Sentiment analysis of airline-related tweets using BERT

Ana Beatriz Wanderley Massuh
Universidade de Brasília(UnB)
Brasil, Brasilia
200060783@aluno.unb.br

In this work we will use a deep learning model called BERT (Bidirectional Encoder Representations from Transformes), to perform the prediction of what emotion a snippet of text represents, such excerpts are posts on the social network Twitter in which several people demonstrate their opinions and feelings at all times, so the model after going through the training should be able to perform an analysis and demonstrate what is the greatest probability of a feeling being expressed in the text excerpt such as a criticism that may involve anger, for example, a bidirectional analysis of a given term such as the phrase "I had a terrible service on the plane", the unidirectional analysis of the word "terrible" would consider only "service . ..." but would not take into account "on the plane", whereas the use of BERT would consider "I had a ... service on the plane" generating a deeper analysis, therefore, our goal when using this method is to get a more accurate emotional analysis due to the "deeply bidirectional ".

I. Introduction

Twitter is a very present tool in people's lives nowadays, either to share pleasant experiences or to talk about unpleasant experiences and warn other people not to make the same "mistakes". But what is the relationship of this with natural language processing? The famous tweets can not only entertain people, but also help in the development of language processing models.

In this paper we will present the use of tweets referring to experiences with certain airlines and the tweets that were posted about them, these were classified into three classifications: neutral, positive and negative.

These tweets were used for training and testing in a model built through the BERT method (Bidirectional Encoder Representations from Transformers), for this work we created a multi-class classifier for sentiment analysis in text corpus, in this case up to 20 tokens.

II. Related works

One of the first steps in creating a model is to choose the methodology to be used. Here are some works related to the use of BERT in the execution of Natural Language Processing tasks.

Devlin *et al.* presented the BERT method, in the article *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* [3], published in 2019.In this paper the BERT (*Bidirectional Encoder Representations from Transformers*), seeks to alleviate unidirectionality constraints by conditioning the contexts left and right in all layers. In other words, this neural network is able to learn the ways of expressing human language. It offers search tools that do not give so much weight to the keywords themselves, as in other Word-Vec methods, but to the contexts of the search, inferring the real meaning of the search and thus being able to offer effective results aligned with what is actually being desired.

He uses MLM (*masked language model*)in pre-training, which randomly masks some of the input tokens, as its goal is to predict the original masked vocabulary id and thus predict the right representation and context, which allows us to pre-train a deep bidirectional transformer.Being its training unsupervised, the Feature−based(generation from left to right of the next words in the sentence, given a representation of the previous sentence) and *Fine-tuning* (which uses pre-trained word embedding parameters from unlabeled text).Being initialized first with the pre-trained parameters, which are readjusted using labeled task data, where each *downstream* has separately fitted models.For the authors the main contribution is to further generalize the findings to deep bidirectional architectures, allowing the same pre-trained model to successfully tackle a broad set of NLP tasks.

Another use of the BERT model is presented in the article Target-Dependent Sentiment Classification With BERT [1], in this article the author makes a comparison of deep learning models with the BERT model and performs a comparison of the accuracy of the models, in addition to testing a bert-based model of its own called TD-BERT.

The experiments demonstrated that complex neural networks had a good return of results with embeddings which does not fit as well when it comes to a BERT model, while incorporating input data into BERT results in an increase in performance in a stable and concise way, but exchanging representations with BERT does not necessarily generate an improvement in the performance of existing neural network models, since they are already much better adjusted with independent representation of context.

III. Proposed Method

Transformer-based methods allow neural networks to make connections between different regions of the text, modulated by these attention mechanisms, so that relatively distant but relevant The transformer-based methods allow neural networks to make connections between different regions of the text, modulated by these attention mechanisms, so that relatively distant but relevant regions receive the different treatment necessary to understand the text. The attention The attention mechanisms do not need to process the text sequentially, word by word, but can receive all the words at once, which words at once, which decreases processing time since it is possible to parallelize the computational operations involved. parallelize the computational operations involved. Understanding the full context of a word, that is, the terms that come before and after, and the relationships between them. Making it possible to develop algorithms for question answering or sentiment analysis or sentiment analysis algorithms, as it is being used.A flowchart of the basic architecture of a bert model is shown in 1.

