FineTuning-based BERT for Spam Emails Identification

Qingyao Meng

Department of Computer Science, University of California, Davis, US.

qymeng@ucdavis.edu

Abstract. In this world of information explosion, people require more effective ways to filter useful information from millions of data. Email as one of the most frequently used form of communication, carries important messages, yet along with messages of fake news, misinformation and scams known as spam emails. Manually categorizing them from non-spam emails requires a lot of time and money and other human along with material resources. In order to deal with this, deep learning, or natural language processing models in particular, is introduced to categorize emails faster and cheaper. The Natural Language Processing model used here is called Bidirectional Encoder Representations from Transformers (BERT). Since BERT is already a pre-trained model, the main task is to do the Fine-Tune part on it, with a dataset that contains around 5000 emails (85% spam emails and 15% non-spam ones). After that the model is tested on a group of 5 emails including 3 commercials/spams and 2 non-spam emails. The result shows that this model could separate them by giving commercials scores closer to 1 (spread from 0.5 to 0.7) and non-spam emails scores close to 1(spread from 0 to 0.1). Therefore, it can be concluded that this model works on small sets of data.

Keywords: BERT, Spam Emails Identification.

1. Introduction

This is an era of information explosion. The amount of published information and data increases so rapidly that every day this study receive, actively or passively, a great amount of information in different forms like pictures, movies and especially texts. The data e.g. Emails and websites all consists of unstructured and disorganized content. However, it is difficult to extract usable value from them due to their ununiform structure. To manually sort these data or write programs to deal with it is possible but doing so would be difficult as it is expensive and time-consuming. This is where Text Classification comes in to save the day.

Text classification, also known as text tagging or text categorization, is the process of systematically grouping texts [1]. Several Text Classification models have been proved to be effective, for example Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). Although they have been proven to be effective in many tasks [2-5], certain problems exists [1]. For instance, it is hard for long sentence to be learned well by the model due to the problem called vanishing gradient. Additionally, while using CNN to train the model, overfitting, explosive gradients, and class imbalance demand a lot of training data. These problems could make the model perform worse.

One new state-of-the-art model is called BERT. BERT stands for Bidirectional Encoder Representation from Transformers. Introduced by Google in 2017 [6]. It is able to read a giving sentence from both directions to figure out what the “masked” word is, which is considered in many tasks due to its excellent performance.

In this regard, open source BERT model is used to do Text Classification. In particular, BERT is used to categorize Spam emails from Non-spam emails. To train the model, this paper would use a dataset that contains both spam (15% of total emails) and non-spam emails (85% of total emails). This process would convert the similarities of texts into numbers so that when the texts are similar, they would have a number closer to 1; and when they are totally different, they would have a number closer to 0. Moreover, as in real world time are limited, and not enough time is given to train a model on different datasets for every different task, so this paper would also like to research on instead of classifying emails, classifying advertisements and commercials based on the same training dataset (spam and non-spam emails).

2. Method

2.1. Dataset description

In this project, this study used the dataset provided by a dataset on Kaggle [7]. The original dataset contains around 5, 000 emails including 4, 825 non-spam emails and 747 spam emails.

The preprocessing consists of splitting the dataset. This study split the dataset into training and test data at a ratio around 80/20. The training data will be the ones to be used to train our BERT model, and the test data will be the ones to test our model’s accuracy.

2.2. Introduction for BERT

The goal of NLP techniques is often to understand naturally spoken human language. In the context of BERT, this means that the model needs to predict a word in the blank. To achieve this, a large specialized, labeled training data repository for training is required. This requires experts in the relevant field to painstakingly hand-label the data [6].

The BERT was considered in this study. Only unlabeled plain text samples, such as the entire English Wikipedia and the Brown corpus, were used for BERT's pre-use training. It continues to learn from unsupervised unlabeled material even when used in a real application (i.e., Google searches). Its pre-training serves as a foundational "knowledge" base. As a result, BERT can be modified in accordance with user preferences and the constantly growing collection of searchable documents. This method is known as transfer learning [6].

