Ali Saad
CS410 Tech Review
11/5/2022

BERT Overview

What if I told you that when you type a question into a search engine, what you see isn't what the system sees? As well as every system may be different in what language it reads that context in. And all these different languages have their own upsides and downsides but overall it improves many natural language processing tasks in order to ultimately find what you're looking for. Today I will be giving an example of one of those language representation models known as BERT, which also stands for Bidirectional Encoder Representations from Transformers. Not only will we be going over what the system is but deep understanding of how it works.

Before we go to deep into it, let me give a little background on why BERT isn't like any other language representation models "Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain 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 finetuned 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." from the article Devlin, Jacob, et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 24 May 2019, arxiv.org/pdf/1810.04805.pdf. This makes BERT a strong language representation model and it is also shown to be conceptually simple and powerful through not just theory or pure logic but by being used numerous times and outputting its dominance. By being able to jointly condition on both the left and right context this allows for a BERT to be able to use the whole sequence rather than just the previous few words. How does BERT do this once might ask? Lets go deeper into what mechanisms BERT uses in order to take a whole sequence at once.

At a high level there are 2 types of mechanisms BERT uses in order to process everything in one sequence then a few words which are known as attention and self-attention mechanisms. In a nutshell the attention mechanism means that instead of going through the sentence sequentially, the entire sequence is used to do the decoding on the currently handled word while using an attention system to give weights to decide which word in the input gets how much say in how the current word is handled. While the Self-Attention mechanism means that even for the encoding of the input sequence itself the context is already used. For example, if you have a sentence with an "it" that is used as a pronoun, the encoding of that token is going to be strongly context dependent.
Self-Attention means similarly to attention there is a weighting function for which other input tokens are relevant for the encoding of the current input tokens. Now from here it goes even deeper where we discuss pre-training and fine-tuning which is what this paper is going to be the majority about, especially how we can fine tune it even more.

Let's start by explaining pre-training and fine-tuning, pre-training is shown to be effective by using natural language inference and paraphrasing which aim to predict the relationships between sentences by analyzing them holistically, as well as token-level tasks such as named entity recognition and question answering, where models are required to produce fine-grained output at the token level…BERT uses masked language models to enable pre trained deep bidirectional representations. This is also in contrast to Peters et al. (2018a), which uses a shallow concatenation of independently trained left-to-right and right-to-left LMs." Devlin, Jacob, et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 24 May 2019, arxiv.org/pdf/1810.04805.pdf. The reason why this is so effective is because now there is no need to put as much strain on the system as less heavily-engineered tasks are needed. BERT is the first language to do it at a sentence-level or sequence level. Instead of doing it just a few words at a time, it's able to capture everything at a token-level as well. Since current techniques restrict the power of pre-trained representations especially for the fine-tuning approach. Now involving fine-tuning this is where BERT comes into play as it improves the fine-tuning based approaches.

A key differentiator between the two to keep in my mind is that unlike pre-training, fine-tuning is relatively inexpensive. What authors Jacob Devlin et al help propose is that the fine-tuning for BERT is "straightforward" by adding one additional layer after the final BERT layer and training the entire network for just a few epochs. They were able to demonstrate this by using standard NLP benchmark problems GLUE,SQuAD, and SWAG, which examine different aspects of nature language inference after fine-tuning 2-3 epochs with the ADAM optimizer. With this approach they received learning rates between 1e-5 to 5e-5, which is now a commonly adopted method after viewing their results.

Now with all that being said, are you not convinced that BERT is one of the most powerful language representation models? With BERT " In particular, these results enable even low-resource tasks to benefit from deep unidirectional architectures" according to Devlin, Jacob et al. Adding another layer to BERT is a major contribution not to take away anything from the pre-trained model but help improve it on the fine-tuning side which ultimately will only help us advance in successfully tackling a broad set of NLP tasks. Next time you search something in a search engine, know that BERT may be one of the languages used to help computers understand the meaning of the ambiguous context into text by using surrounding text to establish context to help identify what you are looking for. I hope this paper allows one to get a deeper understanding of the NLP language representation model BERT and show how Jacob Devlin et al helped introduce a more effective fine-tuning method.