SEMINAR - B.TECH SEMESTER 7

Oct 21, 2021

TRANSFORMERS

Guide : Shivangi Modi
Presented by Krunal Rank (U18CO081)

Introduction

SEQUENCE TO SEQUENCE MACHINE LEARNING

Sequence to Sequence Machine Learning involves machine learning processes for analysing patterns in the given sequential data.

Sequential data is a type of data that is arranged according to a certain parameter, timestamp being the usual one. For example, time series based stock market data, user search results data or wave based audio data.

Sequence to Sequence Machine Learning involves usage of a new kind of machine learning technique - Recurrence. The common implementation of this technique is Recurrent Neural Networks.

Oct 21, 2021

Introduction

THE ADVENT OF TRANSFOMERS

Recurrent Neural Networks are an extension of conventional feed forward network for handling input data of variable length.

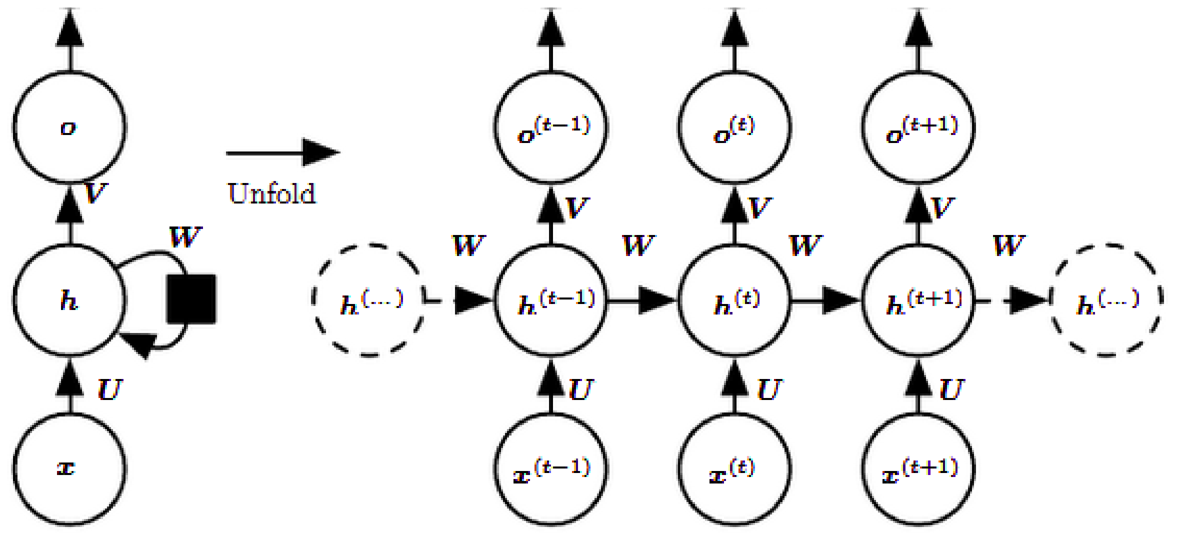
RNN and its extensions had a major drawback - sequential nature of processing input data, something which prevented them from parallel processing.

The Paper on Transformers (Attention is all you need by Google Brain) solved this particular problem with the help of a new concept - Self Attention and Multi Head Attention.

APPLICATIONS

- Machine translation - Translate a sequence of words from one human language to another
- Summarisation - Create condensed summaries out of given document that captures maximum information
- Speech to text recognition - Create transcriptions out of raw audio data
- Named entity recognition - Identify named entities in sentences
- Biological sequence analysis - Identify features, functions and structures of DNA and RNA.

Background



$$\begin{aligned}
a^{(t)} &= b + Wh^{(t-1)} + Ux^{(t)} \\
h^{(t)} &= \tanh(a^{(t)}) \\
o^{(t)} &= c + Vh^{(t)} \\
\hat{y}^{(t)} &= \operatorname{softmax}(o^{(t)})
\end{aligned}$$

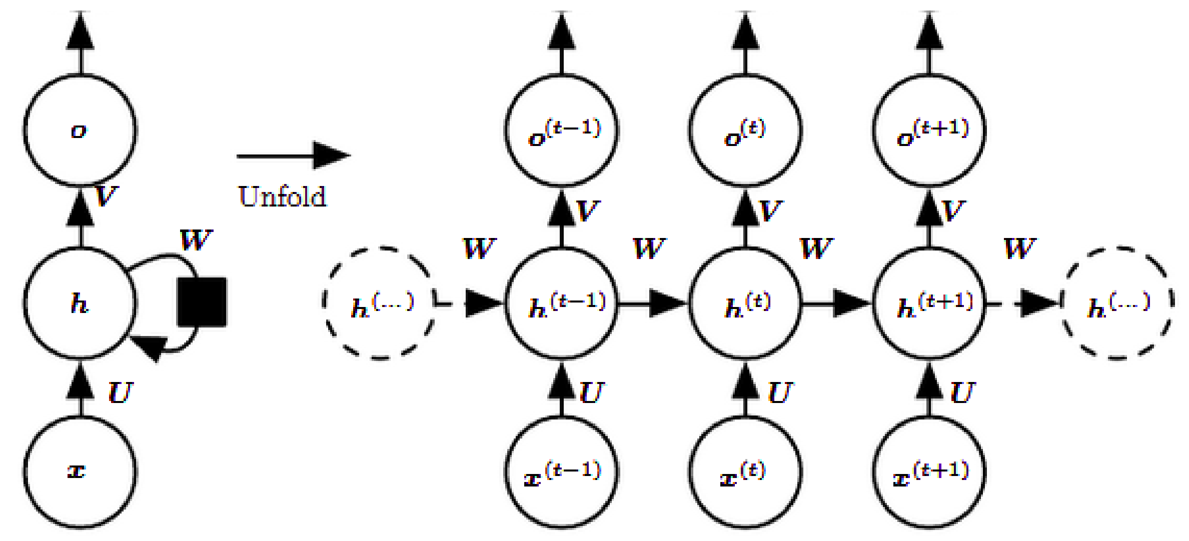
RECURRENT NEURAL NETWORKS

As explained earlier, RNNs can adequately handle time series data because they possess a recurrent hidden layer whose activation depends on the outputs of previous input data in the time series.

RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations.

Another way to think about RNNs is that they have a “memory” which captures information about what has been calculated so far.

