BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Presented by M. Moussou Koulibaly Traore

Work plan:

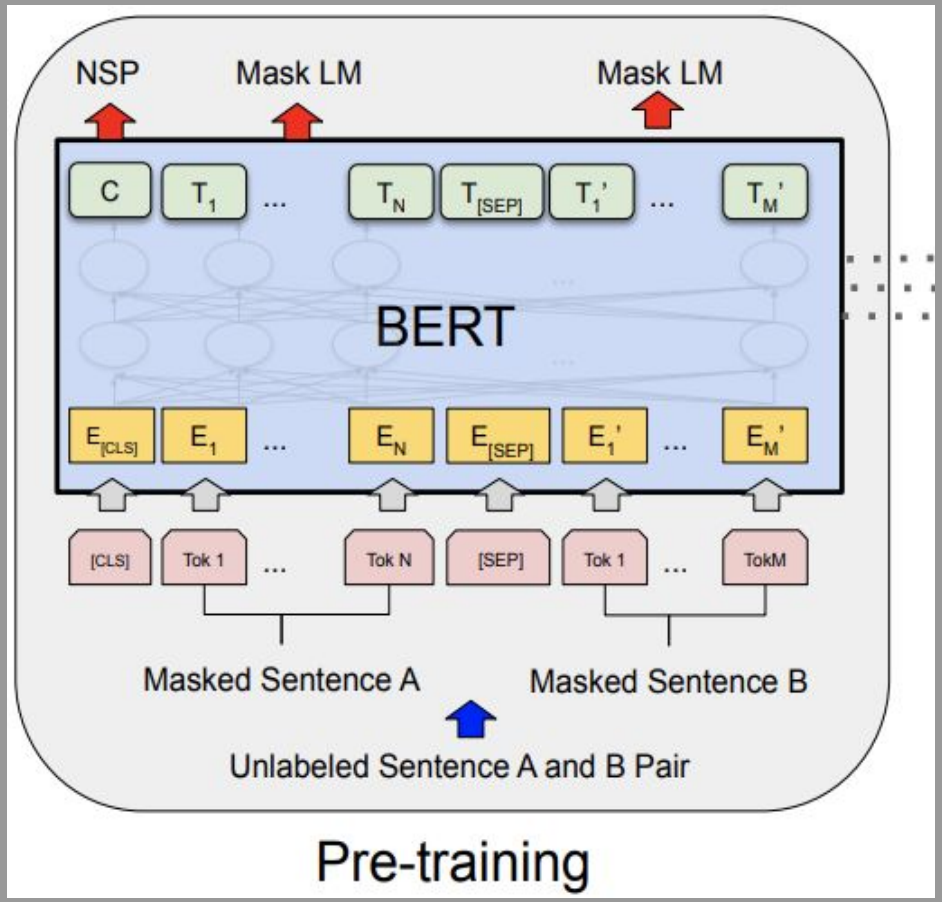
- ★ Introduction
- ★ Core Idea: BERT
  - ➢ BERT Steps
  - ➢ BERT's model architecture
- ★ Components
- ★ Implementation
- ★ Result
- ★ Conclusion

