CS60010: Deep Learning

Spring 2021

Sudeshna Sarkar

Module 5

BERT and Family

22 Mar 2021

1-of-N Encoding

apple = [1 0 0 0 0]

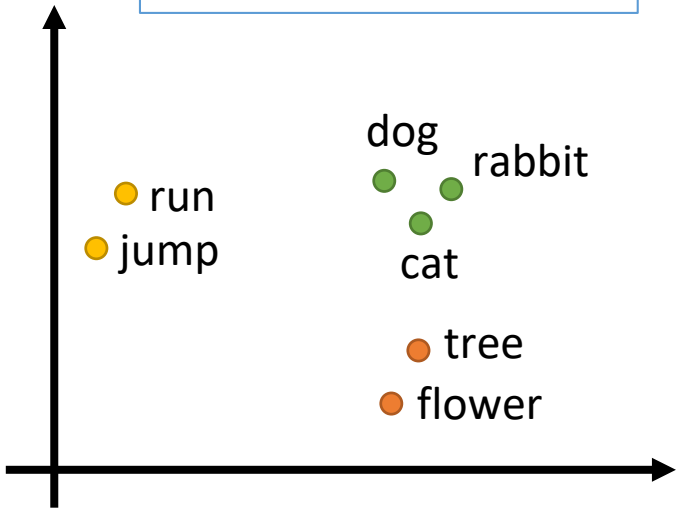
bag = [0 1 0 0 0]

cat = [0 0 1 0 0]

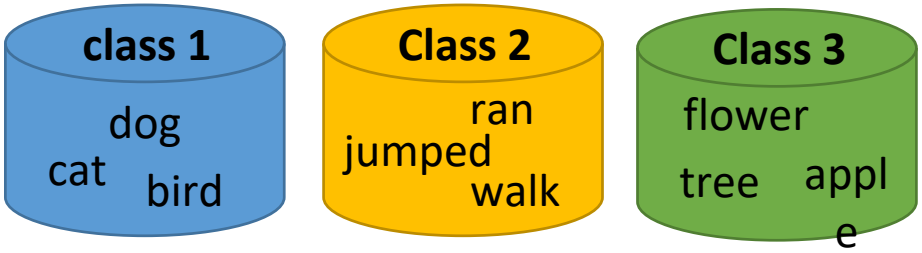
dog = [0 0 0 1 0]

elephant = [0 0 0 0 1]

Word Embedding



Word Class



A word can have multiple senses.

Have you paid that money to the bank yet ?

It is safest to deposit your money in the bank .

https://arxiv.org/abs/1902.06006

The victim was found lying dead on the river bank .

They stood on the river bank to fish.

The hospital has its own blood bank.

The third sense or not?

word2vec represent each word type with a single vector

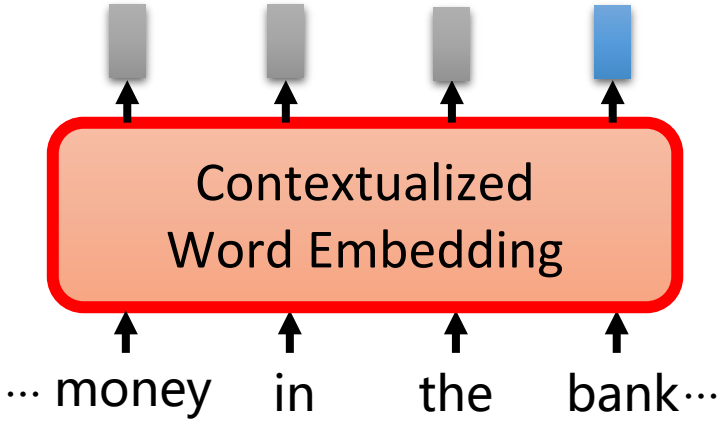
play =[0.2, -0.1, 0.5, ...]