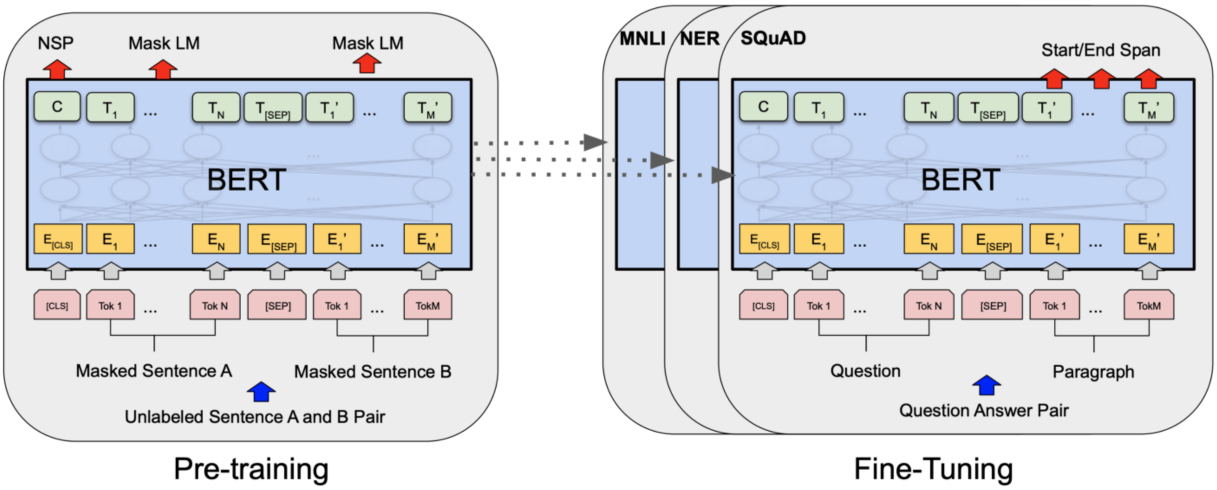

Figure 1. The structure of the BERT [8].

BERT was made feasible by Google's study of Transformers, as was already revealed. The transformer in the BERT can provide excellent comprehend ability for the text. Based on it, each word in the sentence can be examined in relation to every other word in the sentence. Transformer enables the BERT model to understand the whole context of a word, so that the searcher's intent can be better understood by looking at all nearby terms.

2.3. Implementation for BERT

The BERT model this study used is provided from Tersorflow Hub [9]. It is a pre-trained model trained on data from Wikipedia and BookCorpus.

BERT is pre-trained on two tasks, Masked Language Modeling and Next Sentence Prediction [10]. Its structure is shown in Figure 1. The Pre-Training procedure for Next Sentence Prediction requires to first generate input sentence by sampling two “spans” of texts, which are called “sentences” even though usually much longer than ones. The A embedding is applied to the first sentence, while the B embedding is applied to the second. When doing the "next sentence prediction" exercise, B is either the real sentence that follows A 50% of the time or a random sentence 50% of the time. The sample size is chosen so that the total length is less than 512 tokens. After WordPiece tokenization, the LM masking is done with a uniform masking rate of 15%, and incomplete word pieces are not given any extra attention [10]. The model is trained over 1,000,000 steps, or around 40 epochs, using a batch size of 256 sequences, which corresponds to the 3.3 billion words in the corpus. Adam uses a learning rate of 1e-4, 1 = 0.9, 2 = 0.999, L2 weight decay of 0.01, a warm-up period for the learning rate during the first 10,000 steps, and linear learning rate decay. All layers in this model have a 0.1 dropout probability. In line with OpenAI GPT, BERT employs a gelu activation rather than the usual relu. The mean masked LM likelihood and the mean next sentence prediction likelihood are added to determine the training loss [10].

2.4. Implementation details

After preprocessing the dataset, BERT model is imported from Tenserflow Hub [9]. Two sentences are tested as sample to show that they have been embedded as vectors. Similarly, few words are embedded and compared to get the cosine values by importing cosine_similarity function from sklearn.metrics.pairwise. When two words have a cosine value close to 1, they are very similar; when they have a cosine value closer to 0, they are very different. For example, two words e[0] = “banana” and e[1] = “grapes” have cosine_similarity([e[0]],[e[1]]) as 0.9911089. This makes sense as they are both fruits; e[3] = “Jeff Bezos” and e[4] = “Elon Musk” have cosine_similarity([e[3]], [e[4]]) as 0.98720354 as they are both famous entrepreneurs; comparing banana with Jeff Bezos gives the value 0.84 but it is not as close as 0.99 or 0.98 from examples before.

