Sentiment Analysis on Amazon Reviews

19BCE082 – Ayush Jain
19BCE084 – Jaishnav Kapadia
19BCE085 – Jay Patel
Institute of Technology,
Nirma University,
Ahmedabad, India

Abstract

The world we see nowadays is becoming more digitalized. In this digitalized world e-commerce is taking the ascendancy by making products available within the reach of customers where the customer doesn't have to go out of their house. For selecting a product, a customer needs to go through thousands of reviews to understand a product.

Sentiment Analysis is basically classification technique in which we apply machine learning algorithms on text-driven databases to classify positive and negative sentences. We will use Sentiment Analysis techniques to check whether it performs well on product review as well or not. For that, we will use Amazon product reviews, which are taken from Kaggle. We are going to use Google's BERT algorithm to classify and check accuracy for sentiment analysis. It is a pre-trained model so it will take less time to fine tune the dataset.

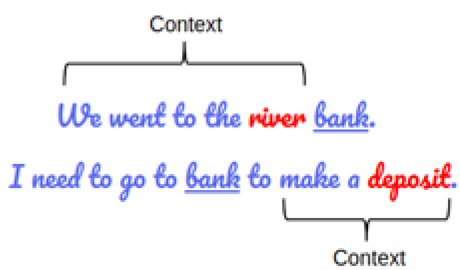
Keywords—Sentiment Analysis, BERT, Classification, Review, Kaggle etc.

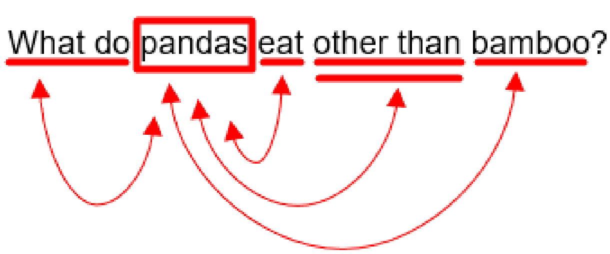
1 INTRODUCTION

People are exchanging things through various e-commerce websites since the world's commercial site has almost completely transitioned to an online platform. As a result, evaluating items before purchasing them is a regular scenario. Customers are also more likely to buy a product based on reviews these days. As a result, evaluating the data from those customer reviews in order to make the data more dynamic has been an important field in recent years. Reading hundreds of reviews to understand a product in this age of developing machine learning based algorithms is time consuming, but we can polarize a review on a certain category to understand its popularity among customers all over the world. The goal of this research is to classify customers' positive and negative comments on various items and to develop a supervised learning model to polarize a huge number of evaluations. Over 88 percent of internet customers trust reviews as much as personal recommendations, according to Amazon research from last year. Any online item with a big number of favorable reviews conveys a strong message about the item's validity. Books, or any other online commodity, without reviews, on the other hand, creates a sense of distrust among potential customers. Simply said, more reviews appear to be more credible. People appreciate other people's opinions and experiences, and reading a review on a product is the sole method to learn what others think about it. Opinions gained from people' experiences with a certain topic. Negative ratings, on the other hand, frequently result in sales losses [2]. Understanding client input and polarizing properly across a big quantity of data is the objective for those who want to succeed. Similar work has been done using the Amazon dataset. [5] used opinion mining to understand the divided opinions around items in a limited selection of Amazon product review datasets. To label our datasets, we employed both a manual and an active learning technique in our model. Various classifiers are utilized in the active learning process to provide accuracy until a desirable level is reached. We processed those labelled datasets once we got a suitable result. We collected characteristics from the treated dataset, which were then categorized by several classifiers. To extract data, we employed a mix of two procedures: the bag of words approaches and tf-idf& Chi square approach for getting higher accuracy.

1.1 BERT for Text Classification

BERT stands for "Bidirectional Encoder Representations from Transformers". It is designed to pre-train bidirectional representation of text by jointly conditioning on both left and right context. As it is pre-trained, it can be fine-tuned faster. Here Bidirectional means it can learn from both the left and right side of the tokens during the training phase. Which is very useful to learn about the different meanings of the same word. You can see the different meanings of the word 'bank' in the below example.

