BERT - Paper

Language Models LM.

A language model is a probabilistic model which lets us assign probabilities to a sequence of words.

A LM allows us to compute the following:

P ['China'] ['Shanghai is a city in"]

'China' → Next Token

'Shanghai is a city in" → Prompt

Usually a NN is trained to predict these probabilities. A NN trained on a large corpora of text is known as LLM.

How to train a LM?

→ Target sequence (10 tokens) → Before my bed lies a pool of moon bright [EOS]

↓

Entropy loss

↑

Output sequence (10 tokens) TK1 TK2 TK3 TK4 TK5 TK6 TK7 TK8 TK9 TK10

↑

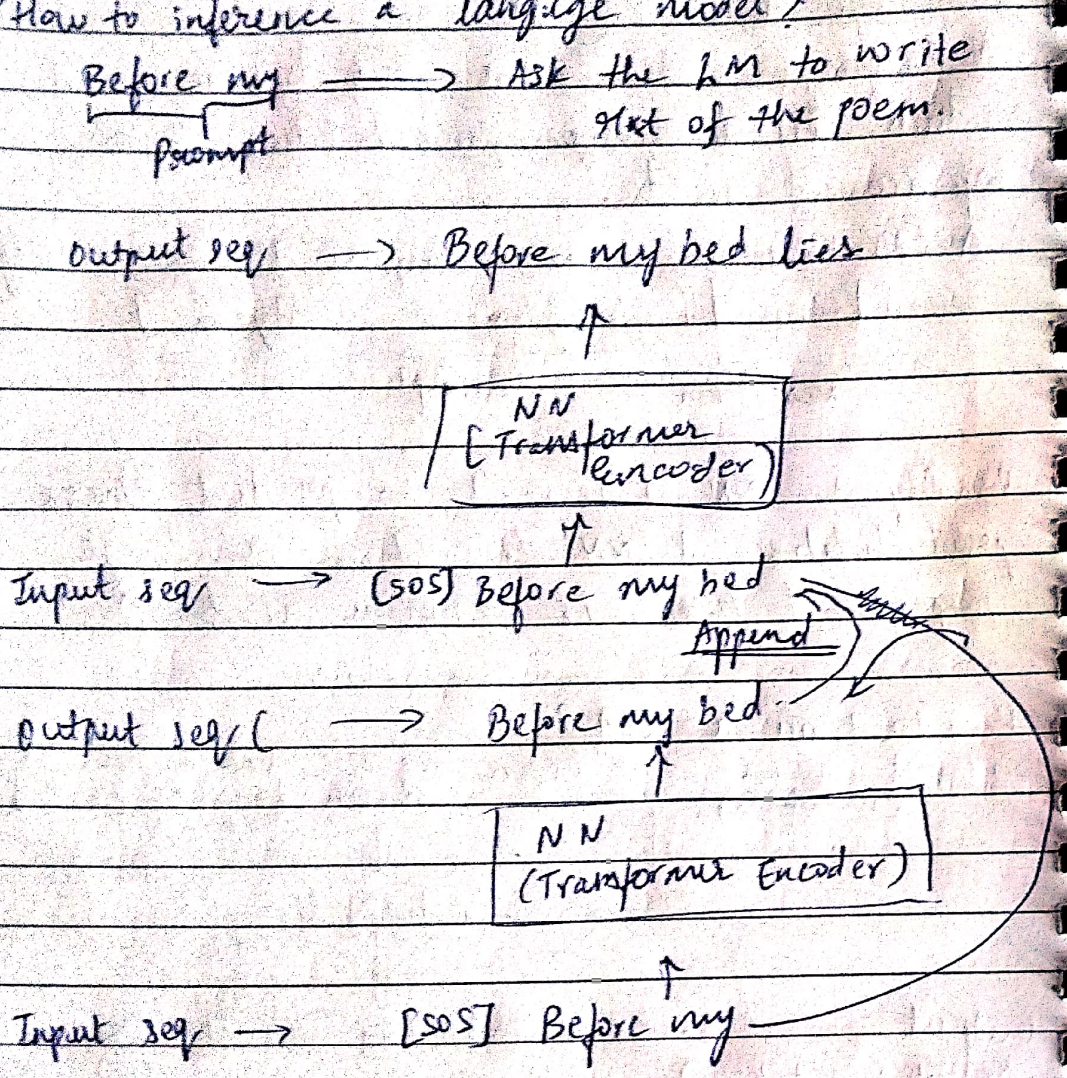
NN (Transformer Encoder)

↑

Input (10 tokens) → [SOS] Before my bed lies a pool of moon bright.