Next the model is built in a functional way. BERT model is built first, followed by the neural network layer which sets dropout as 0.1 and sigmoid as 1.

3. Result and Discussion

Table 1. Predicted result based on the test data.

Test data	Predicted result
Reply to win Â£100 weekly!....	0.65731551
You are awarded a SiPix Digital Camera!....	0.80427531
it to 80488. Your 500 free text messages are valid until 31 December 2005.	0.57503237
Hey Sam, Are you coming for a cricket game tomorrow	0.06529502
Why don't you wait 'til at least wednesday to see if you get your .	0.02541768

The model is trained in 5 epochs and after the last epoch the accuracy is 0.94. To test this model, a group of emails containing 3 spam emails (commercials) and 2 non-spam emails are chosen and presented in Table 1. The result shows that the first 3 spam emails have values 0.64, 0.71 and 0.57, closer to 1, and that the last 2 non-spam emails have values 0.06, 0.02, closer to 0.

Some lines of commercials are copied and tested and got similar results. The guess is that since most of spam emails are commercials, they share the same wording and tone. This could probably be the reason that this model gives similar results on both spam emails and commercials. In addition, some advanced models e.g. Transformer might be better to predict in this case, which will be considered in the future study [11].

4. Conclusion

This study uses the Natural Language Processing model BERT to try to separate spam emails and non-spam emails from a given group of emails. It first imports a dataset of around 5000 emails from Kaggle and splits them into training and test data, then imports BERT from Tensorflow Hub to get the embedded vectors for some sample statements, later builds the model functionally and trains it, at last is tested on few sample emails. The result of the model, tested on 5 emails containing 3 spam emails and 2 non-spam ones, is that spam emails get values closer to 1 (0.5, 0.6, 0,7) and non-spam emails get values closer to 0 (0.06, 0.02). It shows that the model works on small sets of data. What still needs to be improved is to have this model work on large sets of data and on other kinds of data. In real life, people deal with much larger number of emails and this model is expected to deal with them all in an acceptable period of time. The model right now runs forever on large set of data, causing computer to die, indicating that some part of the model needs to be improved (for example, better-trained). Another real-life problem is that time is luxury on training and no enough time will be given on training different kinds of data. Thus, this model, trained on the dataset of emails, is expected to work on not only emails but also TV commercials, misinformation, malicious and so on in the future.

References

[1] Monkeylearn 2022 What is text classificationhttps://monkeylearn.com/what-is-text-classification/

[2] Cai G et al. 2022 Privacy - preserving CNN feature extraction and retrieval over medical images International Journal of Intelligent Systems

[3] Yu Q et al. 2020 Improved denoising autoencoder for maritime image denoising and semantic segmentation of USV China Communications 17(3) 46-57

[4] Pandey S et al. 2022 RNN - EdgeQL: An auto - scaling and placement approach for SFC International Journal of Network Management e2213

[5] Zhang X et al. 2021 January Benchmarking LF-MMI, CTC And RNN-T Criteria For Streaming ASR. In 2021 IEEE Spoken Language Technology Workshop (SLT) (pp. 46-51) IEEE

[6] Tachtarget 2020BERT language model https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model

[7] Kaggle 2018 SMS Spam Prediction https://www.kaggle.com/code/jepsds/sms-spam-prediction/data

[8] Miro 2022 https://miro.medium.com/max/828/1*p4LFBwyHtCw_Qq9paDampA.png

[9] TFhub 2022 bert_en_uncased_preprocesshttps://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3

[10] DevlinJ et al. 2018 Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[11] Vaswani AShazeer N Parmar N Uszkoreit J Jones L Gomez A N ... & Polosukhin I 2017 Attention is all you need Advances in neural information processing systems 30.