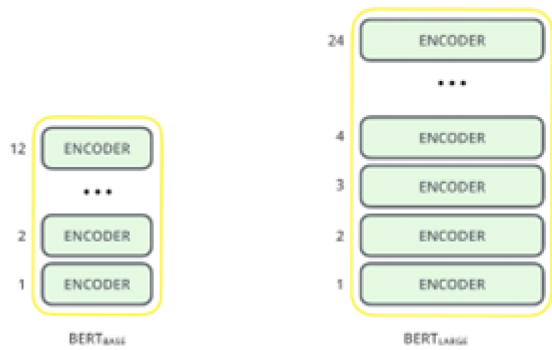




In the above example about pandas, we can see that the BERT Classifier, check in both directions and gives importance accordingly, the words "other than" are given higher importance than the rest, which increases its reference accuracy.

BERT architecture is built upon transformers. It has two variants.

1. **BERT base: -** it consists of 12 layers of transformers, 12 attention heads and 110 million parameters.
2. **BERT large: -** it consists of 24 layers of transformers, 16 attention heads and 340 million parameters.

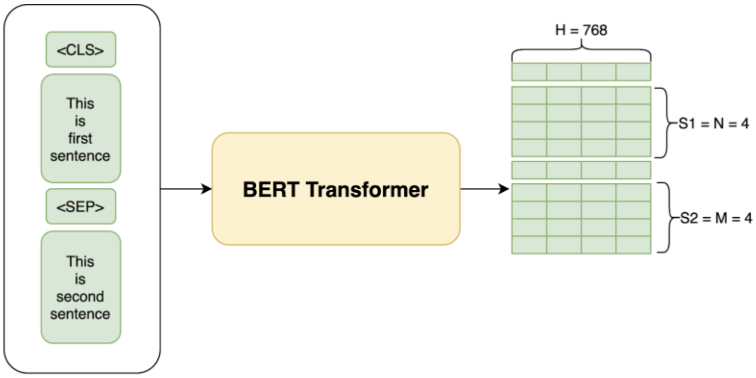


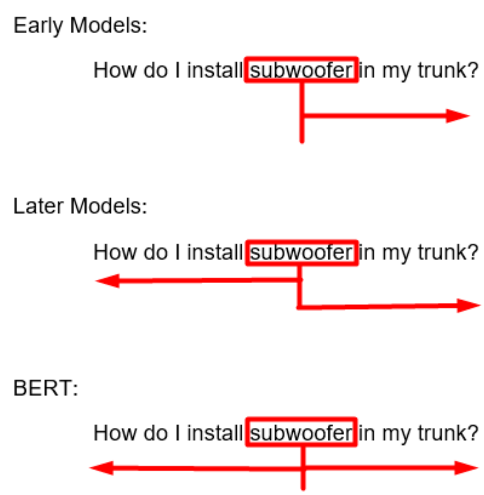
These transformers work as encoder blocks. Difference between these two models is in terms of accuracy only. BERT large gives little more accuracy than BERT base. But BERT large takes comparatively more time to train data than BERT base.

We have used BERT base architecture to train our data for Sentiment Analysis.

There are few requirements that needs to be fulfilled to use the context in transformer:

- We use special tokens called [SEP] and [CLS]. [SEP] token is embedded at the end of every sentence and [CLS] token is embedded at the start of every sentence to distinguish between two sentences.





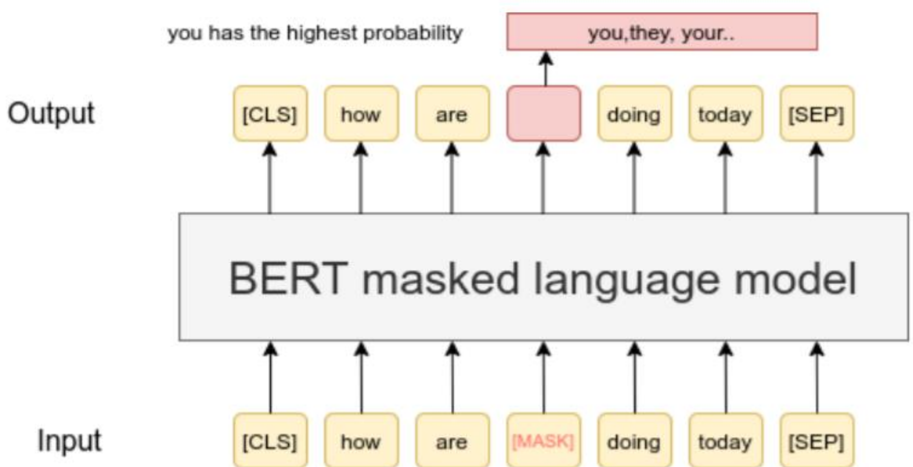
- Our dataset will have different length of sentences but BERT accepts only fixed length of sentence for uniformity of BERT vectors. We can set maximum length up to 512 tokens. For the adjustment of the vector length, we pad or truncate the tokens. for padding, a special token called [PAD] is used.

