BERT Analysis For Multi-class Text Classification

Kunal Patrikar[1], Hritik Belani[2,3], Veerja Kadam[3], and Kiran Kamble[3]

Walchand College of Engineering, Sangli http://www.walchandsangli.ac.in/

Abstract. BERT (Bidirectional Encoder Representations from Transformers) is a language representation model developed by Google Research. BERT is a successful example from the most recent generation of deep learning-based algorithms. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

This paper contains the study of BERT algorithm for multi-class text classification.BERT have multiple algorithms and we have chosen pre-trained distilbert-base-uncased and modified the model for our specific task.Then we prepared the data according to what the model architecture expects.Then we Fine-tuned the modified pre-trained model.

Keywords: BERT, · Transfer Learning, · Transformer.

1 Introduction

BERT is a language model that can be used directly to approach NLP tasks like summarization, question answering etc.Summarization is the task for forming shorter and more abstract text summary from the available text data without changing its semantic meaning.BERT is basically the transformer architecture trained to learn language representations, and conceived to be used as the main architecture to NLP tasks. It mainly differs from the precedent language models because its learned representations contain context from both sides of the sentences (left and right from the word itself).

BERT works in two steps, First, it uses a large amount of unlabeled data to learn a language representation in an unsupervised fashion called pre-training. Then, the pre-trained model can be fine-tuned in a supervised fashion using a small amount of labeled trained data to perform various supervised tasks. Pre-training machine learning models have already seen success in various domains including image processing and natural language processing (NLP) and BERT is one such successful algorithm.

Generally, language models read the input sequence in one direction: either left to right or right to left. This kind of one-directional training works well when the aim is to predict/generate the next word.

The motive behind using BERT is to have a deeper sense of language context because BERT uses bidirectional training. Sometimes, it's also referred to as "non-directional". So, it takes both the previous and next tokens into account simultaneously. BERT applies the bidirectional training of Transformer to language modeling, learns the text representations.

Bidirectional Transformer is at the center of BERT's design. This is significant because often, a word may change meaning as a sentence develops. Each word added builds the overall meaning of the word being focused on by the NLP algorithm. The more words that are present in total in each sentence or phrase, the more ambiguous the word in focus becomes. BERT accounts for the built up meaning by reading bidirectionally, accounting for the effect of all other words in a sentence on the focus word and eliminating the left-to-right momentum that biases words towards a certain meaning as a sentence progresses.

Note that BERT is just an encoder. It does not have a decoder. The encoder is responsible for reading text input and processing. The decoder is responsible for producing a prediction for the task.

BERT was pre-trained using only an unlabeled, plain text corpus (namely the entirety of the English Wikipedia, and the Brown Corpus). It continues to learn unsupervised from the unlabeled text and improve even as its being used in practical applications (ie Google search). Its pre-training serves as a base layer of "knowledge" to build from. From there, BERT can adapt to the ever-growing body of searchable content and queries and be fine-tuned to a user's specifications. This process is known as transfer learning.

As mentioned above, BERT is made possible by Google's research on Transformers. The transformer is the part of the model that gives BERT its increased capacity for understanding context and ambiguity in language. The transformer does this by processing any given word in relation to all other words in a sentence, rather than processing them one at a time. By looking at all surrounding words, the Transformer allows the BERT model to understand the full context of the word, and therefore better understand searcher intent.

2 Literature Survey

BERT was built upon recent work in pre-training contextual representations but crucially these models are all unidirectional or shallowly bidirectional. The following are some of the major pre-trained models of BERT which have been de-

veloped till now: BERT-Large, Uncased (Whole Word Masking): 24-layer, 1024-hidden, 16-heads, 340M parameters
BERT-Large, Cased (Whole Word Masking): 24-layer, 1024-hidden, 16-heads, 340M parameters
BERT-Base, Uncased: 12-layer, 768-hidden, 12-heads, 110M parameters
BERT-Large, Uncased: 24-layer, 1024-hidden, 16-heads, 340M parameters
BERT-Base, Cased: 12-layer, 768-hidden, 12-heads , 110M parameters
BERT-Large, Cased: 24-layer, 1024-hidden, 16-heads, 340M parameters
BERT-Base, Multilingual Cased (New, recommended): 104 languages, 12-layer, 768-hidden, 12-heads, 110M parameters
BERT-Base, Multilingual Uncased (Orig, not recommended) (Not recommended, use Multilingual Cased instead): 102 languages, 12-layer, 768-hidden, 12-heads, 110M parameters
BERT-Base, Chinese: Chinese Simplified and Traditional, 12-layer, 768-hidden, 12-heads, 110M parameters

GLUE SCORE[6] : The General Language Understanding Evaluation (GLUE) benchmark is a collection of resources for training, evaluating, and analyzing natural language understanding systems.

