

BERT 同时进行两个任务: { Masked Language Model,
Next Sentence prediction

BERT 语言模型任务一: Masked Language Model.

Masked Language Model 构建语言模型, 让模型通过上下文预测那个被遮盖或替换的部分
做 loss 的时候只计算被遮盖部分的 loss.

1. 随机把一句话中 15% 的 token (字或词) 替换成以下内容:
 - (1) 有 80% 的几率被替换成 [MASK] e.g my dog is hairy → my dog is [MASK].
 - (2) 10% 被替换成任一 token e.g my dog is hairy → my dog is apple
 - (3) 10% 不变 e.g my dog is hairy → my dog is hairy.
2. 之后让模型预测和还原被遮盖掉或替换掉的部分. 计算损失也只计算这部分

BERT 语言模型任务二: Next Sentence Prediction

[CLS] 句子1 [sep] 句子2 [sep]

句子1 句子2 可能是相关的, 或者是不相关的, 训练过程中会让这两种情况出现的数量为 1:1

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

| | | | | | | | | | | | |
|-------------------------|------------------|-----------------|------------------|-----------------|-------------------|------------------|-----------------|--------------------|-------------------|------------------|------------------|
| Token Embedding: | E_{cls} | E_{my} | E_{dog} | E_{is} | E_{cute} | E_{sep} | E_{he} | E_{likes} | E_{play} | E_{ing} | E_{sep} |
| | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow |
| Segmentation Embedding: | E_A | E_A | E_A | E_A | E_A | E_A | E_B | E_B | E_B | E_B | E_B |
| | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow | \uparrow |
| Position Embedding: | E_0 | E_1 | E_2 | E_3 | E_4 | E_5 | E_6 | E_7 | E_8 | E_9 | E_{10} |

Token Embedding: 正常词向量, 即 Pytorch 中 nn.Embedding.

Segmentation: 上句 token 全为 0. 下句全为 1.

Position Embedding: 与 Transformer 中不一样, 不是三角函数, 而是学习得来.

Multi-Task learning.

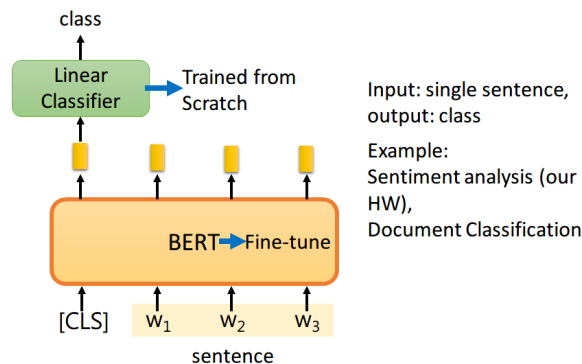
Input: [CLS] calculus is a [MASK] of math [SEP] it [MASK] developed.
by newton and Leibniz [SEP]

Output: false, branch, was

Fine-Tuning: Bert Fine-Tuning 共有4种任务

任务1: Classification

How to use BERT - Case 1



在开头加上一个代表分类的符号 [CLS], 然后将该位置的 Output 丢给 Linear Classifier.

Linear Classifier 参数从头开始学。
BERT 中参数会微调。

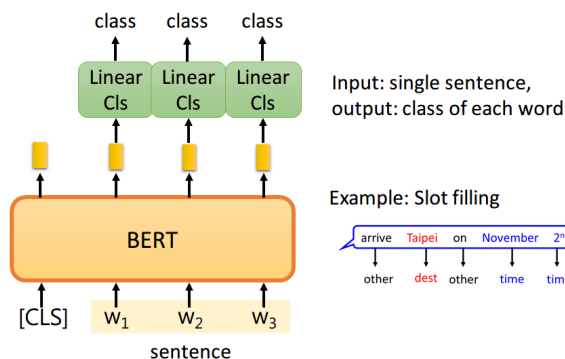
① Bert 内部是 Transformer, 而 Transformer 内部是 self-attention [CLS] 的 output - 适合整句的完整信息。

② self-attention 中, 自己和自己的值占大头, [CLS] 没有任何意义, 不会因为单个词对结果有太大影响。

③ 可将所有词结果 concat 起来。

任务2: Slot Filling

How to use BERT - Case 2



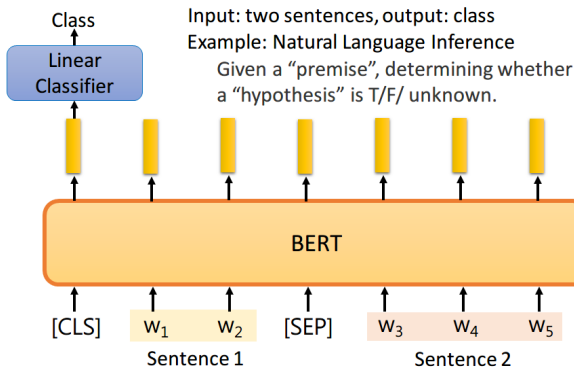
Example: Slot filling

arrive Taipei on November 2nd
other dest other time time

将句子中各个词对应的 output 分别送入 Linear Classifier

任务3: NLI (自然语言推理)

How to use BERT – Case 3



给出一个前提 (sentence1), 给出一个假设 (sentence2), 模型判断这个假设是正确, 错误, 还是不知道。也是用 [CLS] 给出预测。

任务4: QA 问答

How to use BERT – Case 4

- Extraction-based Question Answering (QA) (E.g. SQuAD)

Document: $D = \{d_1, d_2, \dots, d_N\}$

Query: $Q = \{q_1, q_2, \dots, q_N\}$

$D \rightarrow \text{QA Model} \rightarrow s$

$Q \rightarrow \text{QA Model} \rightarrow e$

output: two integers (s, e)

Answer: $A = \{q_s, \dots, q_e\}$

In meteorology, precipitation is any product of the condensation of 17 spheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupe and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain 77 at 79 are called "showers".

What causes precipitation to fall?

gravity s = 17, e = 17

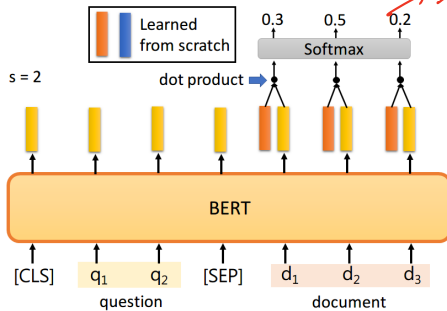
What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?

graupe
Where do water droplets collide with ice crystals to form precipitation?
within a cloud s = 77, e = 79

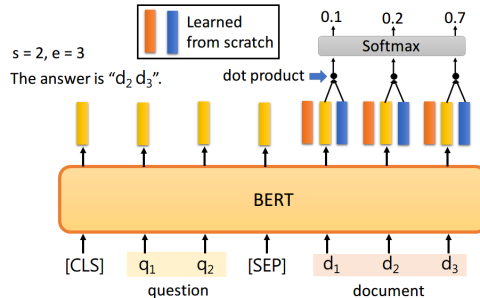
图中的标注取简单
答案一定会在文章中呈现 (start, end)

若 $s > e$ 在正确情况下会报错

How to use BERT – Case 4



How to use BERT – Case 4



将问题和文章用 [SEP] 分隔放入 BERT, 得到黄色输出。此时还需训练两个 vector (橙, 蓝) 分别和黄色向量 (输出) 进行 dot product。然后通过 softmax 看哪一个输出的值最大。