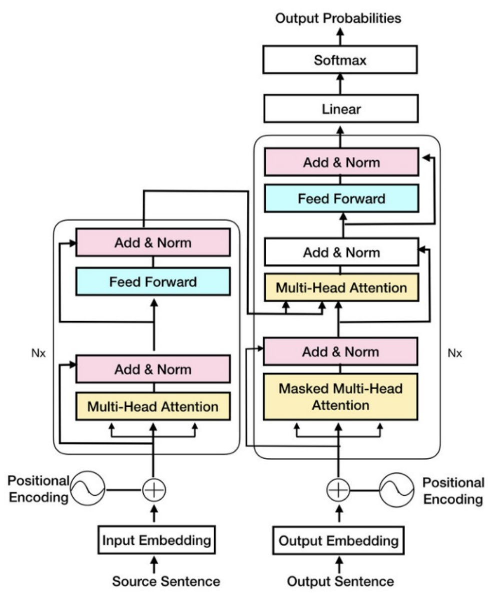

Fig. 1. Transformer model architecture. [4]

But the BERT model only requires the encoding mechanism, because its main goal is to create an efficient language model 2.

Following on from this, the 3 illustrates all the steps of the methodology that was used in this work.

A. Loading the Dataset

For this work we used the dataset provided on the OpenML website, The dataset consists of tweets referring to comments about their experience with some airlines. The texts were classified in 3 categories, they are: neutral, positive and negative. In the image 4 you can see the distribution of the classifications inside the corpus.

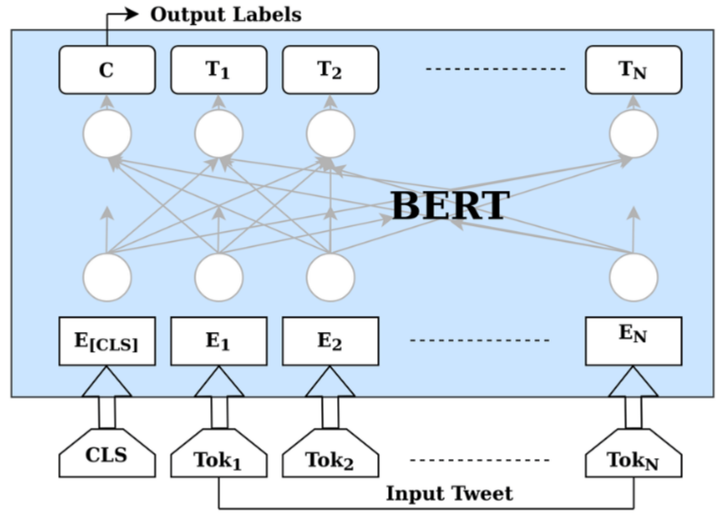

Fig. 2. BERT model architecture.

[2]

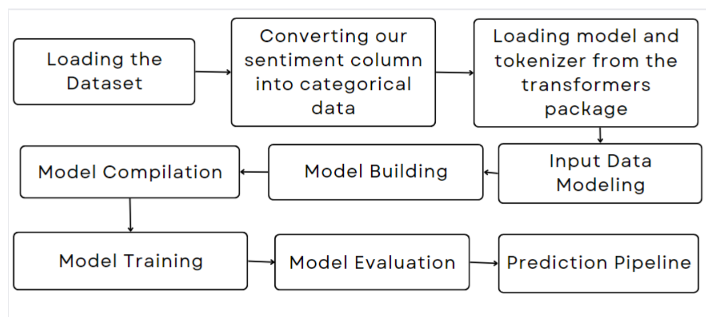

Fig. 3. Flowchart of the used methodology.

The data were separated into 75 percent for training and 25 percent for testing, and then irrelevant columns such as the ”username” column were removed.