Background



$$\begin{aligned} a^{(t)} &= b + Wh^{(t-1)} + Ux^{(t)} \\ h^{(t)} &= \tanh(a^{(t)}) \\ o^{(t)} &= c + Vh^{(t)} \\ \hat{y}^{(t)} &= \mathrm{softmax}(o^{(t)}) \end{aligned}$$

DRAWBACKS OF RNN

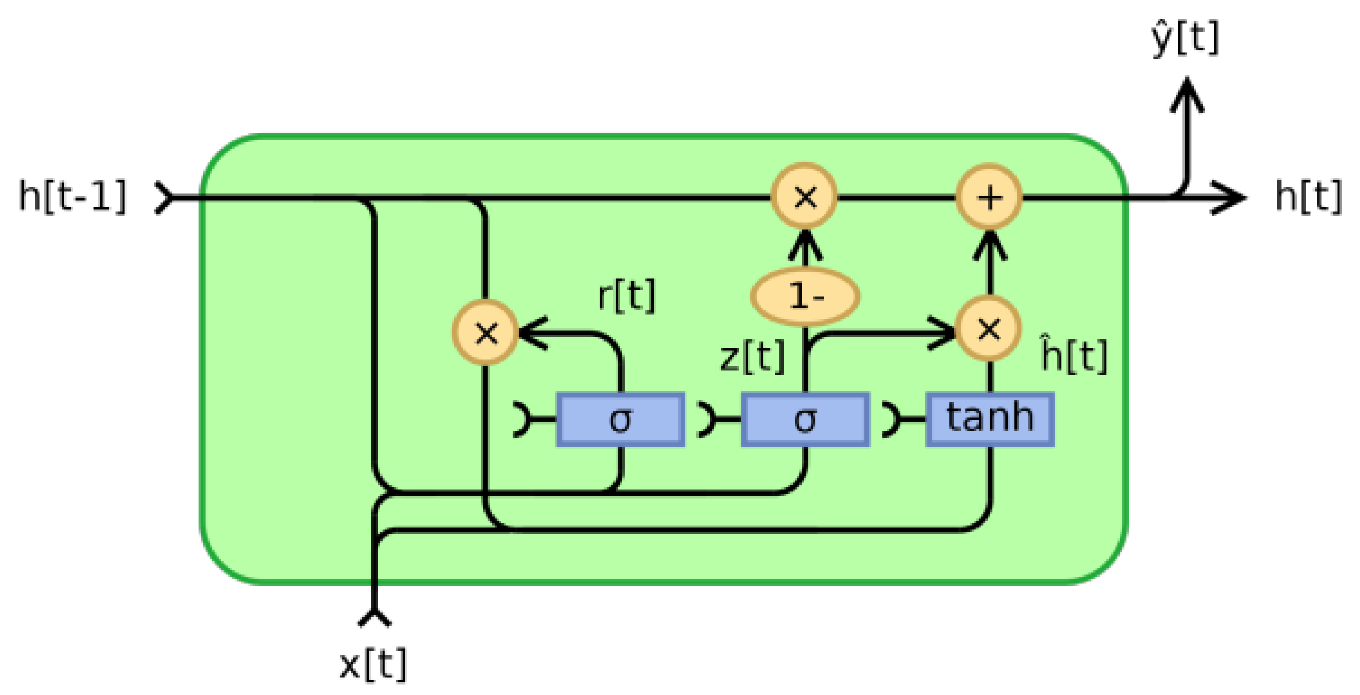
- Exploding gradients - The values of the parameters (W, U) tend to increase by tremendous amounts which needs to be avoided. It can be avoided by clipping the gradients using a certain threshold value.
- Vanishing gradients - The values of the parameters(W, U) tend to diminish and reduce to nearly 0 if the sequences are fairly long.

The problem of Vanishing gradients can be solved using an upgraded version as suggested in the paper

- **RNN suggested in the paper Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling** by Junyoung Chun and others
- **Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network** by Alex Sherstinsky (though LSTM was a concept discovered in 1737 and further researched in 1997)

Oct 21, 2021

Background



$$h_t = (1 - z_t)h_{t-1} + z_t \tilde{h_t}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$

$$\tilde{h_t} = tanh(W x_t + U(r_t \cdot h_{t-1}))$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

GATED RECURRENT NETWORK

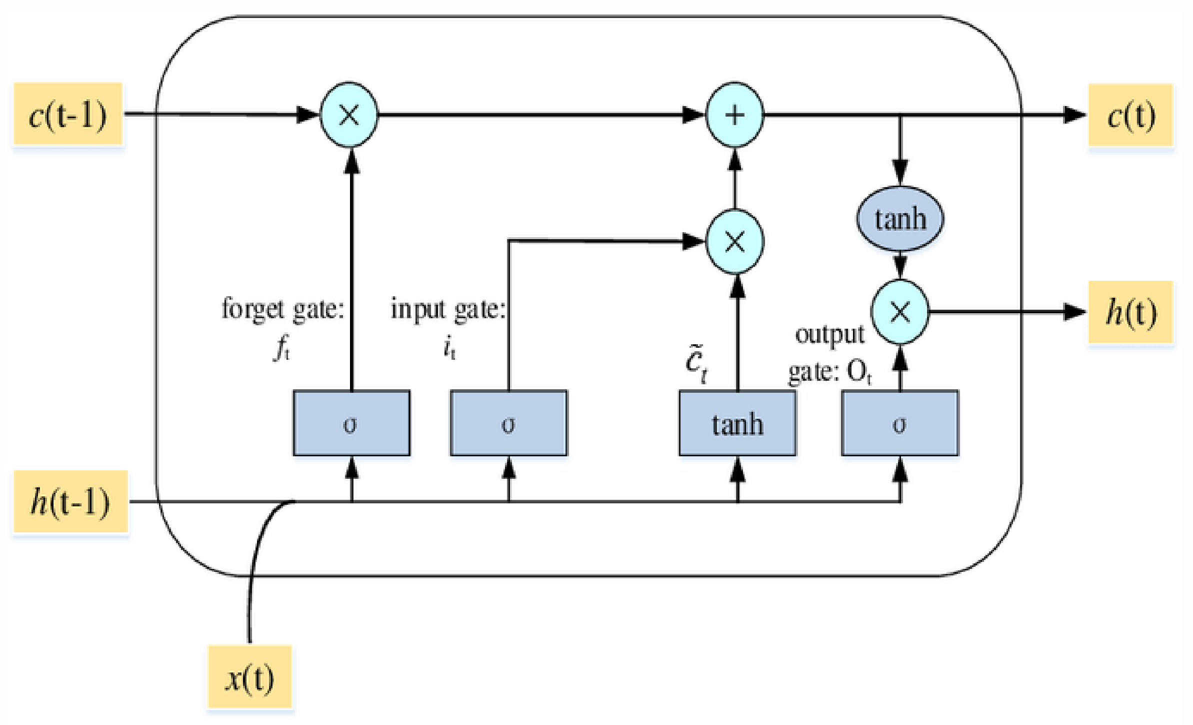
To solve the vanishing gradient problem, a variant of RNN having a more sophisticated activation function was introduced called Gated Recurrent Units (GRU).

