IE643: Deep Learning : Theory and Practice **Jul-Nov 2019**

BERT: Bidirectional Encoder Representations from Transformers

Author: Devlin et al. *Scribes: Vamsi Krishna Reddy Satti [160050064]*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

Note: *Please be aware that some writing here may unintentionally overlap with the contents of my project report and hence that of my team members, though it was not referred to during writing of this work.*

1 Introduction

This report is a summary for the paper titled 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding' by Devlin *et al* [2]. The paper introduces a new language model which is pre-trained using deep bidirectional representations from unlabeled text by jointly conditioning over both left and right contexts simultaneously in all layers of a transformer model. Thus, this pre-trained BERT model can be fine-tuned for a specific task without major architectural changes. One can say that it has enabled effective ways of transfer learning in the field of natural language processing (NLP). BERT achieves this deeply bidirectional training by proposing novels techniques called Masked Language Modelling (MLM) and Next Sentence Prediction (NSP).

2 Problem Description

One of the main difficulty in NLP is the shortage of human-labeled training data. Modern deep learning-based NLP models are known to see huge improvements with larger amounts of data. To mitigate this shortage of annoted training examples, researchers use a variety of methods for training general purpose language models using large unannotated text on the web. The is commonly known as *pre-training.* The pre-trained model can then be *fine-tuned* on small-data NLP tasts like question answering and sentiment analysis. The paper the proposes an effective pre-training method from unlabled text by jointly conditioning on both the left and right context in all layers of a transformer architecture.

3 Background

BERT builds on top of a number of clever ideas that had been coming up then in the NLP community. ELMo by Peters *et al.* [4] looks at the entire sentence before assigning each word in it an embedding instead of using a fixed embedding. It uses a bi-directional LSTM trained on a specific task to be able to create those embeddings. ULMFiT by Howard *et al.* [3] introduced methods to effectively utilize a lot of what the model learns during pre-training – more than just embeddings, and more than contextualized embeddings. ULM-FiT introduced a language model and a process to effectively fine-tune that language model for various tasks. BERT proposes its pre-training methods on a transformer architecture as introduced by Vaswani *et al.* [7].

4 Methodology

BERT specifically involves an effective pre-training method that has been applied to transformers. During pre-training, the model is trained over an unlabeled corpus like Wikipedia over a proposed set of pre-training tasks. This model is then fine-tuned during training possibly with by appending an additional smaller auxiliary model. BERT achieves this bi-directional training by proposing a novel technique called Masked Language Modelling (MLM). BERT also builds a Next Sentence Prediction (NSP) Model using Transformers. Both MLM and NSP are trained jointly and their combined loss function is minimised.

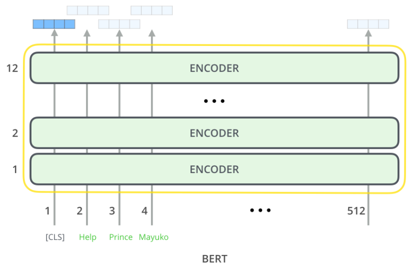

Figure 1: Input and output in BERT. [Source [1]]

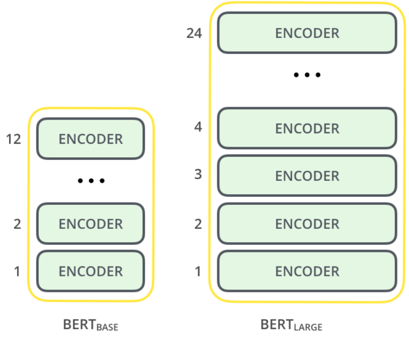

Figure 2: BERT-BASE and BERT-LARGE. [Source [1]]

The authors train BERT on two different model sizes:

- BERT BASE: A smaller model comparable in size to the OpenAI Transformer [5] for fair comparison.
- BERT LARGE: A huge model to achieve state of the art results as reported in the paper.

BERT is essentially a trained Transformer Encoder stack with a number of encoder layers (referred to as Transformer Blocks in the paper) depending on BASE/LARGE version and much larger feed-forward layers, more attention heads than those in Vaswani *et al.* [7].

4.1 Input Representation

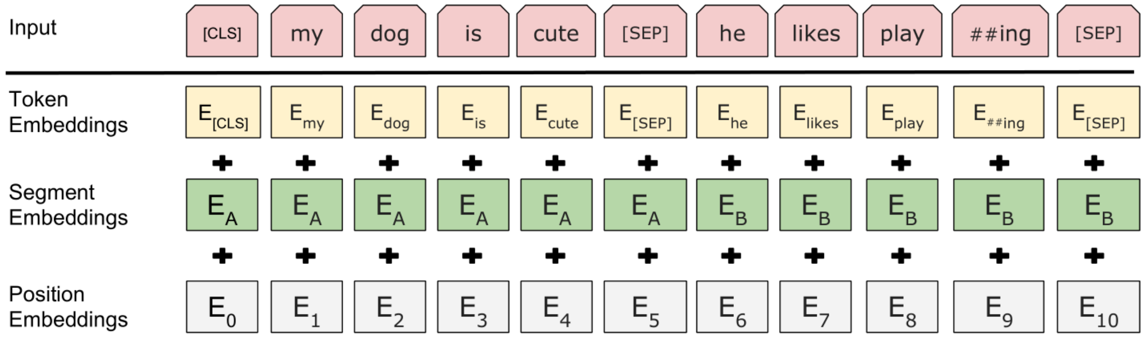

Figure 3: Input representation in BERT. [Source [2]]

The input representation of a token is constructed by using token embeddings along with positional embeddings. This is visualised in Figure 3.