→ How to inference a language model?

Before my (prompt) → Ask the LM to write next of the poem.



Introducing BERT

BERT's arch is made up of layers of encoders of the Transformer model.

- BERT's base
  1) 12 encoder layers
  2) The size of the hidden size of the feed forward layer is 3072
  3) 12 attention heads.
- BERT's large
  1) 24 encoder layers
  2) Size of hidden size of feed forward layer is 4096.
  3) 16 Attention heads

Differences with Vanilla Transformer:

1) Embedding vector is 768 & 1024 for 2 models.
2) Positional embeddings are absolute and learnt during training & limited to 512 positions.
3) The linear layer head changes according to the application.
4) The linear layer head changes according to the application.

→ Used the wordpiece tokenizer, which also allows sub-word tokens. The vocabulary size is ~30,000 tokens.

BERT vs GPT/LLaMA

BERT stands for Bidirectional Encoder Representations from Transformers.

1. Unlike common LMs, BERT does not handle 'special tasks' with prompts, but rather, it can be specialized on a particular task by means of fine tuning.
2. Unlike common LMs, BERT has been trained using the left context and the right context both.
3. " " " " , BERT is not built specifically for text generation.
4. " " " " , BERT has not been trained on the Next Token Prediction Task, but rather, on the Masked Language Model and Next sentence Prediction task.

Tasks in GPT/LLaMA vs BERT

Question Answering in GPT/LLaMA
→ Prompt Engineering

Question Answering in BERT → Fine Tuning
Base-Trained BERT
↓
Fine Tune on QA

→ The importance of the left & the right context in BERT.

⇒ BERT - pre training

Masked Language Model (MLM)

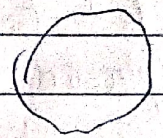
Also known as Cloze task, It means that randomly selected words in a sentence are masked, and the model must predict the right word given the left & right context.

Rome is the capital of Italy, which is why it host many government buildings.

↓ Randomly select one or more tokens and replace them with the special token [MASK]

Rome is the [MASK] of Italy, which is
" " " " " " "

↓



↓

capital

Left and right context in BERT

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

$$\text{softmax}\left[\frac{\underbrace{Q}_{(10,768)} \times \underbrace{K^T}_{(768,10)}}{\sqrt{768}}\right]$$

= None of the tokens is substituted by $-\infty$ to make it zero (0). This is the reason it is Bidirectional Encoder

Each token "attends" token to its left and tokens to its right in a sentence.
→ Basically NO MASK

→ Masked Language Model (MLM): details

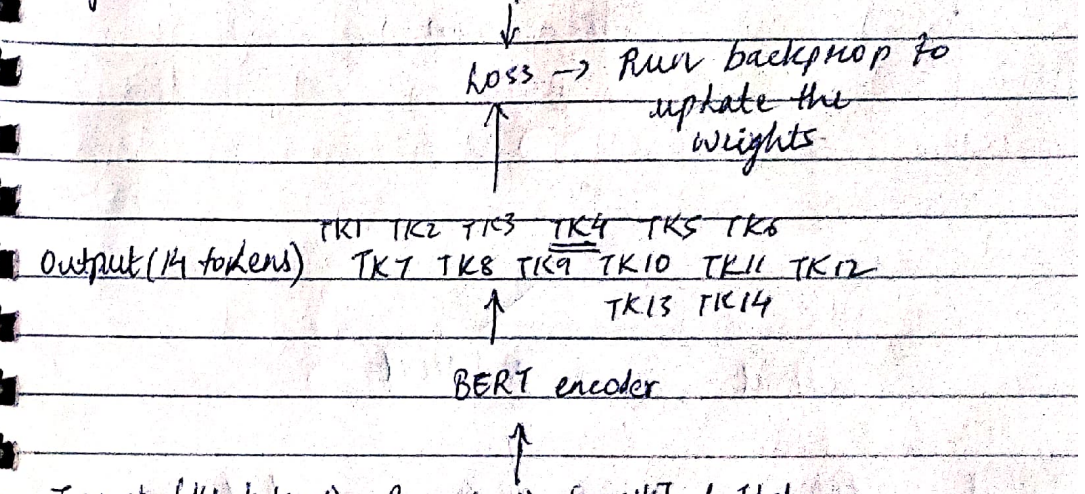
Rome is the capital of Italy, which is why it hosts many government buildings.