Each unit can capture dependencies of various time scales. It uses gates to modulate the flow of information inside the unit.

The following gates are used:

- Update Gate
- Reset Gate

Background



$$h_t = o_t tanh(c_t)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t)$$

$$c_t = f_i c_{t-1} + i_t \tilde{c}_t$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1})$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1})$$

LONG SHORT TERM MEMORY (LSTM)

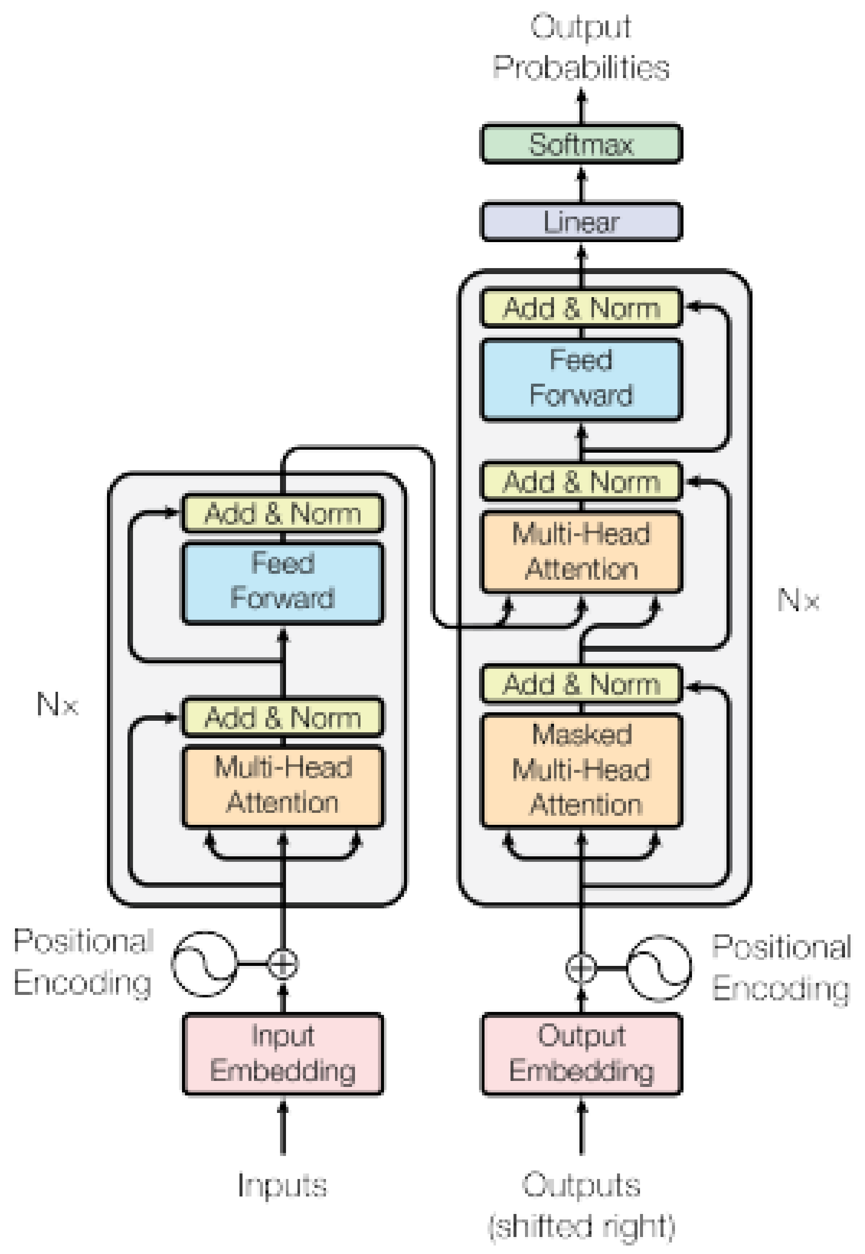
Another variant of RNN is the LSTM that contains an additional gate.

The nomenclature of gates in LSTM is as follows:

- Input Gate
- Forget Gate
- Output Gate

Each LSTM maintains a memory c at time t.