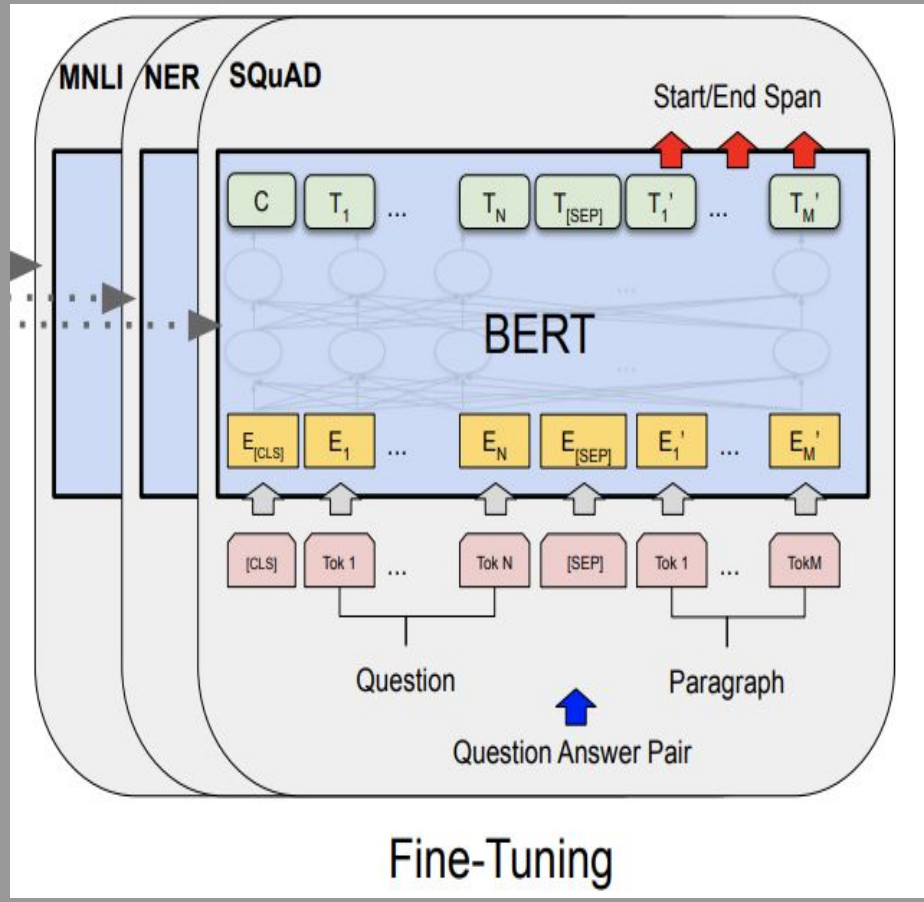
Introduction

One of the biggest challenges in NLP is the lack of enough training data. Overall there is enormous amount of text data available. But when we need to create data for specific we divide into many fields. And this can give a small trained labeled examples. Unfortunately, in order to perform well, deep learning based NLP. And many researchers try to solve this by . developed various techniques for training general purpose language representation models using the enormous piles of unannotated text on the web. And pre-trained model is one technic like **BERT.** **BERT** can solve many taskd like NMT, QA ,text generation, text classification ect..

BERT's Steps

BERT has mains steps: **pre-trained** and **fine-tuning.** For the first BERT learn language in two unsupervised tasks(mask langage modelling and Next sentence prediction). In the second the main question is "**How to use BERT for specific task**". We have the representation in this following graphs:

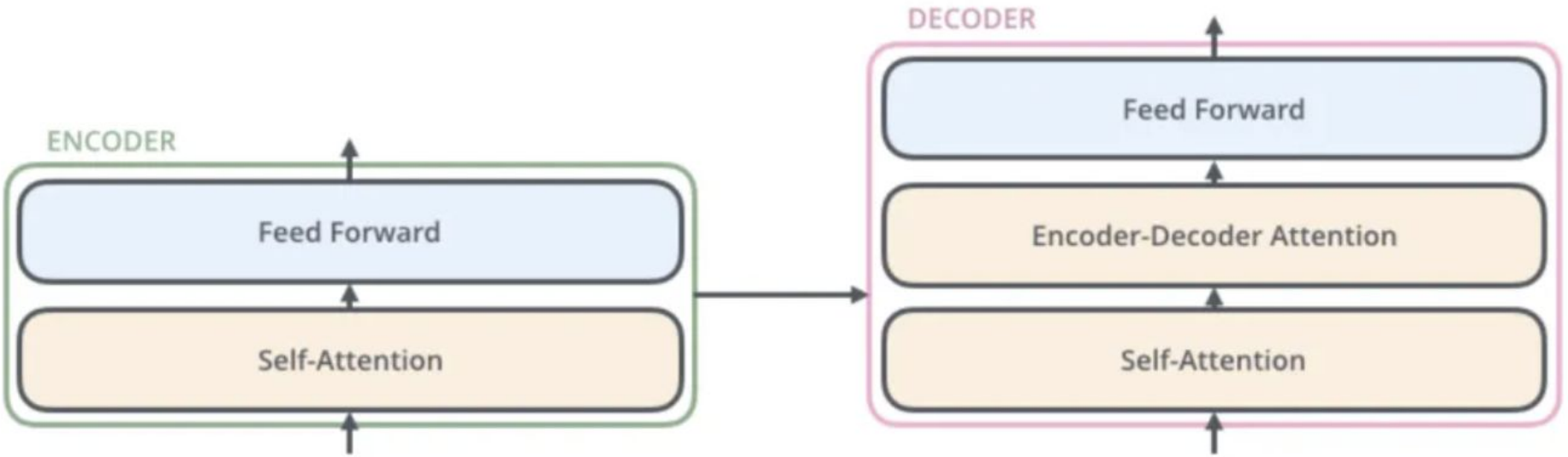

NSP
Mask LM
Mask LM
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair
Pre-training


