Ka Ming Chan
NetID: kmchan2

BERT Technology Review

This technology review introduces a language representation model named Bidirectional Encoder Representations from Transformers, abbreviated as BERT. It is a natural language processing (NLP) pre-training technique. A pre-training technique is training a model on unannotated text data in order to form parameters for other tasks. BERT was developed by researchers at Google AI Language, and was impactful for its results in helping with Question Answering (SQuAD v1.1), Natural Language Inference (MNLI), and others. The key innovation of BERT is its Transformer capability to train text data bidirectionally - aside from traditional ways of reading text left-to-right or right-to left. Words will be jointly conditioned on both left and right context in all layers. BERT provides a deep language context, when it is pre-trained on unsupervised language. For instance, it can give different contextualized embeddings to the same exact word that is used in two situations by determining from the sentence and story itself. It is unique from traditional NLP methods and models. After pretraining, BERT can be finetuned on smaller datasets to optimize its performance on specific tasks.

The process of the BERT framework has two steps: pre-training and fine-tuning. The model is first trained on unlabeled data over different pre-training tasks, and then proceeds to separate downstream tasks with different fine-tuned models, using the same pre-trained parameters. In the pre-training stage, it has two tasks: Masked LM (MLM) and Next Sentence Prediction (NSP). Starting with Masked LM, BERT is attempting to allow the words in the text data to be understood by both left and right words surrounding them, in other words bidirectional. In order to train a deep bidirectional representation, some percentage of the input tokens are masked at random, and then those masked tokens are predicted. The paper would hide 15% of the word data with [MASK] tokens. These tokens are predicted based on the context of the non-masked words in the sequences. The output prediction requires the following steps: adding a classification layer on top of the encoder output, multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension, and calculating the probability of each word in the vocabulary with softmax. This method of predicting masked words allows a bidirectional pre-trained model to be obtained. Though, there will be a mismatch between the pre-training and fine-tuning because the masked words do not appear in the fine-tuning. There, 15% of the predicting token positions are random. There will be a probability that the token shall be [MASK] or not. This model converges slower than directional models.

The second task of the pre-training stage is the Next Sentence Prediction (NSP). Many important downstream tasks such as Question Answering (QA) and Natural Language Inference (NLI) are based on understanding the relationship between two sentences, which is not directly captured by language modeling. In order to train a model that understands sentence relationships, a binarized next sentence prediction task that is pre-trained for can be trivially generated from any monolingual corpus. The model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document. In the inputs of pairs, 50% of the second sentences are the subsequent sentences in the original document, and the other 50% of the second sentences are random sentences from the corpus. The random sentences are assumed to be disconnected from the first. To distinguish between the two sentences, a [CLS] token is inserted at the beginning of the first sentence, and a [SEP] token is inserted at the end of each sentence. There will be a sentence embedding and a positional embedding added to each token. After that is done, the second sentence is predicted whether or not it is connected to the first. The entire input sequence goes through the Transformer model, the output of the [CLS] token is transformed into a 2×1 shaped vector using a simple classification layer, then the probability of IsNextSequence is calculated with softmax. This task is beneficial towards Question Answering and Natural Language Inference.

After the described processes of pre-training, BERT is fine-tuned with a wide variety of language tasks. It is straightforward since the Transformer allows BERT to model many downstream tasks. BERT can accept task-specific inputs and outputs and fine-tune all parameters end-to-end. For examples of the possible language tasks, classification tasks such as sentiment analysis can be done by adding a classification layer on top of the Transformer output for the [CLS] token. Question Answering tasks can be trained by learning two extra vectors that mark the beginning and the end of the answer. Using BERT, Named Entity Recognition tasks can be trained by feeding the output vector of each token into a classification layer that predicts the NER label. Most hyper-parameters stay the same as in BERT training.

BERT receives many meaningful scores proving itself: GLUE score of 80.5%, MultiNLI accuracy of 86.7%, SQuAD v1.1 question answering Test F1 of 93.2, and SQuAD v2.0 Test F1 of 83.1. BERT is powerful in that it accounts for ambiguous meanings of words by reading bidirectionally, accounting for the effect of all other words in a sentence. It is innovative as the left-to-right or right-to-left momentum biases the words. BERT has been error-prone with Google's NLP techniques to date, and has the potential to drastically improve artificial systems across the board.

References

https://arxiv.org/abs/1810.04805

https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model

https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html