Machine learning (ML) models in natural language processing (NLP) have been frequent in research over the past decade. The introduction of many models such as recursive neural networks (RNN) as well as improvements, such as bidirectional recursive neural networks (BRNN) and long-short term memory (LSTM) architecture, have made ML a hot topic in improving NLP models. In 2018, Bidirectional Encoder Representations from Transformers (BERT) was introduced, a state of the art ML model which introduced the use of a transformer, consisting of an encoder and decoder that is able to transform tokens (words) into machine-understandable vectors. Using the transformer, BERT is able to convert text into its own language model. As well, BERT, compared to most RNN models, uses the transformer on the entire text at once rather than sequentially reading in tokens. Finally, the core model of BERT has already been pre-trained by Google, requiring minimal code to have BERT serve its intended purpose.

BERT consists of 2 unique training approaches, masked language modeling (MLM) and next sentence prediction (NSP). The use for MLM is for context recognition. Given BERT a sentence with missing words (masked tokens), BERT should output a complete sentence filling in those missing words. While this is a more naive method for context recognition (Humans will understand relations between words/nouns to fill in the gap while BERT will only understand the linguistic pattern of a sentence), MLM is still a powerful tool. NSP is similar to MLM; given two sentences, BERT should classify whether one logically follows the other.

In total, BERT utilizes both of these training strategies, hoping to minimize the combined loss function. However, this is only the core of the BERT model. Since BERT has two possible outputs, text (MLM) as well as a vector (NSP), BERT is able to be used as a trainable function. In practice, by adding a new layer to BERT, we are able to convert BERT to output different representations of the language model. For example, we can append a classification layer to BERT in order to convert NSP into a classifier. This makes BERT a powerful tool in all sorts of applications.

Since BERT is a very generalized model, I would like to put BERT up to the test to a more specific task against a specific model (TBD). I will try to ensure that BERT is able to perform such a task as well as use a model that has been conditioned for the task. Using new techniques such as word2vec and pos-tagging, I would like to see if BERT can outperform or perform just as well as another model (seemingly more suited for a task). This should prove that BERT is a successful generic model.