O texto é tratado, retirando as pontuações e as stopwords.

A seguir, foi realizada a vetorização com ContVectorizer e Tfid para então realizar a regressão logística com 3000 épocas.

O CountVectorizer é basicamente a contagem de ocorrência de palavras num corpo de texto, onde os termos são representados em colunas e os documentos como linhas.

Tfidf (term frequency – inverse document frequency), onde TF é a frequência do termo num documento e transformada em matriz como mostra a equação a seguir:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

IDF é logaritmo do número de documentos divididos pelo número de documentos que contém a palavra w. O inverso da frequência determina o peso das palavras raras entre todos os documentos. Ela é definida pela seguinte equação:

$$idf(w) = log(\frac{N}{df_t})$$

E finalmente o TF é multiplicado pelo IDF como mostra a abaixo:

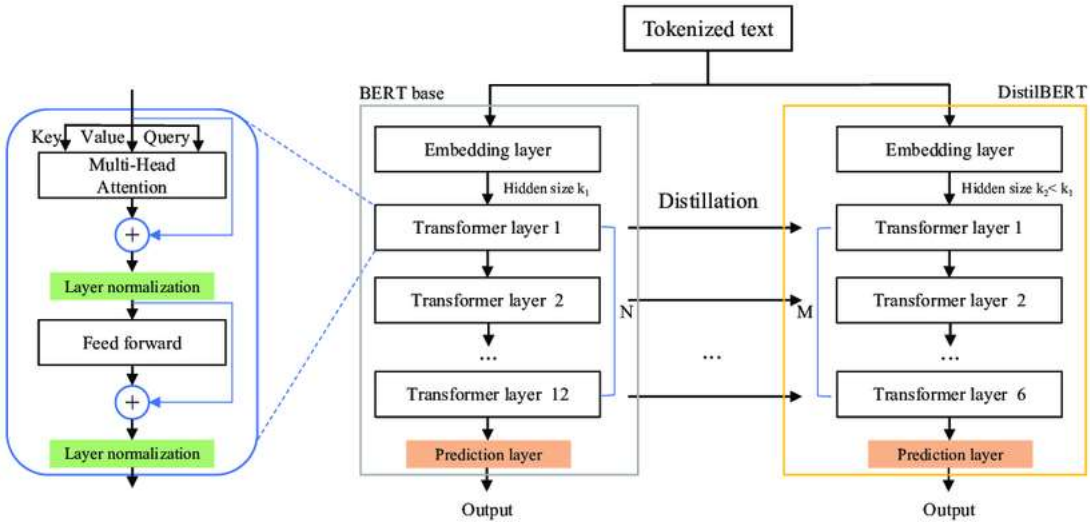
$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

O outro modelo foi treinado com os vetores resultantes do DistilBERT.

DistilBERT é um transformador pequeno, rápido, barato e leve pré -treinado pela base distilling BERT. A proposta do DistilBERT é solucionar o desafio de operar modelos pré-treinados que requerem grande infraestrutura.

A arquitetura é a mesma da BERT, onde os números de camadas (Deep Learning) foram reduzidos em dois. As operações de usadas na arquitetura Transformer (linear layer e layer normalization) foram otimizadas modernizando a estrutura de álgebra linear. As investigações realizadas mostraram que as variações na última dimensão do tensor (dimesão de tamanho oculto) tem um impacto menor na eficiência computacional do que outras variáveis como número de camadas. Portanto o foco do DistilBert é diminuir o número de camadas.

A imagem abaixo mostra a arquitetura do DistilBERT.



Conclusão

A tabela abaixo mostra as métricas para regressão logística utilizando CountVectorizer e TFIDF.

	precision	recall	f1-score	support
0	0.90	0.91	0.91	29349
1	0.91	0.90	0.91	28554
accuracy			0.91	57903
macro avg	0.91	0.91	0.91	57903
weighted avg	0.91	0.91	0.91	57903

Enquanto a tabela abaixo mostra as métricas de avaliação utilizando DistilBERT para vetorização e regressão logística.

	precision	recall	f1-score	support
False	0.97	0.97	0.97	29506
True	0.97	0.97	0.97	28397
accuracy			0.97	57903
macro avg	0.97	0.97	0.97	57903
weighted avg	0.97	0.97	0.97	57903

Utilizando o DistilBERT, temos a otimização na acurácia de 6%, não necessitando de grandes ajustes no momento da transformação em vetores.

Apesar da diminuição de passos para a vetorização, o tempo para a transformação demorou mais do que o pipeline utilizando CountVectorizer e TFIDF, necessitando também da GPU do Google Colab.

Referências:

[1] Dharmendra Sahani - Understanding CountVectorizer, Tfidftransformer & Tfidfvectorizer with Calculation

[2] Tracyrenee - How sklearn's CountVectorizer and TfidfTransformer compares with TfidfVectorizer

[3] https://www.kaggle.com/code/paoloripamonti/twitter-sentiment-analysis

[4] Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF - DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter - 2020