CS410 Text Information System Tech Review

A Brief Review of BERT and the State-Of-The-Art BERT Expansions

Hyeonjae Cho
hc53@illinois.edu

These days, all cutting-edge models are based on BERT or GPT. Both two models use transformer architecture as the basis. This implies that it is important to understand the transformer architecture thoroughly before I start to go over BERT. Therefore, this report first reviews transformer architecture and then explains BERT. In the last part, the most recent expanded BERT models are listed with brief explanations.

The evolution history of deep-learning-based machine translation models helps us understand the background of each model and the trend of techniques. In 1986, RNN first came out and LSTM, Seq2Seq followed. From 1986 to 2017, RNN was mainly used as a base model for NLP tasks. In 2015, the notion of attention was first suggested but the NLP model was still combined with RNN. In 2017, the Transformer model removed RNN or CNN and is solely based on attention architecture. That's the reason why the paper named "Attention is all you need". After the paper had been published, almost every NLP model is based on Transformer architecture.

So why did we come out with attention? The Seq2Seq model, which was the leading model before attention architecture, consists of an encoder and decoder. Here, the encoder creates a fixed-length-sized context vector containing the input sequence's information. This context vector is pushed to the first hidden state of the decoder. A decoder is based on RNNLM(RNN Language Model), so it predicts the next word based on the input. The problem occurs at this point. As the input sequence gets longer and longer, the worse model performs. As the context vector goes through decoder layers, the gradient vanishes, which is the intrinsic flaw of RNN. This is also called a long-term dependency problem. Therefore, attention alleviates the problem by providing weight vectors that contain relevance between words within the input sequence. Attention is used in every decoder step, but the weight keeps changing depending on the given step. As you see in the picture on the left, the previous hidden state in the decoder and all input sequence terms are used and produce weight after applying softmax to make a value range from 0 to 1.