MNLI
NER
SQuAD
Start/End Span
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Question
Paragraph
Question Answer Pair
Fine-Tuning

Components

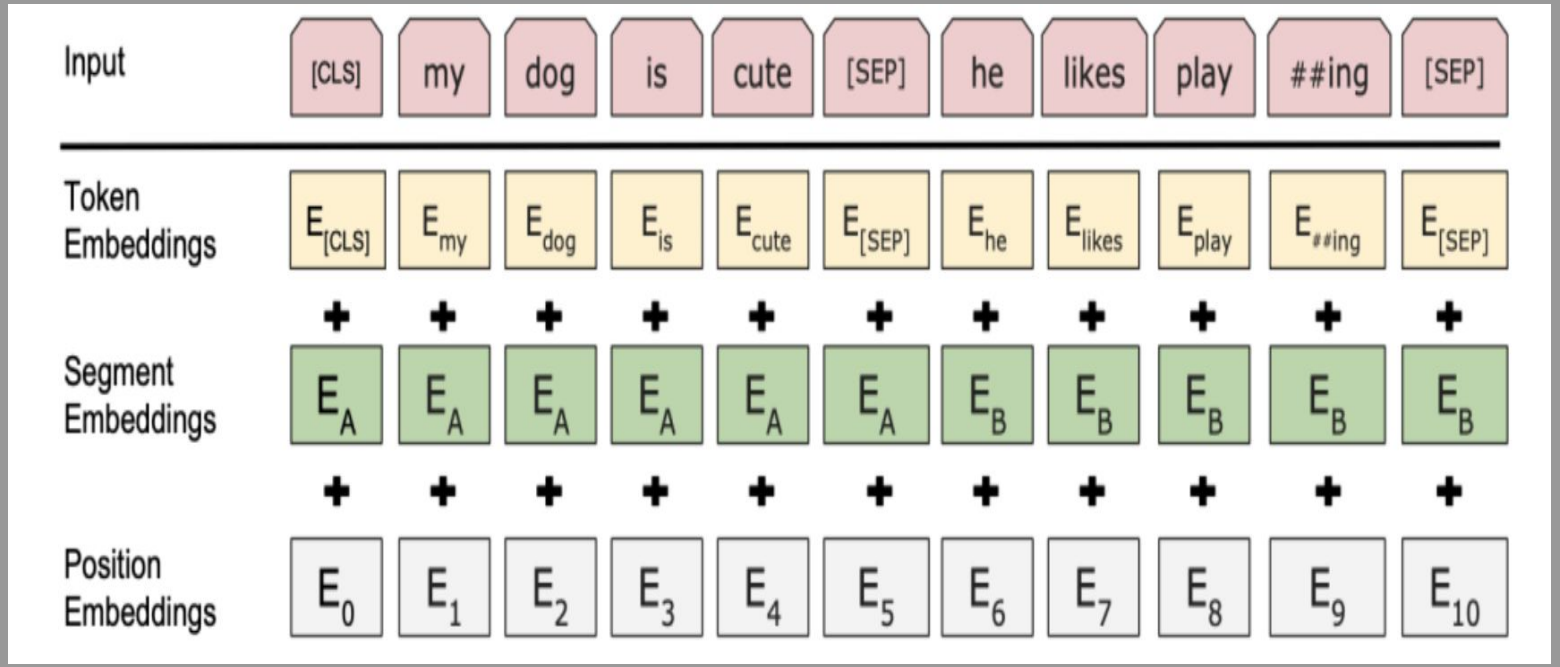
BERT’s main component is the transformer architecture(Encoder +Decoder).

We have the representation in the following graph.


DECODER
Feed Forward
Encoder-Decoder Attention
Self-Attention
ENCODER
Feed Forward
Self-Attention

Source

For BERT, the input sequence is formatting for three types of embedding. There are in the following graph:



BERT's architecture

We have two architectures for BERT. There are:

- **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

Implementation

In the paper, they do experiments using BERT fine-tuning on 11 NLP tasks like Question-Answering, Classification, and General language understanding `evaluation or (GLUE). I will implement Classification for` the Corpus of Linguistic Acceptability (CoLA) dataset for single sentence classification. It is a binary single-sentence classification task, where the goal is to predict whether an English sentence is linguistically “acceptable” or not.