Background



TRANSFORMER

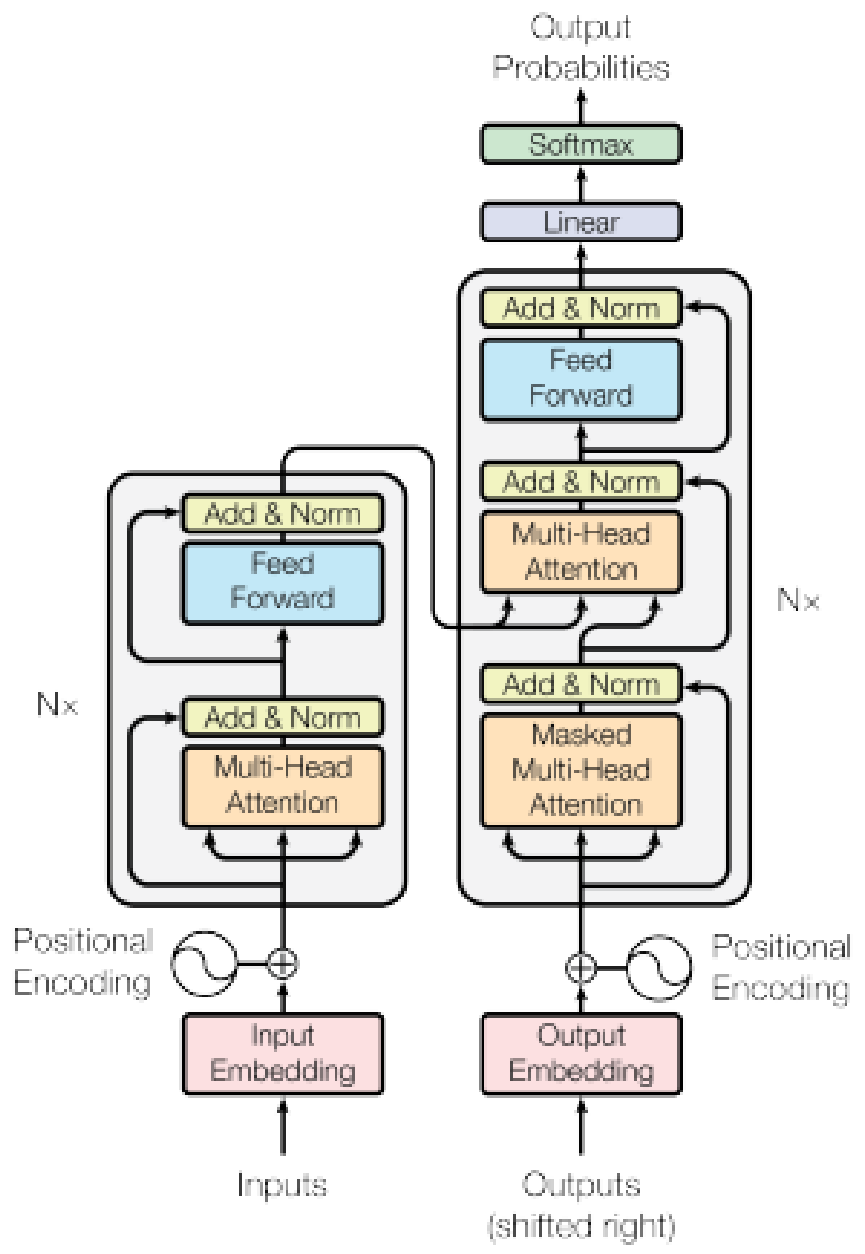
LSTMS and GRUs however process the data sequentially and hence, cannot use the parallel processing powers of the modern day GPUs.

Transformers introduced by the paper **Attention is all you need** by Google Brain, were a game changer.
It introduced a concept of Self Attention.

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

Oct 21, 2021

Background

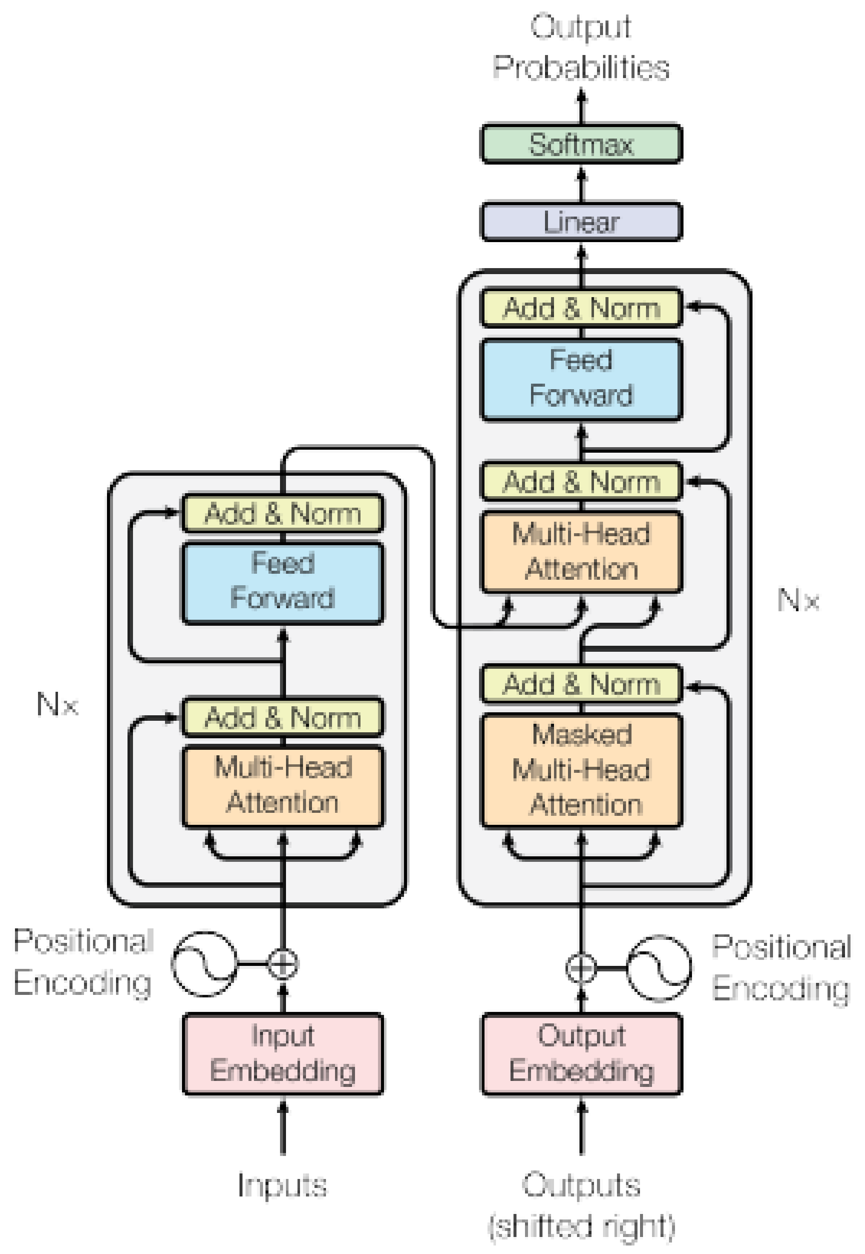


TRANSFORMER

The encoder is composed of a stack of N = 6 identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. A residual connection is employed around each of the two sub-layers, followed by layer normalization.

The decoder is also composed of a stack of N = 6 identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack.

Background



TRANSFORMER

In practice, the attention function is computed on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V .

