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 构建了语言模型，让模型通过上下文预测那个被遮盖或替换的部分
（在Loss的时候只计算被遮盖部分的loss.

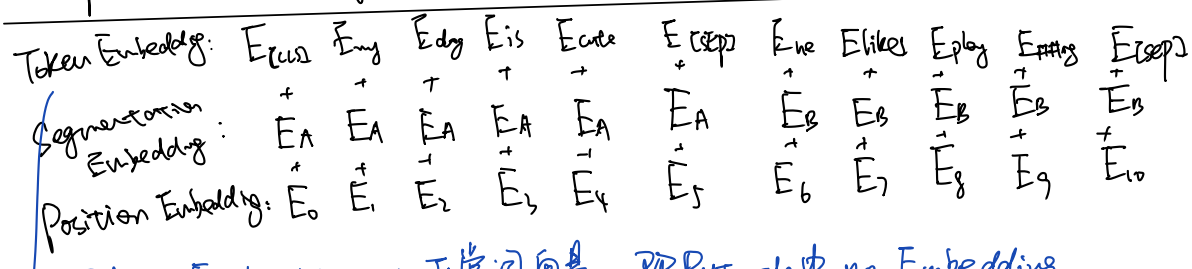
1. 随机把一句话中15%的token（字或词）替换成以下内容：
   (1) 有80%的机率被替换成[MASK] e.g my dog is hairy → my dog is [MASK].
   (2) 10%被换成任一token e.g my dog is hairy → my dog is apple
   (3) 10%不变 e.g my dog is hairy → my dog is hairy.
2. 之后让模型预测和还原被遮盖掉或替换掉的部分，计算Loss只计算这部分

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可能是连续的，或者是不连续的，实际训练过程中会让这两种情况出现的概率为1:1

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



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，而是学习得来的.

Multi-Task learning.

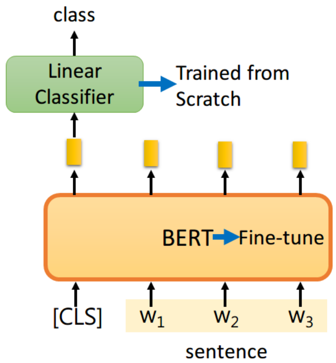
Input : [CLS] calculus is a [MASK] of math [SEP] it [MASK] developed by newton and leibniz [SEP]

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



Input: single sentence, output: class

Example: Sentiment analysis (our HW), Document Classification

句子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参数从头开始训练.
BERT中参数微调.

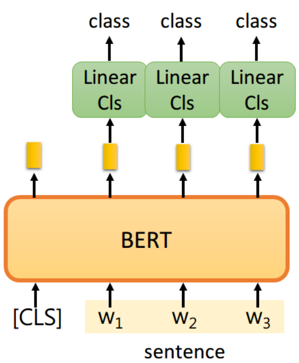
① Bert内部是Transformer，而Transformer内部是self-attention，[CLS]的output一定会包含整句话的完整信息.

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的影响偏向

③ 或将所有结果concat起来.

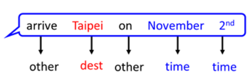
任务2：Slot Filling

How to use BERT – Case 2



Input: single sentence, output: class of each word

Example: Slot filling



将句子中每个字对应的output分别送入Linear Classifier

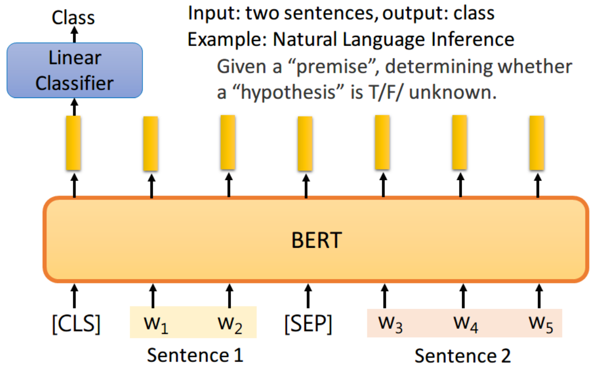
任务3：NLI（自然语言推理）

How to use BERT – Case 3

Input: two sentences, output: class

Example: Natural Language Inference

Given a "premise", determining whether a "hypothesis" is T/F/ unknown.



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。

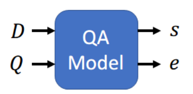
任务4：QA问答

How to use BERT – Case 4

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

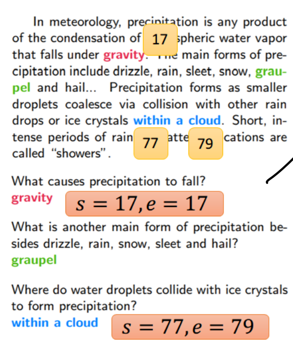
Document: $D = \{d_1, d_2, \cdots, d_N\}$

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



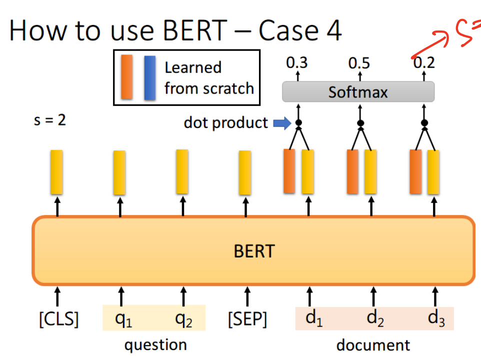
output: two integers (s, e)

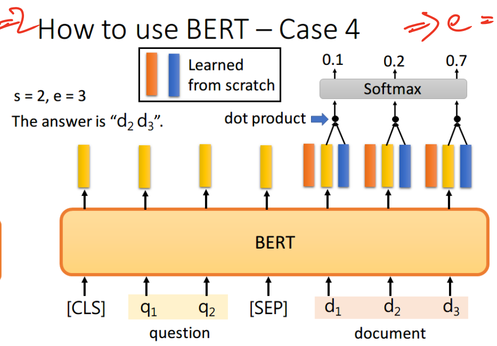
Answer: $A = \{q_s, \cdots, q_e\}$



图中的比较简单 答案一定会在文章中出现（从s到e）

若s>e 在正确情况下输出没有答案





⇒ s=2 ⇒ e=3

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