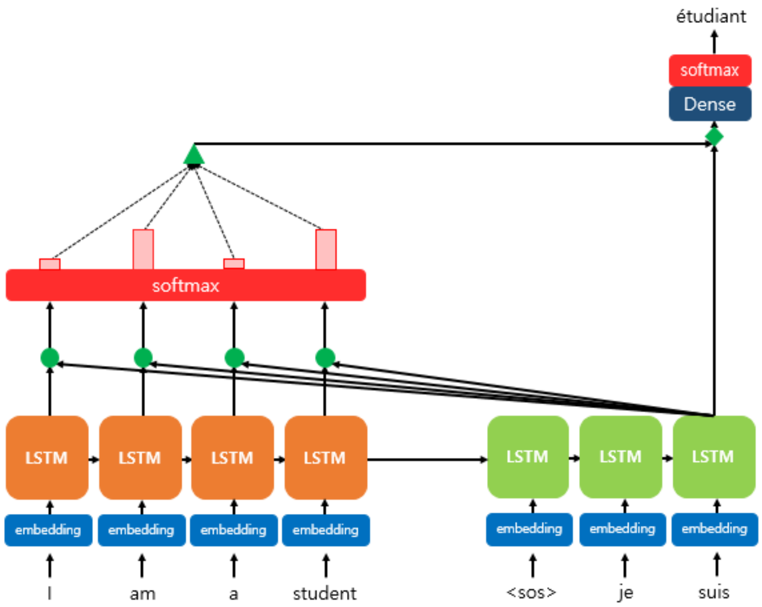


figure1 : https://velog.io/@sjinu/개념정리- Attention-Mechanism

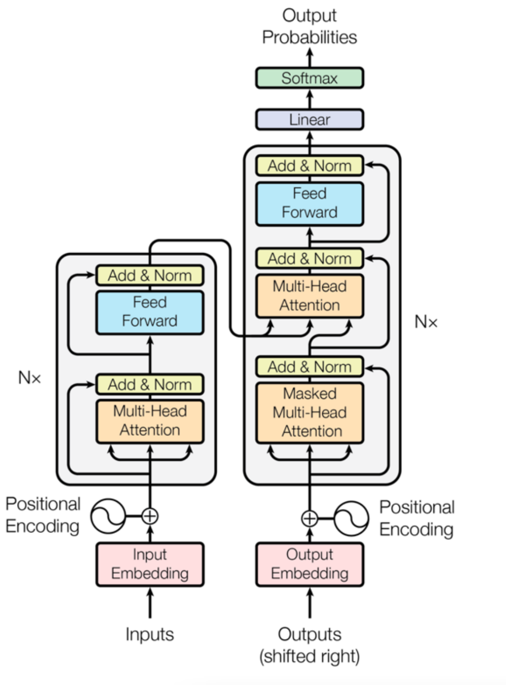


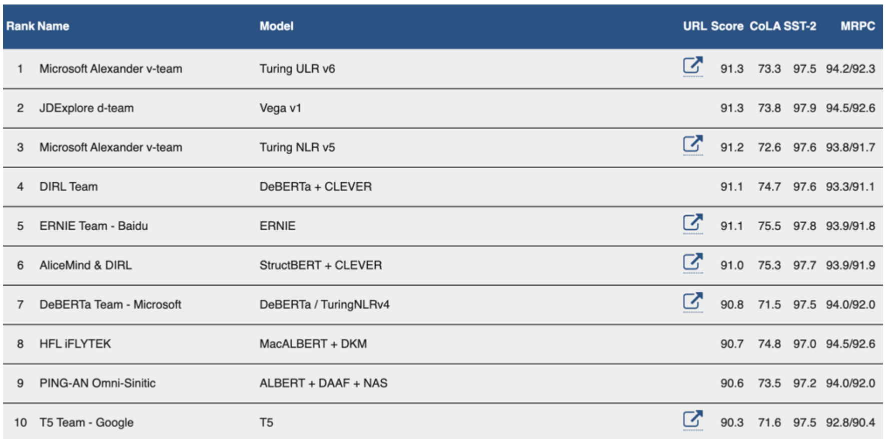
figure2 : https://arxiv.org/abs/1706.03762

In the encoder part of the Transformer, there are four main components. 1) embedding matrix, 2) positional encoding, 3) multi-head attention, 4) add + norm. In the beginning, the input sequence is represented as an embedding matrix and it is combined with positional encoding in an element-wise way. Then encoding has self-attention architecture inside, which computes the attention score representing the relevance degree of tokens within the input sequence. This helps the model train the information of the input sequence. Next, in add + norm part, the residual learning technique is used to enhance the training speed of the model. The combined result of positional encoding and embedding matrix is given as input again here by jumping off the multi-head attention part. Finally, the residual input and the output of multi-head attention are combined and then normalized. This becomes the output of the encoder and is used as an additional input in each layer of the decoder.

The decoder of the transformer has two attention architectures. First is self-attention and it works in the same way as that of the encoder. The self-attention part in the decoder assists the model to train the information of the output sequence it produces. The other attention part requires the output of the encoder and learns which output sequence is most related to the input sequence. This part is also called encoder-decoder attention. In this way, all the layers in the decoder get the output of the encoder as input.

To summarize, the transformer also uses the encoder-decoder architecture. However, it does not use RNN and rather exploits encoders and decoders multiple times. This makes the input sequence fed into the transformer model all at once unlike the previous models. Normally, we use the same number of layers in both the encoder and decoder for the transformer architecture.

BERT, which stands for Bidirectional Encoder Representations from Transformers is introduced in 2018. It is pre-trained using large-scale Wikipedia and BooksCorpus data. The model applies bidirectional training of the transformer, which is in contrast to the past when the model is trained in only one way. This helped the model capture the context of the language better than single-direction language models. In addition, BERT become famous in that data scientists only need to do transfer learning - just add one more neural net designed for their own task, and done – and the model performs quite well. BERT has two versions: BERT-Base and BERT-Large. The BERT-base model is the stack of 12 transformer encoder layers and BERT-Large consists of 24 layers.



Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC
1	Microsoft Alexander v-team	Turing ULR v6	↗	91.3	73.3	97.5	94.2/92.3
2	JDExplore d-team	Vega v1		91.3	73.8	97.9	94.5/92.6
3	Microsoft Alexander v-team	Turing NLR v5	↗	91.2	72.6	97.6	93.8/91.7
4	DIRL Team	DeBERTa + CLEVER		91.1	74.7	97.6	93.3/91.1
5	ERNIE Team - Baidu	ERNIE	↗	91.1	75.5	97.8	93.9/91.8
6	AliceMind & DIRL	StructBERT + CLEVER	↗	91.0	75.3	97.7	93.9/91.9
7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4	↗	90.8	71.5	97.5	94.0/92.0
8	HFL iFLYTEK	MacALBERT + DKM		90.7	74.8	97.0	94.5/92.6
9	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6	73.5	97.2	94.0/92.0
10	T5 Team - Google	T5	↗	90.3	71.6	97.5	92.8/90.4

The screenshot from the GLUE benchmark on the left let us know which models are state-of-the-art right now. In the top 10, five models are BERT-based enhanced models. The short description containing techniques of each model is as follows:

- DeBERTa

  The paper was published in ICLR 2021 by Microsoft. “De” stands for “Disentangled”. The model suggested two novel techniques: 1) disentangled attention mechanism, and 2) enhanced mask decoder. Disentangled attention is the technique that makes attention weight be represented as the combination of content vector and position vector. Enhanced mask decoder is used in the layer where predicts [mask] token by adding absolute position information.

- StructBERT

  The paper is published by Alibaba Group and the full title is “Incorporating Language Structures into Pre-training for Deep Language Understanding”. This model focused on the sequence of words and sentences for fluency. The model adds one more step, which predicts the sequence of certain tokens and sentences in the pre-training period. It suggests two pre-training objectives: 1) word structural objective, and 2) sentence structural objective. This leads the model better understand the sequential order of tokens.

- ALBERT

  ALBERT stands for “A Lite BERT for Self-supervised Learning Language Representations”. It suggested factorized embedding parameterization. Unlike BERT, it decomposes a large vocabulary embedding matrix into two small matrices. Each matrix contains token information and the output of contextualized representation learned by the transformer. Not only that, unlike the original transformer architecture, it suggests different layers can share the same parameters. Especially, the transformer layers can use the same parameter such as shared attention. This blocks the number of parameters grows exponentially as the network depth goes deeper. As a result, ALBERT trains about 1.7 times faster than BERT-large.