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Instead of performing a single attention function with d model dimensional keys, values and queries,it was found beneficial to linearly project the queries, keys and values h times with different, learned linear projections to dk , dk and dv dimensions, respectively.
Multi-head attention allows the model to jointly attend to information from different representation sub-spaces at different positions. With a single attention head, averaging inhibits this.

$$\begin{aligned}\mathrm{MultiHead}(Q, K, V) &= \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O \\ \text{where } \mathrm{head}_i &= \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)\end{aligned}$$

Literature Review

SPEECH TO TEXT RECOGNITION

Encoder - Decoder based Sequence to Sequence models have proved to be state of the art models in Automatic Speech Recognition for a long time. One of the papers **Very deep self-attention networks for end-to-end speech recognition** by Ngoc Quan Pham explains the same.

The audio data is preprocessed first by converting raw waves into spectrograms that show frequency modifications over time. This can be done by using Short Time Fourier Transform to change audio data domain to frequency based.

The speech embedding layer is used to reduce dimensionality of the input data by using 1 Dimensional convolutional layers. This is identical to Down Sampling of given audio data.

The embeddings serve as the input data for the encoder.

Literature Review

SPEECH TO TEXT RECOGNITION

Output is processed character by character. 34 output characters are used as classes that include alphabetic characters, punctuation marks, unknown character markers and a Start of Sentence (SOS) and End of Sentence (EOS) marker.

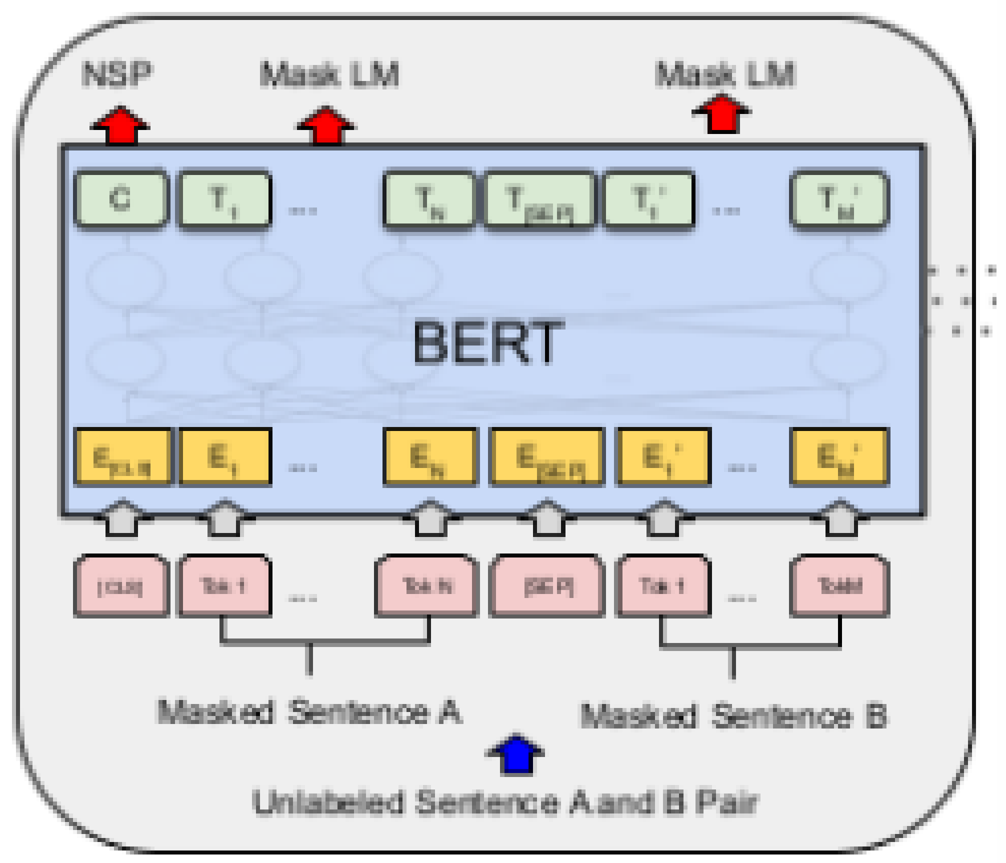
Token Embeddings are created using Output character vectors. These token embeddings are an addition of token vector embeddings and positional embeddings. These embeddings serve as an input for the Decoder.

The model is trained using the preprocessed data, using Adam's optimizer and a Learning Rate Schedule. The Decoder output is then passed to the character softmax and its loss is calculated using Categorical Cross Entropy.

Concluding the process, mapping acoustics directly to characters is actually a challenging task. The above pro-
cedure successfully demonstrated how this challenging problem can be approached.

Oct 21, 2021

Literature Review



Pre-training

BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS

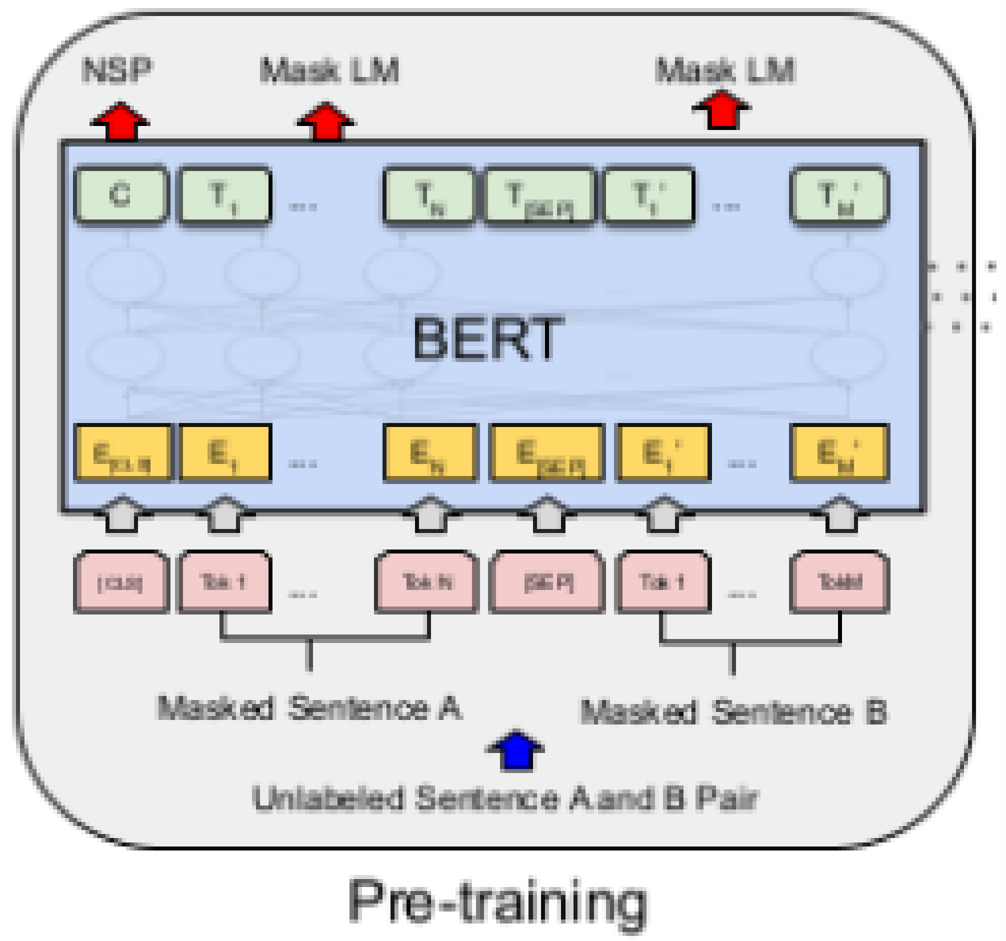
Bidirectional Encoder Representations from Transformers is a language representation model that is trained on unlabelled text by jointly conditioning on both left and right context in all layers.

