CS 410 Text Information System Tech Review

An Introduction to BERT

Zhenchen Yu (zy23)

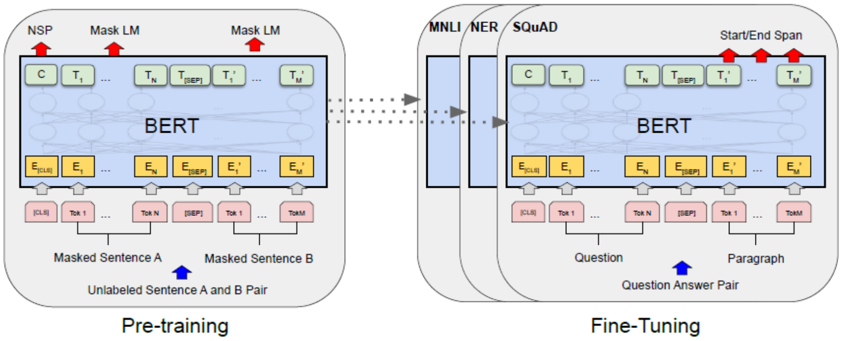
October 26th, 2020

Introduction

BERT (Bidirectional Encoder Representations from Transformers) is a deep learning language model that was created and introduced by researchers from Google in 2018. When released, it caused an earthquake in the NLP research areas because it beat eleven NLP tasks including GLUE (General Language Understanding Evaluation), SQuAD (Stanford Question Answering Dataset), SWAG (Situations With Adversarial Generations), and others.[1]

BERT is not out of nowhere. It absorbs the advantage of multiple previous state-of-the-art models, such as ELMo, OpenAI GPT and Transformer etc. BERT is powerful in that unlike the previous language models that are trained either from left to right or from right to left, BERT is trained bidirectional, which enables the model to learn better about the relationship between the words. In addition, BERT uses Transformer instead of LSTM or RNN, which are much slower, to train the large dataset that includes the 800M words BooksCorpus and 2,500M words English Wikipedia. BERT is also flexible in dealing with all kinds of tasks, thanks to its fine-tuning component.

There are two major components in BERT: Pre-training and Fine-Tuning. BERT uses feature-based approach for pre-training and use self-attention mechanism during the fine-tuning.



Word Embedding

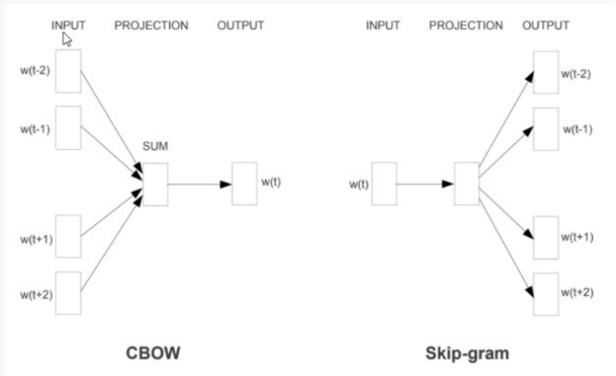
To understand BERT, we need to first understand what is "word embedding". In short, word embedding is a from text to text where words that have the same meaning have a similar representation. For example, we have a list of words, [woman, girl, man, boy]. We can map those four words into a two-dimensional vector space [gender, age]. Hence, the procedure of the mapping is word embedding. Of course, it is not an easy task when we have millions of words to map into a lower dimensional space. One solution is to use autoencoder network to complete the mapping. From Wikipedia, "An autoencoder is a type of artificial neural network used to learn efficient data coding in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal "noise".[2]

[1] https://en.wikipedia.org/wiki/BERT_(language_model)

[2] https://en.wikipedia.org/wiki/Autoencoder

Word2Vec

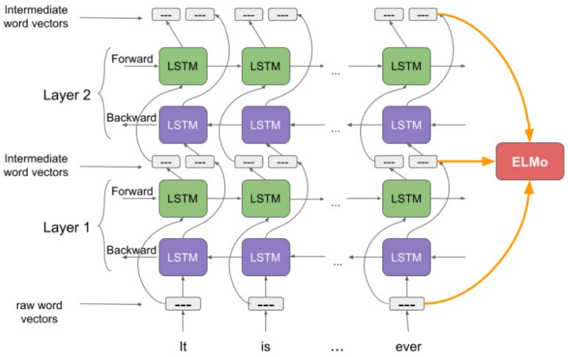
One of the most popular word embedding method is called “Word2Vec”. Natural Language Processing is difficult because the data is unstructured, ambiguous and unlabeled. Hence the traditional supervised learning techniques cannot be used on NLP. Word2Vec word embedding solves this problem by leveraging the context to predict the current word. Under this setting, each word is labelled, and the variables are all the words before and after the word to predict. There are two different learning models under word embedding, Continuous Bag-of-Words (CBOW) and Continuous Skip-Gram Model. CBOW uses context to predict the current word while continuous skip-gram uses the current word to predict the surrounding words.