CoLA : The Corpus of Linguistic Acceptability.
SST-2 : Stanford Sentiment Treebank.
MRPC : Microsoft Research Paraphrase Corpus.
STS-B : Semantic textual similarity.
QQP: Quora Question Pairs.
MNLI-m : The Multi-Genre Natural Language Inference Corpus.
QNLI(v2): Question-answering Natural Language interface.
RTE: Recognizing Textual Entailment.

Model	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI(v2)	RTE
BERT-Tiny	64.2	0.0	83.2	81.1/71.1	74.3/73.6	62.2/83.4	70.2	70.3	81.5	57.2
BERT-Mini	65.8	0.0	85.9	81.1/71.8	75.4/73.3	66.4/86.2	74.8	74.3	84.1	57.9
BERT-Small	71.2	27.8	89.7	83.4/76.2	78.8/77.0	68.1/87.0	77.6	77.0	86.4	61.8
BERT-Medium	73.5	38.0	89.6	86.6/81.6	80.4/78.4	69.6/87.9	80.0	79.1	87.8	62.2
BERT base	78	55.8	83.7	84.1	86.3	90.5	91.1	90.9	87.7	68.6
DistilBERT	75.2	42.5	81.6	81.1	82.4	88.3	85.5	90.6	87.7	60

The Following is the description of the Data-sets used in the above GLUE-SCORE table.

- CoLA - is the sentence grammatical or ungrammatical.
- SST-2 - is the movie review positive, negative or neutral.
- MRPC - is the sentence B paraphrase of sentance A.

• STS-B - How similar are sentence A and sentence B.
• QQP - Are the two questions similar.
• MNLI-mm - Does sentence A entail or contradict sentence B.
• QNLI - Does sentence B contain the answer to the question in sentance A.
• RTE - Does sentence A entail sentence B.
• WNLI - Sentence B replaces sentence A ambigious.
pronoun with one of the nouns - is this the correct noun.

We use HuggingFace Algorithm for our analysis which is based on BERT-Base, Uncased.Hugging Face offers models based on Transformers for PyTorch and TensorFlow 2.0. There are thousands of pre-trained models to perform tasks such as text classification, extraction, question answering, and more. With its low compute costs, it is considered a low barrier entry for educators and practitioners.

3 DataSet

The Youtube video dataset contains almost 4000 videos with the following format for classification:
link-Video ID
title-Title of the video
description-Description of the video
category-Category of the video
The category/classes along with there statistics are: Travel Vlogs(1154), Food(901), Art and Music(947), History (593)

4 Methodology

We can use a pre-trained distilbert-base-uncased BERT model and then leverage transfer learning as a technique to solve specific NLP tasks. Transfer learning is key here because training BERT from scratch is very hard. The original BERT model was pre-trained with a combined text corpus containing about 3.3 billion words. The pre-training takes about 4 days to complete on 16 TPU chips, whereas most fine-tuning procedures from pre-trained models will take about one to few hours to run on a single GPU.
This process can be implemented with the following tasks:

4.1 Choose a pre-trained BERT model according to the language needs for our task.

For our task we choose the distilbert-base-uncased because it is pre-trained on the same data used to pre-train BERT using a technique known as knowledge distillation with the supervision of the bert-base-uncased version of BERT. DistilBERT is a transformers model, smaller and faster than BERT, which was pre-trained on the same corpus in a self-supervised fashion, using the BERT

base model as a teacher.This way, the model learns the same inner representation of the English language than its teacher model, while being faster for inference or downstream tasks. The model has 6 layers, 768 dimensions and 12 heads, totalizing 66M parameters. It can be trained 60% faster than the original uncased base BERT, which has 12 layers and approximately 110M parameters, while preserving 97% of the model performance.

	No of parameters (millions)	Inference Time (s)
BERT base	110	668
DistilBERT	66	410

In terms of inference time, DistilBERT is more than 60% faster and smaller than BERT.

