BERT

The following are the primary difficulties when building deep learning and natural language processing (NLP) classification models.

- Data collection (labelled data) and deep training

While there is a huge amount of text-based data available, very little of it has been labelled to use for training a machine learning model. Data collection is burdensome, time-consuming, expensive, and is the number one limiting factor for successful NLP projects.

- Another key limitation was that these models did not take the context of the word into account

Past language models like Word2Vec, Glove2Vec built context-free word embeddings

To overcome the problem of data collection and deep training, the field of computer vision has long made use of transfer learning. Transfer learning is the ability to use a model trained for a different but similar task to accelerate your solution on a new one. It takes far less effort to retrain a model that can already categorize trees than it does to train a new model to recognize bushes from scratch

Transfer Learning in NLP = Pre-Training and Fine-Tuning

To overcome the problem of understanding the text we needed a word embedding model to understand the context of the word. BERT is a language model which provides context.

In Comes BERT, which overcomes the above mentioned problems. It needs less data, less training time and provides state of the art results on 11 different natural language processing tasks. These natural language processing tasks include, amongst others, sentiment analysis, named entity determination, next sentence prediction, semantic role labeling, text classification and coreference resolution

WHAT IS BERT?

BERT (Bidirectional Encoder Representations from Transformers) is a new model by researchers at Google AI Language, which was introduced and open-sourced in late 2018 and has since caused a stir in the NLP community.

BERT is an acronym for Bidirectional Encoder Representations from Transformers.

Firstly, BERT is based on the Transformer architecture which are designed to produce sentence encodings.

Secondly, BERT is pre-trained on a large corpus of unlabelled text including the entire Wikipedia (that's 2,500 million words!) and Book Corpus (800 million words).
This pre-training step is half the magic behind BERT's success. This is because as we train a model on a large text corpus, our model starts to pick up the deeper and intimate understandings of how the language works. This knowledge is the swiss army knife that is useful for almost any NLP task.

And finally, the most impressive aspect of BERT is that we can fine-tune it by adding just a couple of additional output layers to create state-of-the-art models for a variety of NLP tasks.

BERT takes an entirely different approach to learning. Basically BERT is given billions of sentences at training time. It's then asked to predict a random selection of missing words from these sentences. After practicing with this corpus of text several times over, BERT adopts a pretty good understanding of how a sentence fits together grammatically. It's also better at predicting ideas that are likely to show up together

BERT stands for Bidirectional Encoder Representations from Transformers. It is designed to pre-train deep bidirectional representations from unlabelled text by jointly conditioning on both left and right context. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP tasks

BERT is pre-trained on two different NLP tasks i.e. Masked Language Modeling and Next Sentence Prediction. The objective of Masked Language Model (MLM) training is to hide a word in a sentence and then have the program predict what word has been hidden (masked) based on the hidden word's context. The objective of Next Sentence Prediction training is to have the program predict whether two given sentences have a logical, sequential connection or whether their relationship is simply random.

BERT is a big neural network architecture with a huge number of parameters that can range from 100 million to over 300 million. It is always better to use a pre-trained BERT model that was trained on a huge dataset. We can then further train the model on our relatively smaller dataset and this process is known as model fine-tuning.

BERT helps understand natural language better. The purpose of BERT is to help google search better interpret what its users are asking.

BERT is open source, and all of this encoded information is available when you deploy it.

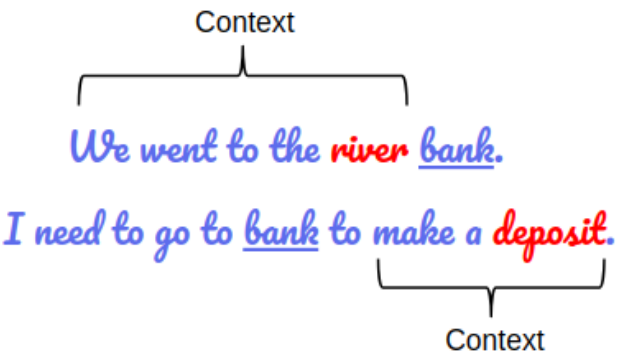
BERT is actually so advanced that it can to an extent use the data it gathers from one language and apply it to any number of others. So even as it continues learning from these 70+ languages, English searches will improve as well.

