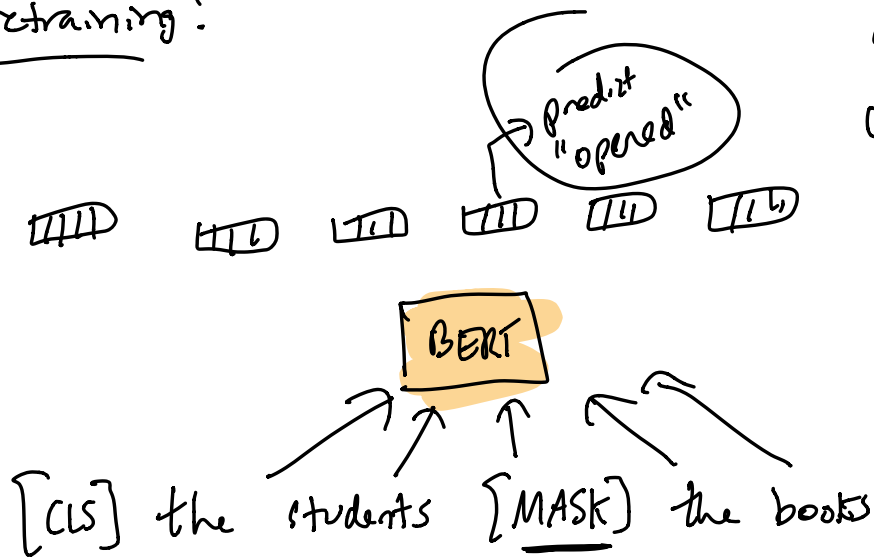
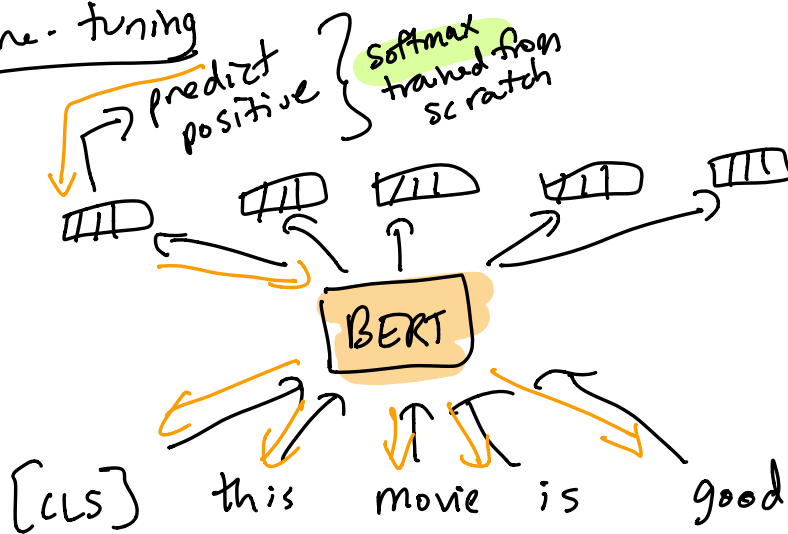


pretraining:



data:
comes from
huge volumes of
unlabeled text

fine-tuning



data:
comes from
labeled reviews
(for sentiment)

downstream task: task that we care about solving
(in this example, sentiment analysis)

can be contrasted w/ the pretraining task,
(in this case is masked LM)

BERT for sentence pair classification:

NLI: natural language inference

"textual entailment"