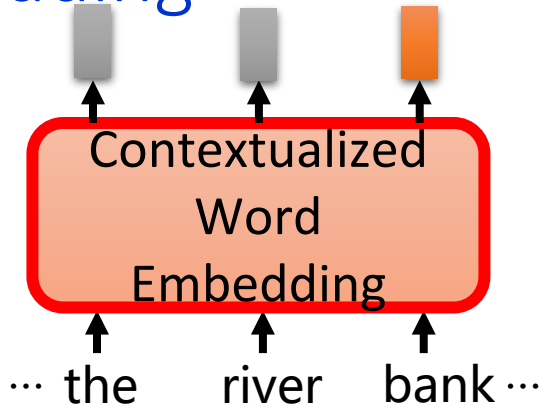
bank =[-0.3, 1.4, 0.7, ...]

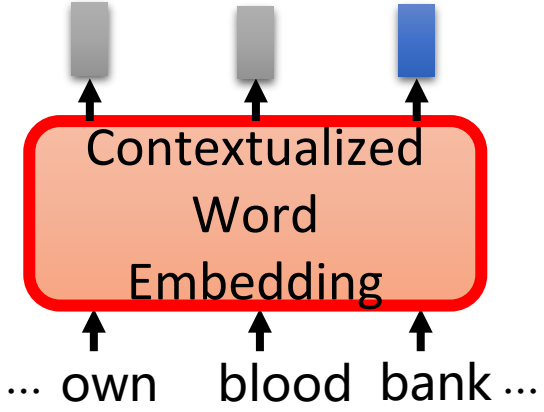
run =[-0.5, -0.3, -0.1, ...]

- The new-look play area is due to be completed by early spring 2010 .
- Gerrymandered congressional districts favor representatives who play to the party base .
- The freshman then completed the three-point play for a 66-63 lead .

Contextualized Word Embedding



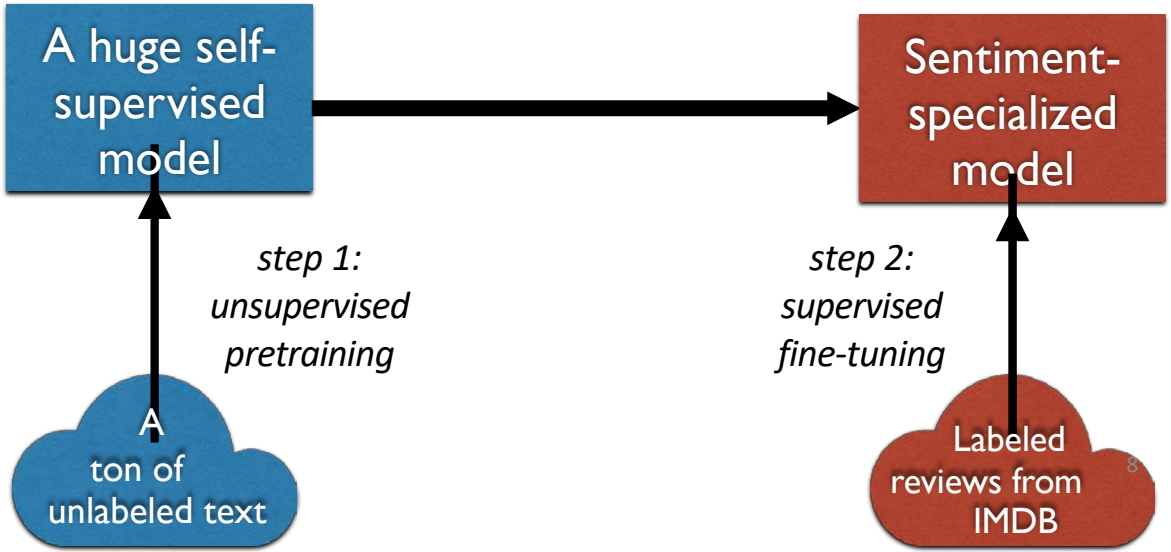




Train contextual representations on text corpus

Transfer Learning

- Can we leverage unlabeled data to cut down on the number of labeled examples we need?
- Take a network trained on a task for which it is easy to generate labels, and adapt it to a different task for which it is harder.
- Train a really big language model on billions of words, transfer to every NLP task!