Bert's key innovation is applying bidirectional training of Transformer, a popular attention model to language modelling.

Transformer is an attention mechanism that learns contextual relations between words in a text. In its vanilla form, Transformer includes two separate mechanisms - an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary

Transformer is the part of the BERT model that gives it increased capacity for understanding context and ambiguity in language. The transformer does this by processing any given word in relation to all other words in a sentence rather than processing them one at a time. By looking at all surrounding words, the transformer allows the bert model to understand the full context of the word and therefore better understand search intent. Transformer is considered a significant improvement because it doesn't require sequences of data to be processed in any fixed order like RNNs and CNNs do.

As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the transformer encoder reads it bidirectionally (at the previous words and at the next words), meaning the entire sequence of words at once. The results demonstrated by the BERT model show that a bidirectionally trained language model can have a deeper sense of language context and flow than single-direction language models.



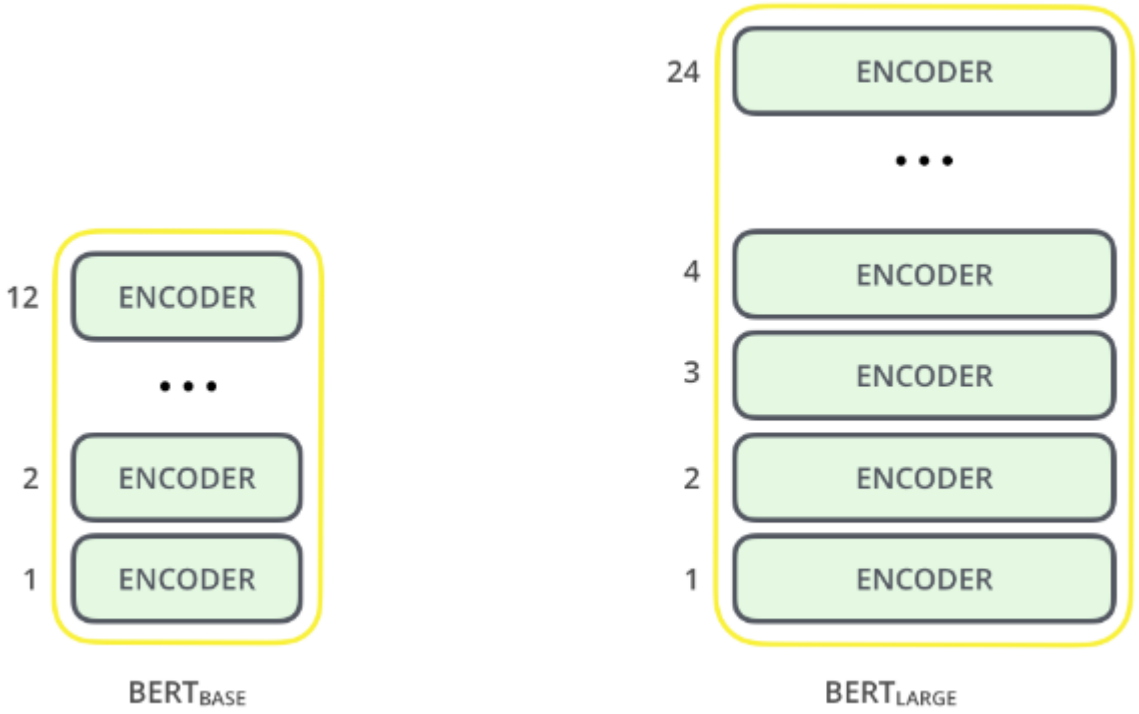
If we try to predict the nature of the word "bank" by only taking either the left or the right context, then we will be making an error in at least one of the two given examples. One way to deal with this is to consider both the left and the right context before making a prediction. That's exactly what BERT does.