Word2Vec Training Models
Taken from "Efficient Estimation of Word Representations in Vector Space", 2013

ELMo

ELMo is another word embedding model. However, it is way more effective than traditional word embedding in that it extracts context-sensitive features from a left-to-right and a right-to-left language model.[3] The graph below demonstrates the structure of the ELMo model. Basically, it uses two layers of LSTM to train the text forward and backward.



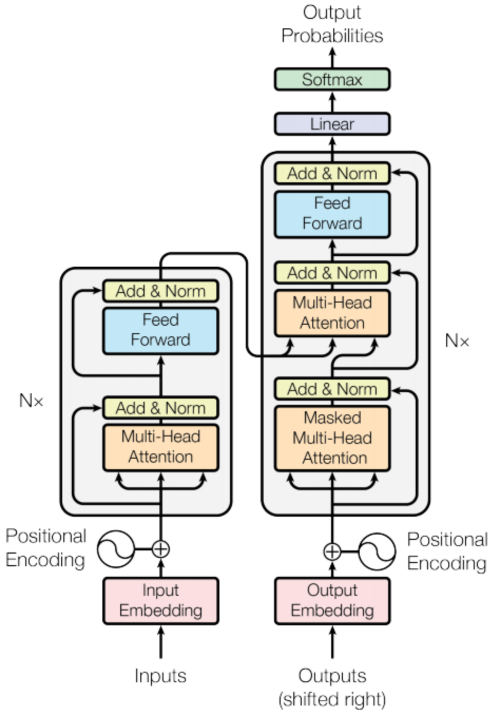
To some extent, BERT’s network structure is quite similar to ELMo since they are both bidirectional, but BERT has a deeper neural network, and BERT uses Transformer instead of LSTM in the training which has a number of advantages.

[3] Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (11 October 2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding".

Fine-tuning Approach and Transformer

Fine-tuning is a way to leverage the pre-trained model to complete a new task. The advantage of this approach is that it doesn't require a lot of parameters to be trained from the start. One of the state-of-the-art models that used this approach is OpenAI GPT.

Although fine-tuning approach is relatively inexpensive compare with the pre-training, it is still quite slow if we use the traditional RNN or LSTM. Most of the latest models are using a framework called Transformer which was also developed by Google in 2017. Like RNN, Transformers are designed to deal with sequential data, but unlike RNN, Transformers do not require that the sequential data be processed in order. For example, if we use RNN to process a sentence, we must feed the model word by word, and each word's hidden state is dependent on the previous word's hidden state. However, Transformers can process the whole sentence simultaneously. Therefore, Transformer can train the data much faster than RNN thanks to its parallelization.



Model structure of the Transformer[4]

There are two components in a Transformer, Encoder and Decoder. The input embeddings, together with the positional encoding, are sent to a Multi-Head Attention layer, which deals with the question - which part of input should we focus? It calculates the relevance between the words in the same sentence. Then a Feed Forward layer is used to digest the Attention vectors and transfer them into the next block. The decoder block has a similar structure. It first passes the output embedding and positional encoding into a Multi-Head Attention layer, which is masked for learning purposes, and then it goes into another Multi-Head Attention layer, which gathers both encoder and decoder attention. Finally, it goes to a Feed Forward layer and then generate the output probabilities. BERT uses this self-attention mechanism to supervise the learning.

[4] Polosukhin, Illia; Kaiser, Lukasz; Gomez, Aidan N.; Jones, Llion; Uszkoreit, Jakob; Parmar, Niki; Shazeer, Noam; Vaswani, Ashish (2017-06-12). "Attention Is All You Need".

Why Is BERT Different

One key innovation in BERT is that it used a novel technique called Masked LM (MLM) which allows bidirectional training in models. Inspired by cloze task, researchers randomly mask some percentage of the input tokens in order to enable bidirectional training. Another task in BERT pre-training is Next Sentence Prediction (NSP), which is used to teach the model the relationship between two sentences. These features are important in tasks such as Question Answering (QA) and Natural Language Inference (NLI).

How to Use BERT

With 110M parameters in the BERT $_{\text{base}}$ and 340M parameters in the BERT $_{\text{large}}$, it is extremely difficult for us to train the BERT from scratch. Fortunately, we don't need to do that. To use BERT, we just need to fine-tune the model for our own specific tasks.

The best way to try out BERT is through the BERT Fine-Tuning with Cloud TPUs notebook hosted on Google Colab.[5] You can also find the BERT model on the TensorFlow hub: https://www.tensorflow.org/hub.

BERT is open sourced so you can look at the code at: https://github.com/google-research/bert

Conclusion

BERT is undoubtedly a breakthrough in the Natural Language Processing area. It is powerful, but more importantly it is approachable to users due to its fine-tuning features, which makes it possible to apply BERT in many practical areas.

For those who wish to learn more details about the model, I highly recommend reading the full paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" and ancillary articles referenced in it.

[5] https://colab.research.google.com/github/tensorflow/tpu/blob/master/tools/colab/bert_finetuning_with_cloud_tpus.ipynb