The pre-training procedure selects 15% of the tokens from the sentence to be masked.

When a token is selected to be masked

- 80% of the time is replaced with [MASK] token.
- 10% of the time it is replaced with a random token, e.g. Zebra
- 10% of the time it is not replaced.

⇒ MLM → training.



⇒ Next Sentence Prediction (NSP)

Many downstream applications (eg. choosing the right answer given 4 choices) require learning relationships between sentences rather than single tokens, that's why BERT has been pretrained on the Next Sentence Prediction Task.

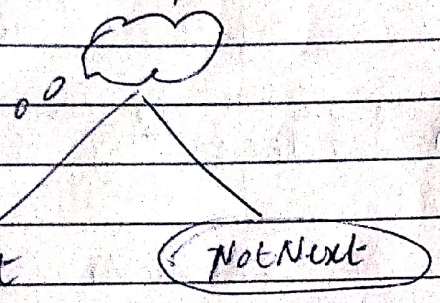
Sentence A → Before my bed lies a pool of moon bright

Sentence B → I could imagine that it's frost on the ground

- 50% of the time we select actual next sentence.
- 50% of the time we select a random sentence from the text.

Sentence A = Before my bed lies a pool of moon bright

Sentence B = I look up and see the bright shining moon



⇒ Next Sentence Prediction (NSP): segmentation embedding.

Given the sentence A and the sentence B, how can BERT understand which tokens belong to the sentence A and which to the sentence B?

We introduce the segmentation embeddings!

We also introduce two special tokens [CLS] and [SEP]

Input [CLS] my dog is cute [SEP] he likes play ##ing [SEP]

Token embeddings $E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$

Segment Embeddings E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B

Position Embeddings E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Next Sentence Prediction (NSP): Training

Target: Not Next

Linear layer (2 output features) + Softmax

Output (20 tokens): TK1 TK2 TK3 TK4 TK5 TK6 TK7 TK8 TK9 TK10 TK11 TK12 TK13 TK14

BERT encoder ← Sentence embedding

Input (20 tokens): [CLS] Before my bed lies a pool of moon bright [SEP] I look up and see the bright shining moon

[CLS] token in BERT

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

$$\text{softmax}\left[\frac{Q_{(10,768)} \times K^T_{(768,10)}}{\sqrt{768}}\right] \times V_{(10,768)} = \text{Attention Output}_{(10,768)}$$

The [CLS] token always interacts with all other tokens, as we do not use any mask.
So, we can consider the [CLS] token as a token that 'captures' the information from all the other tokens.

First Attention Output ← [CLS] (1st row & 1 column dot prod)

* BERT Fine tuning
  Text Classification
  → Assigning a label to a piece of text.
  Eg. Imagine we are running an ISP & receive complaints from customers. We may want to classify requests coming from the users such as hardware, software or billing

My router's led is not working, I tried changing the power socket but nothing	My router's web page doesn't allow me to change password anymore. I tried restarting but nothing	In this month, bill I have charged 100$ instead of the usual 60$, why is that?
HARDWARE	SOFTWARE	BILLING

TRAINING

Target (1 token): Hardware

Cross Entropy Loss → Run backprop to update the weights

Linear Layer (3 output features) + softmax

↑

Output tokens (16 tokens): TK1 TK2 TK3 TK4 TK5 TK6 TK7 TK8 TK9 TK10 TK11 TK12 TK13 TK14 TK15 TK16

↑

BERT Encoder

↑

Input (16 tokens) [CLS] My router's led ......

Question Answering Task

Context

Answering ques is the task given a ~~task~~ context

Context: "Shanghai is a city in China, it is also a financial center, its fashion capital and industrial city".

Ques: "What is the fashion capital of China"

Answer: "Shanghai is a city in China, it is also a financial center, its fashion capital and industrial city."

2 problems

1. We need to find a way for BERT to understand which part of the input is the context, which one is the question.
2. We also need to find a way for BERT to tell us where the answer starts and where it ends in the context provided.

-> Apply sentence A & sentence B logic.

[CLS] .... [SEP] ..... [SEP]

Sentence A: Question

Sentence B: Paragraph