ARCHITECTURE

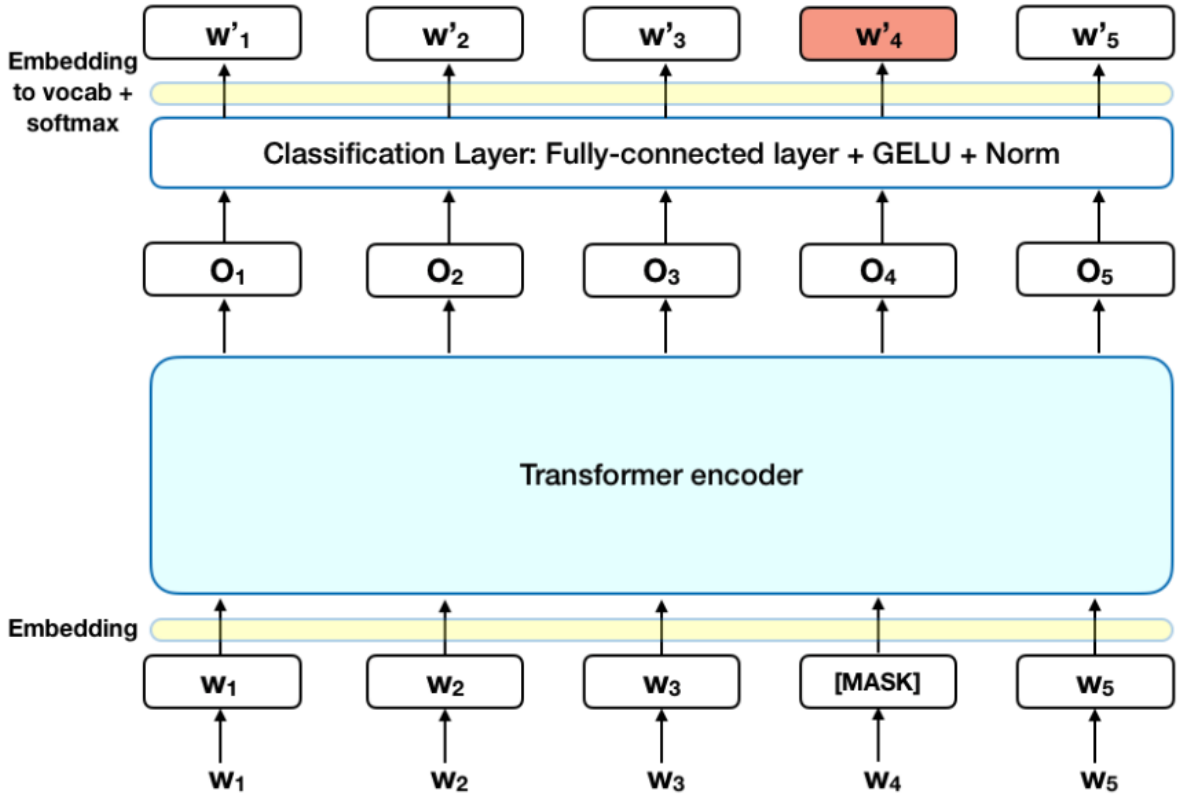
The BERT architecture builds on top of Transformer. We currently have two variants available

BERT Base: 12 layers (transformer blocks), 12 attention heads, and 110 million parameters

BERT Large: 24 layers (transformer blocks), 16 attention heads and, 340 million parameters



How Bert works?