Results:

After implementation we have the following results;

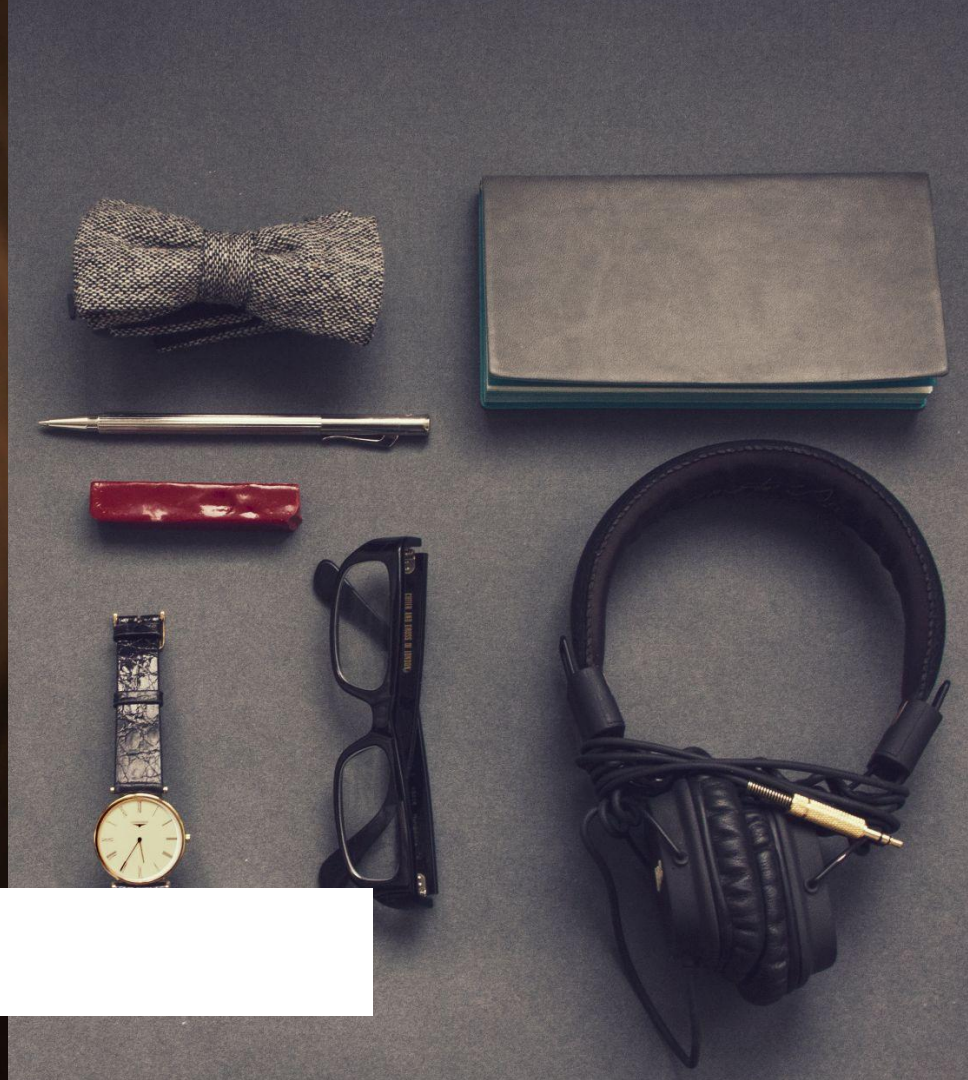
```
======== Epoch 1 / 4 ========
Training...

  Average training loss: 0.20

Running Validation...
  Accuracy: 0.85

======== Epoch 2 / 4 ========
Training...

  Average training loss: 0.17

Running Validation...
  Accuracy: 0.85

======== Epoch 3 / 4 ========
Training...

  Average training loss: 0.10

Running Validation...
  Accuracy: 0.85

======== Epoch 4 / 4 ========
Training...

  Average training loss: 0.09

Running Validation...
  Accuracy: 0.85

Training complete!
```

```
Predicting labels for 516 test sentences...
DONE.
```

```
Positive samples: 354 of 516 (68.60%)
```

Conclusion

This paper show that BERT is very interesting and can solve many tasks in NLP.



Thank You