According to the authors of the paper **BERT**, BERT is conceptually simple and empirically powerful, obtaining a state of the art result on Natural Language Processing tasks.

As far as BERT is concerned, there are two steps in the framework:
- Pre-training
- Fine-tuning

Oct 21, 2021

Literature Review



BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS

Pre-training BERT has two tasks:

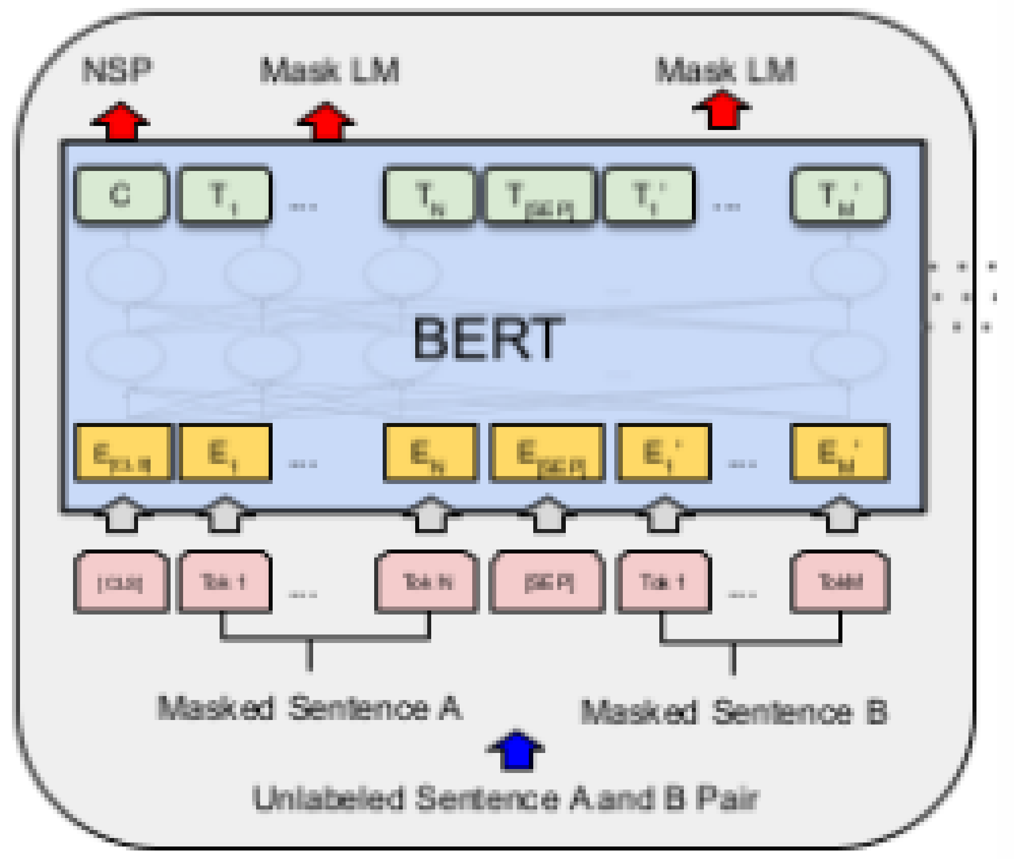
- Masked Language Model
- Next Sentence Prediction

In order to train the deep bidirectional representations, some random tokens in the input are masked and those tokens are predicted.
15% of the tokens are masked in each sequence. Now this may lead to a problem in fine-tuning process where in the [MASK] token is not observed in fine-tuning process.

Hence, out of the 15% masked tokens, 80% of these tokens are replaced with [MASK] token, 10% of these tokens are replaced with any random wordpiece and the rest are left unchanged.

The other unmasked tokens are then used to predict the masked tokens using a cross entropy loss.