B. Converting the sentimente column into categorical data

In this step it was necessary to map the sentiment labels into a dictionary (neutral:0, positive:1, negative:2) and then using tocategorical from tensorflow.keras.utils we converted the now integer sentiment columns into a data matrix for both the training and test data.

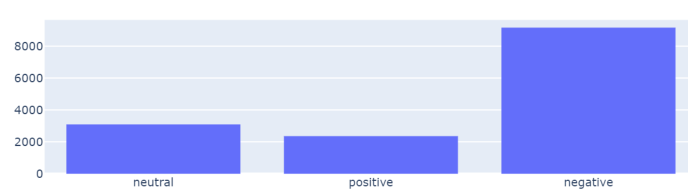

Fig. 4. Flowchart of the used methodology.

C. Loading the model and tokenizer from the transformers package

To process the execution it was necessary to install the transformers package, and from this package the AutoTokenizer was imported to convert the input text into tokens. TFBertModel was also imported, so that it could use the Bert-base-cased model, which is a model pretrained on a large corpus of English data.

D. Input Data Modeling

Before entering the training step, we need to convert the textual input data into BERT's input data format using a tokenizer. Tokenizer takes all the necessary parameters and returns tensor in the same format BERT accepts. After the data modeling, the tokenizer will return a dictionary containing "Inputxids", "attentionxmask" as keys to their respective data.

On average each tweet has between 10 and 20 words, for this reason maxlen equal to 20 was used, i.e. the reading was limited to the first twenty tokens of each tweet and those smaller than that will be padded.

E. Model Building and Compilation

To build the model we first imported all the necessary libraries from keras and used a functional API to design the model, BERT layers accept three input arrays, inputids(inputids means our input words encoding, then attention mask), attentionmask, Tokentypeids(tokentypeids is necessary for the question-answering model;tokentypeids was not passed).

To do the embedding we used the bert-base-case mentioned above, then we used GlobalMaxPool1D, then a dense layer of 128 neurons with the activation 'relu', then we used dropout, followed by another dense layer with now 32 neurons and for y a dense layer with 3 neurons, one for each possible answer.

For the compilation we should set the learning parameters, learningrate takes the value of $5e-05$ and the learning rate for the model will be significantly lower, Loss gets CategoricalCrossentropy since we are passing categorical data as the target and finally balancedaccuracy will take care of the average accuracy for all classes.

F. Model Training and Evaluation

Having configured the model having the xtrain and the ytrain we have the possibility to train the model, the training and improvement of the BERT model was done using two epochs, of batchsize equal to 36, were used only two epochs because the training data was very large and besides taking a long time, also there would be no need for many more epochs.

Next we need to run the test with the test data and check if our output is accurate.

G. Prediction pipeline

To perform the prediction, we will need to model the data in the same format as the training data and then call model.predict() which will return a probability matrix and then find the index with the highest chance and map it to its respective sentiment label.

IV. Experimental Results

For the experimentation process we did what was done in the Model Evaluation mentioned in the previous section, i.e., we used the model on the test data and then compared the predicted answers with the real answers. After that the sklearn.metrics classificationreport was used to verify the metrics used.

The table shows the results obtained with the model.

	neutral	positive	negative	accuracy	macro avg
precision	0.55	0.69	0.89		0.71
recall	0.60	0.68	0.87		0.72
f1-score	0.58	0.69	0.88	0.80	0.71
support	577	456	2628	3661	3661

	weighted avg
precision	0.81
recall	0.80
f1-score	0.81
support	3661

V. Conclusion

Taking into account the tools used for the construction of this work, we can conclude that we reached considerably good results. However, using the BERT tool, which is currently considered to be one of the best tools for this type of task, besides having a sizeable set of labeled data, the results obtained could be somewhat more usable.

References

[1] Zhengjie Gao, Ao Feng, Xinyu Song, and Xi Wu. Target-dependent sentiment classification with bert. *Ieee Access*, 7:154290–154299, 2019.

[2] Sunil Gundapu and Radhika Mamidi. Transformer based automatic covid-19 fake news detection system. *arXiv preprint arXiv:2101.00180*, 2021.

[3] Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.