4.2 Modify the pre-trained model architecture to fit our specific task.

BERT was designed to be pre-trained in an unsupervised way to perform two tasks: masked language modeling and next sentence prediction. In the masked language modeling, some percentage of the input tokens are masked at random and the model is trained to predict those masked tokens at the output. For the next sentence prediction task, the model is trained for a binary classification task by choosing pairs of sentences A and B. In our specific task, we need to modify the base BERT model to perform text classification. This can be done by feeding the first output token of the last transformer layer into a classifier of our choice. That first token at the output layer is an aggregate sequence representation of an entire sequence that is fed as input to the model. The package we use in our implementation already has several modified BERT models to perform different tasks, including one for text classification, so we don't need to plug a custom classifier.

4.3 Prepare the training data according to our specific task.

To work with BERT, we also need to prepare our data according to what the model architecture expects. For the text classification task, the input text needs to be prepared as following: Tokenize text sequences. In this specification, tokens can represent words, sub-words, or even single characters. For example, the word 'requisitions' is tokenized as ['re', 'qui', 'sit', 'ions']. Here, the two hash signs preceding some sub-words denote that a sub-word is part of a larger word and preceded by another sub-word.
Truncate and pad your sequences to the maximum sequence length suitable for your task, respecting the hard limit of 512 tokens per sequence according to the BERT specification.

Annotate your tokenized sequences with the special tokens '[CLS]' and '[SEP]' to mark the beginning and end of each sequence, respectively.
Convert your tokenized sequences into sequences of indices that are specific for the BERT vocabulary.
Create a sequence mask to indicate which elements in a sequence are tokens and which are paddings.
Create the numeric sequential array to be used for the positional embeddings, which is required by the transformer.

4.4 Fine-tune the modified pre-trained model by further training it using our own dataset.

After choosing and instantiating a pre-trained BERT model and preparing our data for model training and validation, we can finally perform the model fine-tuning. This is very similar to training a model from scratch, except usually for fine-tuning we have far less training data, less hyperparameters to tune.We fine-tuned the entire distilled BERT-based model for four epochs only and were able to obtain accuracy greater than 80% for all four categories.

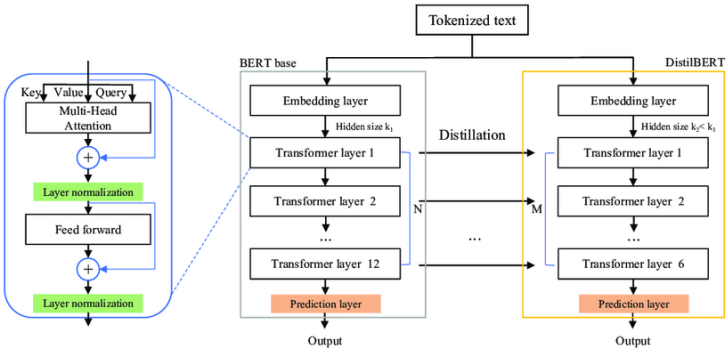

Fig. 1. Architecture of DistilBERT

5 Result Analysis

We split the data into 80% for training, 10% for validation, and 10% for testing. The following is the obtained confusion matrix for YouTube dataset:

Test Accuracy: 89.415%

	precision	recall	f1-score
art_music	0.90	0.96	0.93
food	0.94	0.83	0.88
history	0.98	0.81	0.88
travel	0.82	0.93	0.87
accuracy			0.89
Macro avg	0.91	0.88	0.89
Weighted avg	0.90	0.89	0.89

References

1. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/arXiv.1810.04805
2. Yu, Shanshan Jindian, Su Luo, Da. (2019). Improving BERT-Based Text Classification With Auxiliary Sentence and Domain Knowledge. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2953990.
3. Sumanth Prabhu, Moosa Mohamed, Hemant Misra Multi-class Text Classification using BERT-based Active Learning 10.48550/arXiv.2104.14289
4. Qing Yu et al 2021 Research on Text Classification Based on BERTBiGRU Model 10.1088/1742-6596/1746/1/012019
5. Yongjun Hu, Jia Ding, Zixin Dou, Huiyou Chang, "Short-Text Classification Detector: A Bert-Based Mental Approach", Computational Intelligence and Neuroscience, vol. 2022, Article ID 8660828, 11 pages, 2022. https://doi.org/10.1155/2022/8660828
6. GLUE SCORE, https://mccormickml.com/2019/11/05/GLUE/
7. Distil-BERT, https://www.researchgate.net/figure/The-DistilBERT-model-architecture-and-components$_fig2_358239462$