Literature Review



Pre-training

BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS

Next Sentence Prediction

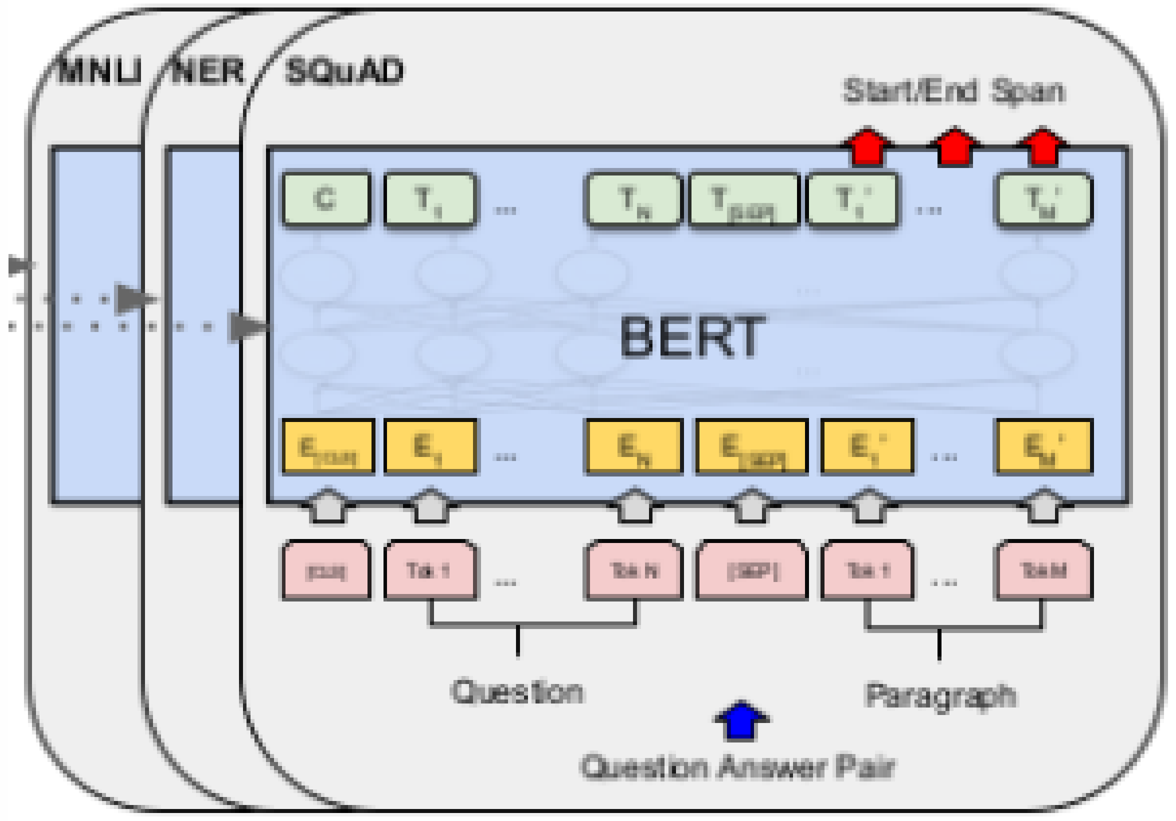
Some of the NLP based tasks such as Question Answering and Natural Language Inference are based on understanding the relationship between sentences, something which is not captured in language representations.

In order to train a model that understands sentence relationships, a prediction task wherein the model is expected to predict whether the given sentences are actually consecutive sentences or not.

Specifically, two sentences are chosen at random by the generator with a 50% chance that the sentence B is the next sentence for chosen sentence A.

These two sentences are separated by [SEP] token. This task is somewhat similar to representation learning objectives.

Literature Review



Fine-Tuning

BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS

Fine-tuning BERT is simpler as compared to pre-training process. For every task, the input and output sequences are plugged into BERT and parameters are fine-tuned end-to-end.

In pre-training stage, sentences A and B can be analogous to:
- Sentence pairs in paraphrasing
- Hypothesis Premise Pairs
- Question Passage Pairs

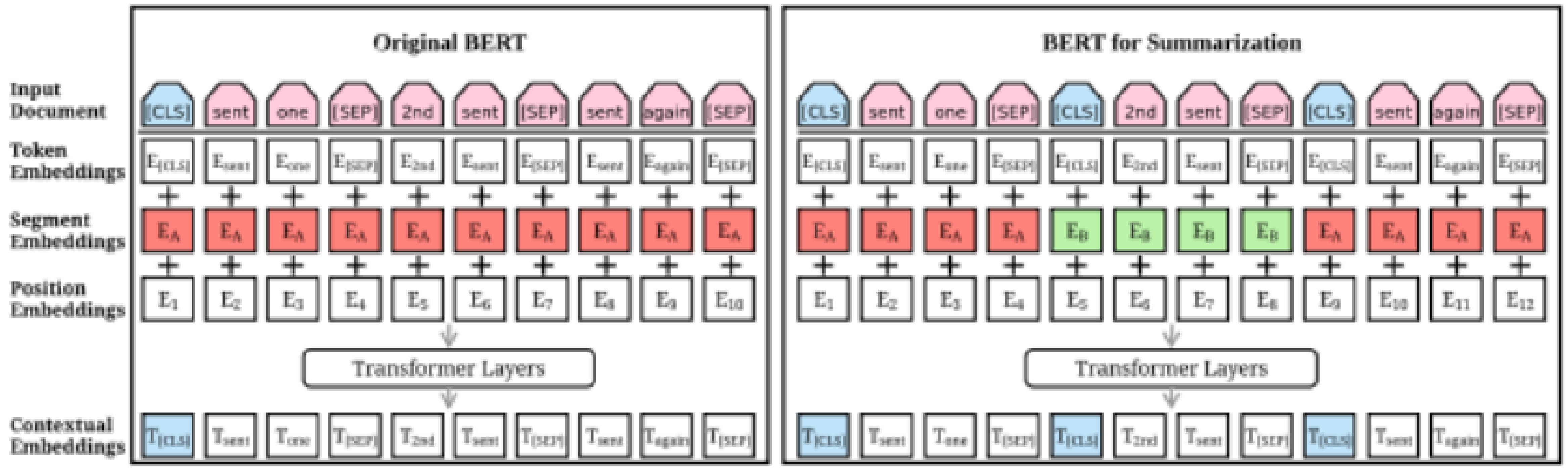
Compared to pre-training, fine tuning is relatively inexpensive and can be done pretty much faster.

Concluding, the major contribution of BERT is further generalizing these findings to deep bidirectional architectures, which allows the same pre-trained model to successfully tackle numerous NLP tasks.