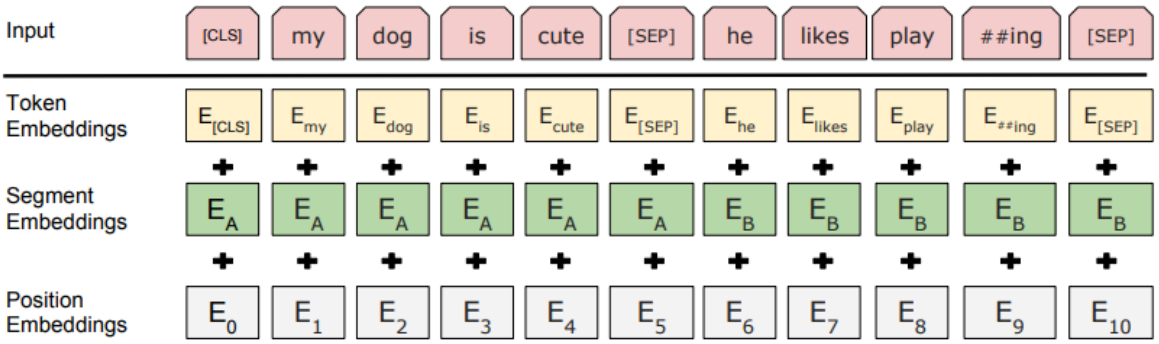
Steps

- Reading data

BERT first reads all the words at once rather than left-to-right or right-to-left

- Vectorization

In traditional NLP, the starting point for model training is word vectors. Word vectors are a list of numbers [0.55, 0.24, 0.90, …] that attempt to numerically represent what that word means. With a numeric representation, we can use those words in training complex models, and with large word vectors, we can embed information about words into our models. BERT does something similar (in fact, its starting point is word vectors), but it creates a numeric representation of an entire input sentence (or sentences). Word vectors are very shallow representations that limit the complexity that they can model but BERT does not have this limitation.



The developers behind BERT have added a specific set of rules to represent the input text for the model. Many of these are creative design choices that make the model even better. For starters, every input embedding is a combination of 3 embeddings:

Position Embeddings - BERT learns and uses positional embeddings to express the position of words in a sentence. These are added to overcome the limitation of Transformer which, unlike an RNN, is not able to capture “sequence” or “order” information

Segment/Sentence Embeddings - BERT can also take sentence pairs as inputs for tasks (Question-Answering). That’s why it learns a unique embedding for the first and the second sentences to help the model distinguish between them. A sentence embedding indicating Sentence A or Sentence B is added to each token. Sentence embeddings are similar in concept to token embeddings with a vocabulary of 2. In the above example, all the tokens marked as EA belong to sentence A (and similarly for EB).

Token Embeddings - A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence.

For a given token, its input representation is constructed by summing the corresponding token, segment, and position embeddings. Such a comprehensive embedding scheme contains a lot of useful information for the model. These combinations of preprocessing steps make BERT so versatile. This implies that without making any major change in the model’s architecture, we can easily train it on multiple kinds of NLP tasks.

- Training

BERT is pre-trained on two NLP tasks - Masked Language Modeling and Next Sentence Prediction

Under Masked Language Modelling step, we are trying to build a model to predict a missing word from within the sequence. Under Next Sentence Prediction, we are trying to build a model to predict Given two sentences A and B, is B the actual next sentence that comes after A in the corpus or just a random sentence?

For the first step in training process, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. The BERT loss function takes into consideration only the prediction of the masked values and ignores the prediction of the non-masked words. In practice, the BERT implementation is slightly more elaborate and doesn't replace all of the 15% masked words

The model is fine-tuned on next-sentence-prediction. In this step, the model tries to determine if a given sentence is the next sentence in the text. For 50% of the sentence pairs, the second sentence would actually be the next sentence to the first sentence. For the remaining 50% of the pairs, the second sentence would be a random sentence from the corpus. The labels for the first case would be 'IsNext' and 'NotNext' for the second case

When training the BERT model, Masked LM and Next Sentence Prediction are trained together with the goal of minimizing the combined loss function of the two strategies. Convergence is slow and BERT takes a long time to train. However, it learns the contextual relationships in text far better.

Most businesses can make use of the pre-trained models that utilized multiple GPUs and took days to train for their application. There are few cases where existing BERT models cannot be used in place or tuned to a specific use case. BERT allows a team to accelerate solutions by tenfold. One can move to identify a business solution to building a proof of concept and finally moving that concept into production in a fraction of the time.