- Finally, there is the concept of attention mask. it is just an array consisting of 1s and 0s indicating which token is padded and which is not. 0 means it is a padded token [PAD] and it does not need to be considered in training.

And finally, BERT is pre-trained on two NLP task:

1. Masked Language Modeling:

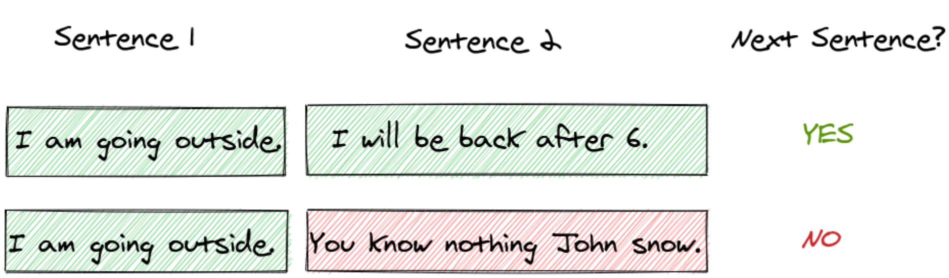
   Generally, models are trained as a practice of predicting the next word. We want a bidirectional model, so instead of predicting the next word, if we mask some words from the given sentence and try to predict that masked word then we can achieve a better result for the bidirectional model.



   Researchers generally replace 15% of the word with [MASK] to distract from focusing on specific positions. As this [MASK] token never appeared in the fine-tuning process, sometimes it is replaced with a random word or left unchanged.

2. Next Sentence Prediction:

   Aim of this task is to establish a relationship between two sentences. Here whatever dataset we take, 50% of data will be train data means the second sentence will be actually the next sentence of the first sentence. For the remaining dataset, the next sentence will be random from the corpus. We label them as 'IsNext' or 'NotNext'.



A. *Advantage of BERT*

- BERT already implements a lot about our language
- Quicker Development
- Less Data required
- Better Result
- It can perform operations like classification, NER, POS, Question Answering

B. Shortcoming of BERT

- BERT is very large
- Slow fine-tuning, Slow inferencing
- Jargon (Domain Specific Language)
- Not all NLP application
- It can not perform operations like Language Modeling, Text Generalization, Translation

Our Approach and Related Work

In this Sentiment Analysis of Amazon reviews, we have used Google's BERT text classification algorithm to train our model and get better results. It is pre-trained on a large dataset for encoding language. So we need less data to train it for specific tasks. We are going to use Kaggle's dataset that has Amazon's 4 million reviews. But as it is very large for our purpose and does not have that much computational efficiency to train it, we will use 100000 reviews to train our model in which around 515 of reviews are positive.

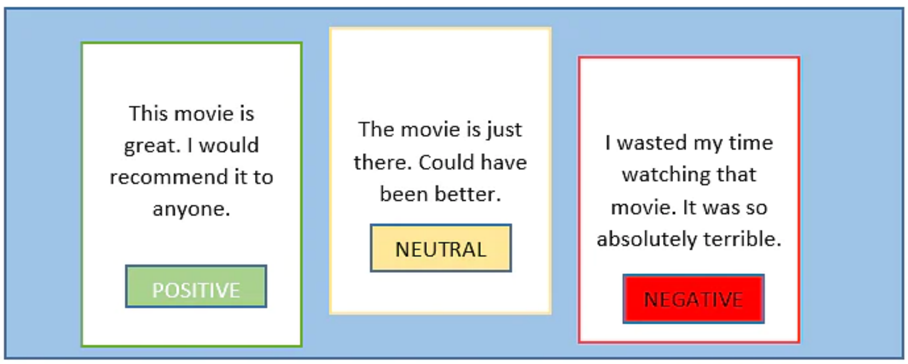
Even 100000 reviews also need very high computational power, we will need a good GPU. So, we trained this model in Google Colab and used GPU "Tesla T4" which is provided by Google itself. We specify it as a device for future purpose. Then the Transformers package of Hugging Face library is needed for the Py torch interface to work with BERT. Hugging Face library is widely accepted for this purpose.