Oct 21, 2021

Literature Review

BERTSUM (FINE-TUNE BERT FOR EXTRACTIVE SUMMARIZATION BY YANG LIU)



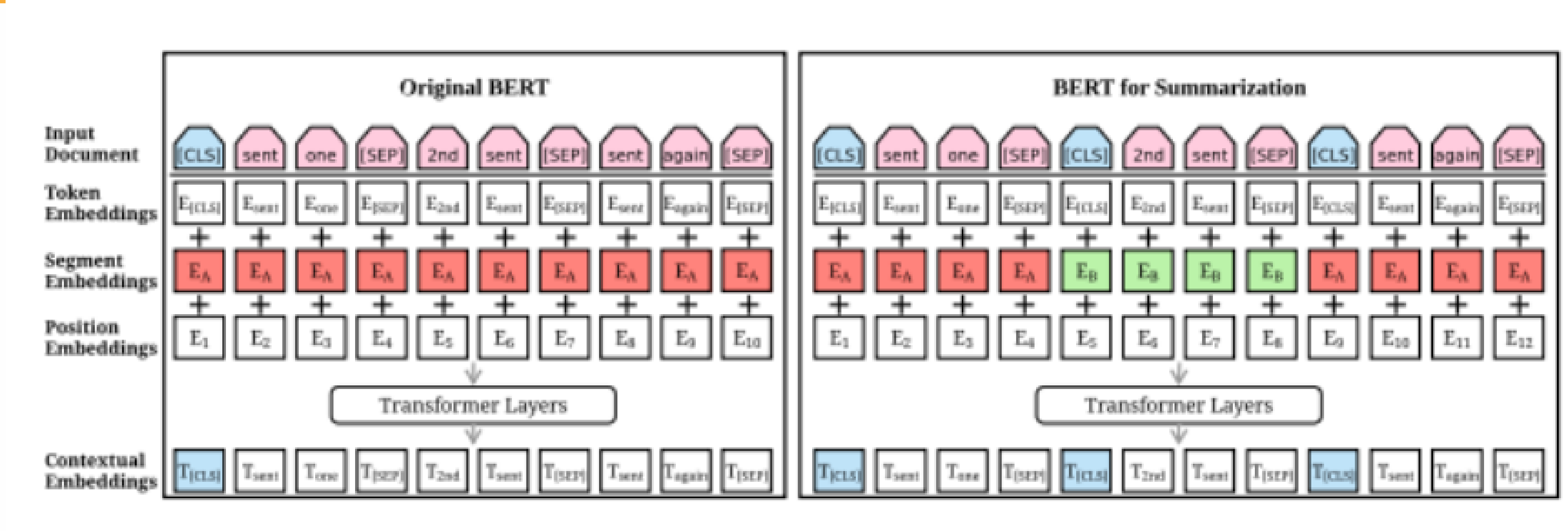
The way in which BERTSUM is represented is quite different from the original BERT. The sentences are separated using [CLS] token and features from the previous sentence are collected and processed by BERTSUM to create features for the current sentence.

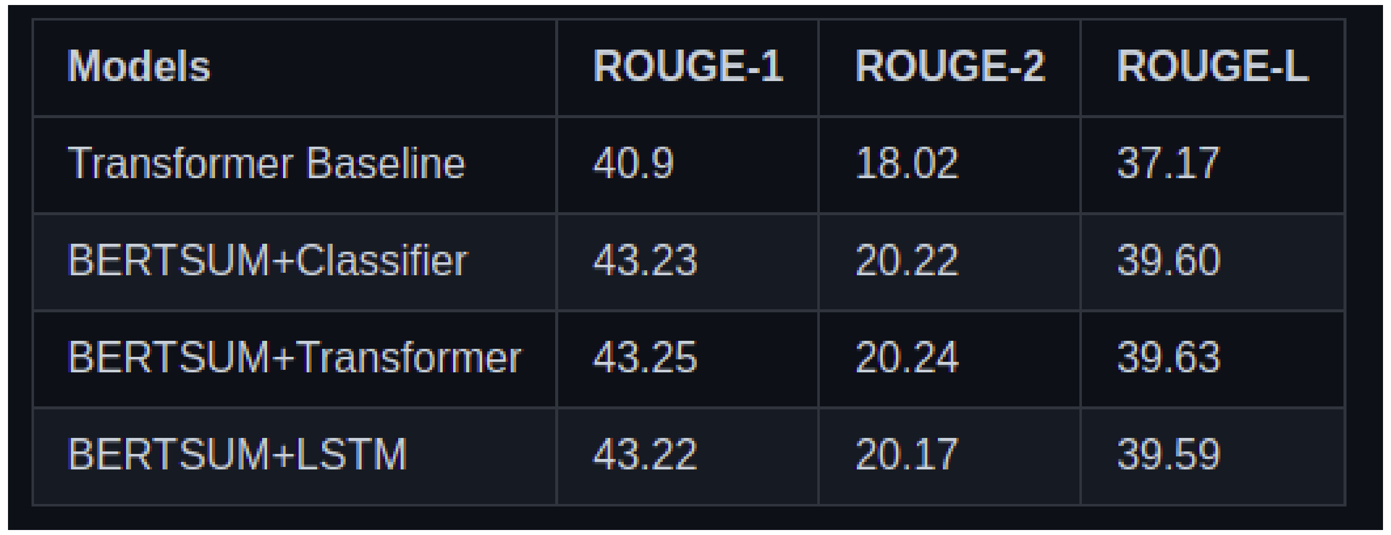
Interval Embedding is also used to demarcate sentences at odd and even positions so that document representations are learned hierarchically where lower Transformer layers represent adjacent sentences, while higher layers, in combination with self-attention, represent multi-sentence discourse.

BERTSUM outputs a list of scores that show representiveness of sentences towards document. Hence, the higher the score is, the more prominent the sentence is.

Literature Review

BERTSUM (FINE-TUNE BERT FOR EXTRACTIVE SUMMARIZATION BY YANG LIU)





Models	ROUGE-1	ROUGE-2	ROUGE-L
Transformer Baseline	40.9	18.02	37.17
BERTSUM+Classifier	43.23	20.22	39.60
BERTSUM+Transformer	43.25	20.24	39.63
BERTSUM+LSTM	43.22	20.17	39.59

Conclusion

I was able to thoroughly study the concepts of Recurrent Neural Network and its limitations.
The limitations of RNN are filled up by slightly modified versions of RNN - Gated Recurrent Units and Long Short Term Memory.
These networks require sequential processing and thereby weren't able to utilise the modern day GPU Parallel Processing, which was overcome with the advent of another new architecture, Transformers.
As I was able to study the above concepts, I also learnt a bit more on one of the use cases of Transformers, Automatic Speech Recognition and one of the improved versions of Transformers, Bidirectional Encoder Representations from Transformers.

Future Prospects

With the advent of modified transformers like BERT, certain NLP tasks have taken a different turn.
In case of summarization, the challenge has shifted from extractive summarization to abstractive summarization that completely changes the problem from choosing the given sentences to creating sentences and phrasings automatically from the provided input text document.
Even more advanced models are under research that have claimed to outperform BERT such as GPT2 and GPT3 by OpenAI, RoBERTa by Facebook AI, etc..