The first token of every sequence is always the token ([CLS]) where CLS stands for classification. The final hidden state from the last layer of transformer corresponding to this token is used as the aggregate representation for classification tasks.

Also sentence pairs are packed together into a single sequence and distinguished using a [SEP] token. A learned embedding called segment embedding is added with denotes which sentence that token belongs to. Position embedding is added similar to the transformer paper [7] except that here this embedding is sinusoidal there but in BERT, it also learned like segment and token embeddings.

4.2 Pre-training methods

In this section, we present the most important contributions of the paper. The two tasks proposed to train BERT as presented in the following sections.

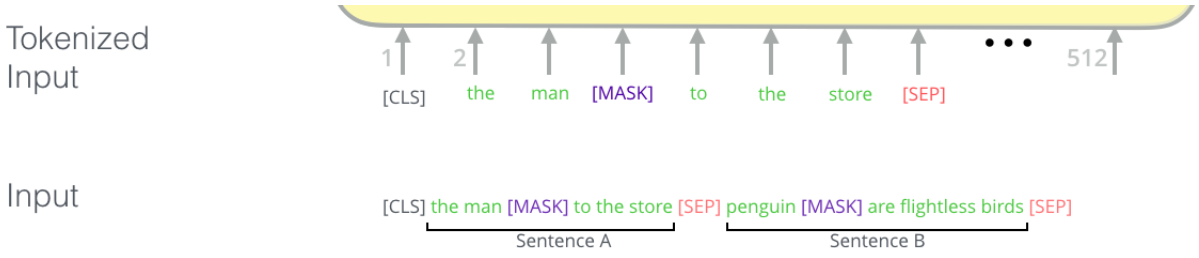
Tokenized Input
1
2
512
[CLS]
the
man
[MASK]
to
the
store
[SEP]
Input
[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]
Sentence A
Sentence B

Figure 4: MLM and NSP in BERT. [Source [1]]

4.2.1 Masked Language Model

Sequential language models predict the next word given previous 'n' words. Instead of this, BERT randomly masks words in the input sentence and tries to predict them given other words. This model is effective because it forces the model to learn to use the words from entire sequence to deduce the missing word.

The paper suggests to randomly replace 15% of tokens in the input with [MASK] and feed them to the transformer network. The problem with this approach is that the model tries to predict when [MASK] is present in the input. But these [MASK] tokens are present during fine tuning. To mitigate this, the authors suggest to replace 10% of the masked tokens with any random word.

4.2.2 Next Sentence Prediction

Some tasks in NLP like question answering, natural language inference etc. may require an understanding of the relationship between two sentences. So in addition to MLM, BERT also uses a NSP task to pretrain the model for these type of tasks. Two sentences are fed to the BERT with a special [SEP] seperating them. 50% of them first sentence is followed by original second sentence and for the remaining times a random sentence will be given. The task here is to predict whether the next sentence is random or not.

4.3 Feature extraction from BERT

In addition to fine-tuning the model for a specific task, pre-trained BERT can also be used as a feature extractor, specifically in tasks where a task-specific model architecture is unavoidable. Various ablation studies ahve been presented in the paper to emperically determine the best strategy for each task.

4.4 Assumptions

A limitation of BERT is the pre-train fine-tune discrepancy. The pretrained bert model gets accustomed to [MASK] tokens while these are actually absent during Fine-tuning. Even though BERT tries to demask 10% tokens, the discrepancy still persists. Another limitation of BERT is independence assumption between masked tokens. While predicting masked tokens BERT assumes that [MASK] tokens are independent given input context. But this may not be true. For example in the sentence "[NEW] [DELHI] is a city" the masked tokens [NEW] and [DELHI] are not independent.

5 Results

To evaluate performance, BERT was compared to other state-of-the-art NLP systems. For a fair comparision, BERT-BASE was used in order to have similar model capacity. Importantly, BERT achieved all of its results with almost no task-specific changes to the neural network architecture. On SQuAD v1.1 [6], BERT achieves 93.2% F1 score (a measure of accuracy), surpassing the previous state-of-the-art score of 91.6% and human-level score of 91.2%. BERT also improves the state-of-the-art by 7.6% absolute on the very challenging GLUE benchmark [8], a set of 9 diverse Natural Language Understanding (NLU) tasks. Infact, in total BERT fine-tuned models had obtained new state-of-the-art results then on eleven natural language processing tasks.

6 Project connection with the paper

Our work was to present methods to do adversarial attack on BERT fine-tuned models. This has important implications since BERT has become a state-of-the-art baseline for many other models in the recent literature. Attacking BERT is one step further to prevent misuse of modern tools and make them robust to security attacks. Unlike vision, adversarial attacks in NLP and especially on BERT has been limitedly explored. My project demonstrated some significant performance improvements to the latest work within a speed - success rate trade off.

7 Conclusion

The paper has presented an effective way to learn deep *bidirectional* dependencies in the language, thus leveraging the effectiveness of *pre-training* in NLP even further. This has enabled many researches and practioners to simply *fine-tune* a BERT model for their specific tasks, thus opening doors for more possibilites and improvement in performances with less computational requirements. BERT essentially gives us an effective way of transfer learning in NLP.

References

[1] Jay Alammar. The Illustrated BERT. https://jalammar.github.io/illustrated-bert/, 2018. [Online; accessed 06-October-2019].

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[3] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018.

[4] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

[5] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

[6] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[8] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2018.