BERT Tokenizer is used to Tokenize the sentence and to convert tokens to ids. Before moving forward, we need to do some modifications in data like adding special tokens [CLS] and [SEP] at start and end of each sentence, Padding or truncating sentence for the uniform length of vectors, and assigning attention mask to understand which token is supposed to be considered and which is not.

From our 1 lakh Reviews, we split them randomly as 90,000 as training set and 10,000 as Validation set. Giving all data as a whole to train the model will give bad impact for computation, so we gave input as batch size of 32. Now to train our classification model, first we need to use any pre-trained BERT model. We used "bert-base-uncased" which has all lower-case letters and it is a base model of BERT base and BERT large. Run this model using pytorch.

Then called the optimizer to choose the best value of different parameters like learning rate, no. of epochs, batch size etc. Then calculated the accuracy of our prediction against the label. Now load the data on the GPU and train the data. feed input data into the network and eliminate the loss via back propagation. Update the parameters using optimizer.step(). Do the same thing on the Validation set but instead of back propagating, calculate the loss and logits. After every epoch, back propagation takes place and updates the parameters which decrease the training loss and increase the accuracy.

epoch	Training Loss	Valid. Loss	Valid. Accur.	Training Time	Validation Time
1	0.16	0.12	0.95	1:05:04	0:02:34
2	0.08	0.14	0.96	1:05:20	0:02:36



If we want to see the graphical representation of this information then below is the graph.

Repeat all the procedures for the test set data which has 10,000 reviews and see what are the results for evaluation measures like accuracy, recall, F-score, support etc.

	precision	recall	f1-score	support
0	0.95	0.93	0.94	5160
1	0.93	0.95	0.94	4840
accuracy			0.94	10000
macro avg	0.94	0.94	0.94	10000
weighted avg	0.94	0.94	0.94	10000

Till now we saw results of different classifiers algorithms like LSVM (Linear Support Vector Machine), MNB (Multinomial Naive Bayes), Random Forest etc. All performed well with good accuracy as per study done by various students and researchers. Most of them found SVM to perform well over other classification algorithms. We used the BERT algorithm to train our data for sentiment analysis. We got a very good accuracy of about 94% for test data. As BERT is pre-trained in our language, it took less data to train specific tasks of sentiment analysis. However, training the model for more epochs may give better accuracy for train data, it may overfit the model.

It has some drawbacks like slow fine-tuning, can't perform Language translation tasks etc. So there is an area for research about Language translation, speeding up algorithm and other functionality as well.

Conclusion

BERT is a new algorithm proposed by google a few times ago. As it is pre-trained on a very large dataset, it needs very less data to train for specific tasks. It is much efficient in classification tasks. It is used for the next sentence prediction task as well which we can understand as we search in search engines and it suggests what might be the next words. it has overall very good performance than other algorithms.

REFERENCES

[1] Levent G¨uner, Emilie Coyne, and Jim Smit, "Sentiment analysis for Amazon.com reviews" KTH Royal Institute of Technology, Stockholm, 2019-03-015.

[2] Tanjim Ul Haque, Nudrat Nawal Saber, and Faisal Muhammad Shah, "Sentiment Analysis on Large Scale Amazon Product Reviews", Ahsanullah University of Science & Technology, IEEE International Conference on Innovative Research and Development (ICIRD), June, 2018.

[3] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin, "DocBERT: BERT for Document Classification," David R. Cheriton School of Computer Science University of Waterloo, arXiv:1904.08398v3 [cs.CL] 22 Aug 201

[4] Google colab notebook about Bert fine-tuning, "https://colab.research.google.com/drive/1pTuQhug6Dhl9XalKB0zUGf4FIdYFlpcX#scrollTo=nSU7yERLP_66,".

[5] "Amazon Reviews for Sentiment Analysis." [Online]. Available: https://www.kaggle.com/bittlingmayer/amazonreviews#train.ft.txt.bz2