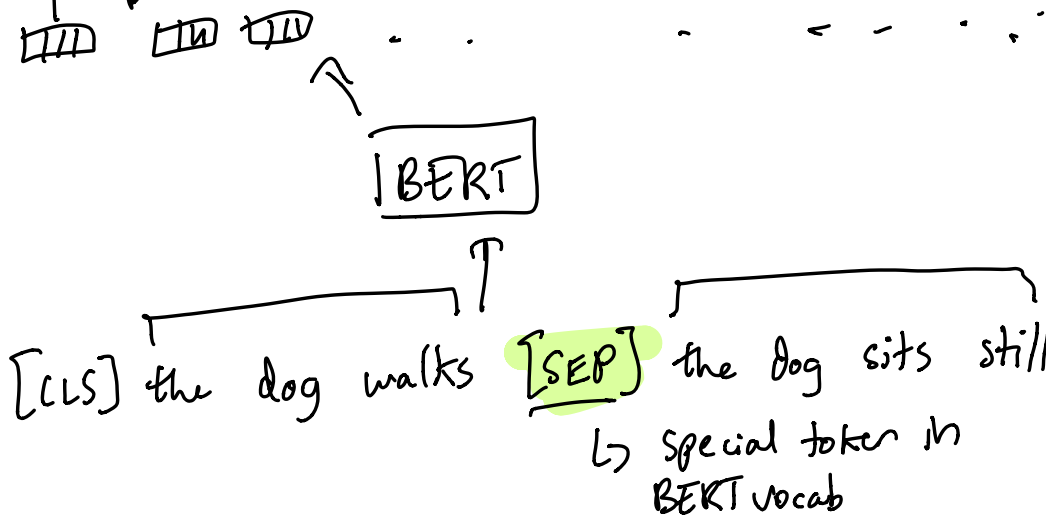
given two sentences (s_1, s_2) , a model must figure out if s_2

$\{ \text{entails, contradicts, neutral} \}$ to s_1 .

$s_1 = \text{"the dog walks"}$
 $s_2 = \text{"the dog sits still"}$ } contradiction

↳ SNLI, MNLI

softmax,
predict contradiction



BERT for extractive question answering:

↳ input: question and a passage

↳ goal: predict a contiguous span of text from the passage that answers the question

ex: SQuAD, QuAC, CoQA, HotPotQA,

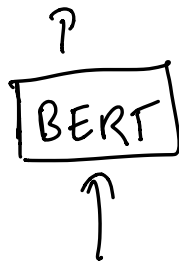
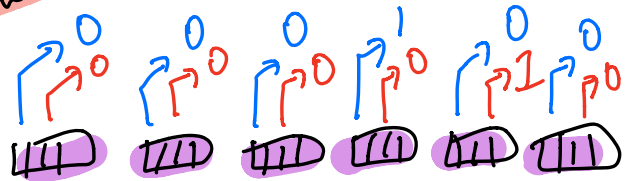
...

Q: Who starred in the Matrix or Neo?

P: 1 2 3 ...
... .. Neo was played by
actor [Keanu Reeves] ...
i j

A: (i, j)

two binary classifiers:
- predicts whether the token is the start of the answer
- predicts whether it is the end of the answer



[CLS] Who starred in the Matrix [SEP] p₁ p₂ p₃ Keanu Reeves p₆

how do we select an answer span at test time?

→ find the span $P_{i...j}$ that maximizes

$$P_{\text{START}}(i) \cdot P_{\text{END}}(j)$$

↳ exclude spans where $j < i$

↳ exclude spans longer than some threshold

advanced variants of BERT:

↳ pretraining improvements, \Rightarrow RoBERTa
more data

↳ longer sequences during pretraining
- BERT = 512 token
- XLNet = 900 tokens

↳ more pretraining objectives
↳ ELECTRA

↳ smaller models

↳ ALBERT, distilBERT, tinyBERT

RoBERTa: simple set of modifications

- train w/ bigger batches
 - ↳ smaller # of total batches
 - ↳ gradient accumulation
 - ↳ bypasses GPU mem. limitations
- has no pretraining task for [CLS]
- pretrain on more data
 - 16 GB ⇒ 160 GB
 - ↳ common crawl
 - ↳ URLs from reddit
- pretrained for longer
 - ↳ more total batches / epochs, 500k steps