A huge self-supervised model
Sentiment-specialized model
step 1: unsupervised pretraining
step 2: supervised fine-tuning
A ton of unlabeled text
Labeled reviews from IMDB

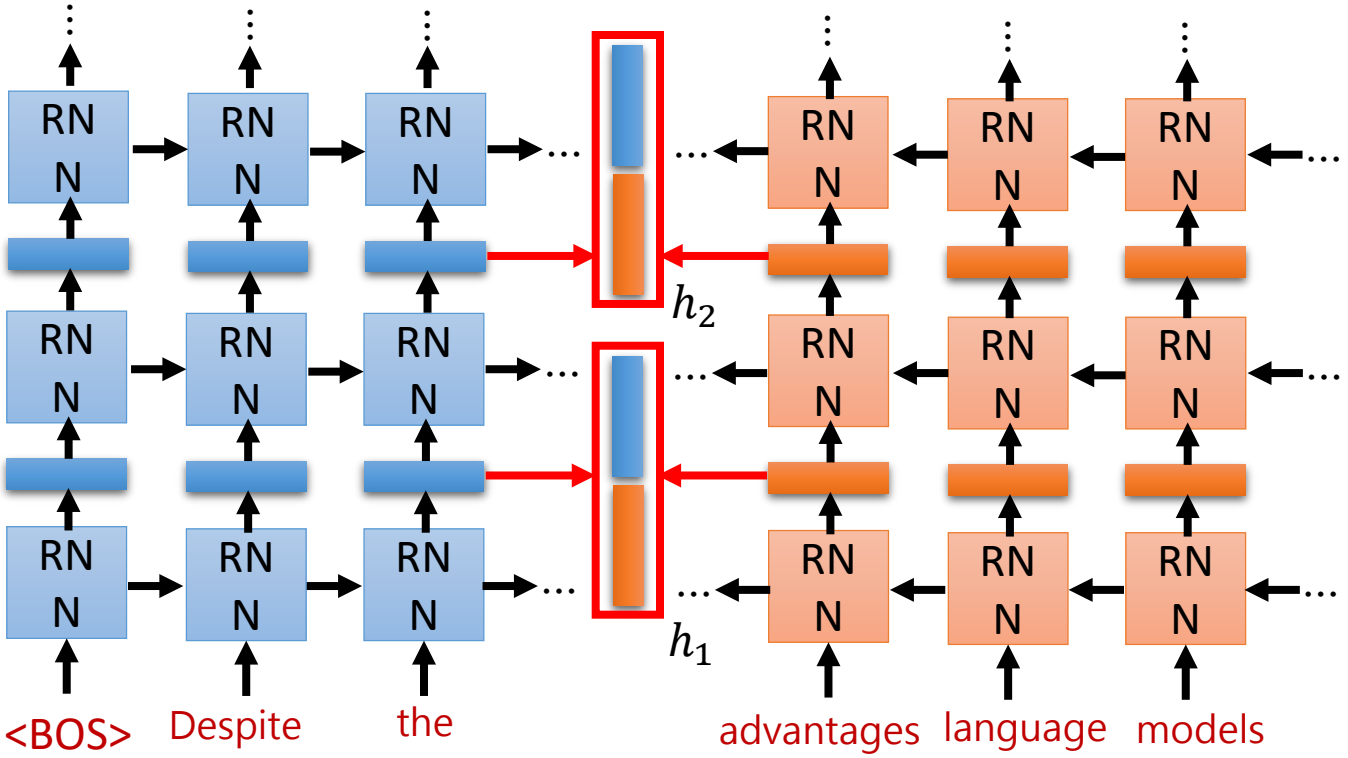
Problem with Previous Methods

- **Problem**: Language models only use left context *or* right context, but language understanding is bidirectional.
- Why are LMs unidirectional?
  - Reason 1: Directionality is needed to generate a well-formed probability distribution.
  - Reason 2: Words can "see themselves" in a bidirectional encoder.

Slide from Jacob Delvin

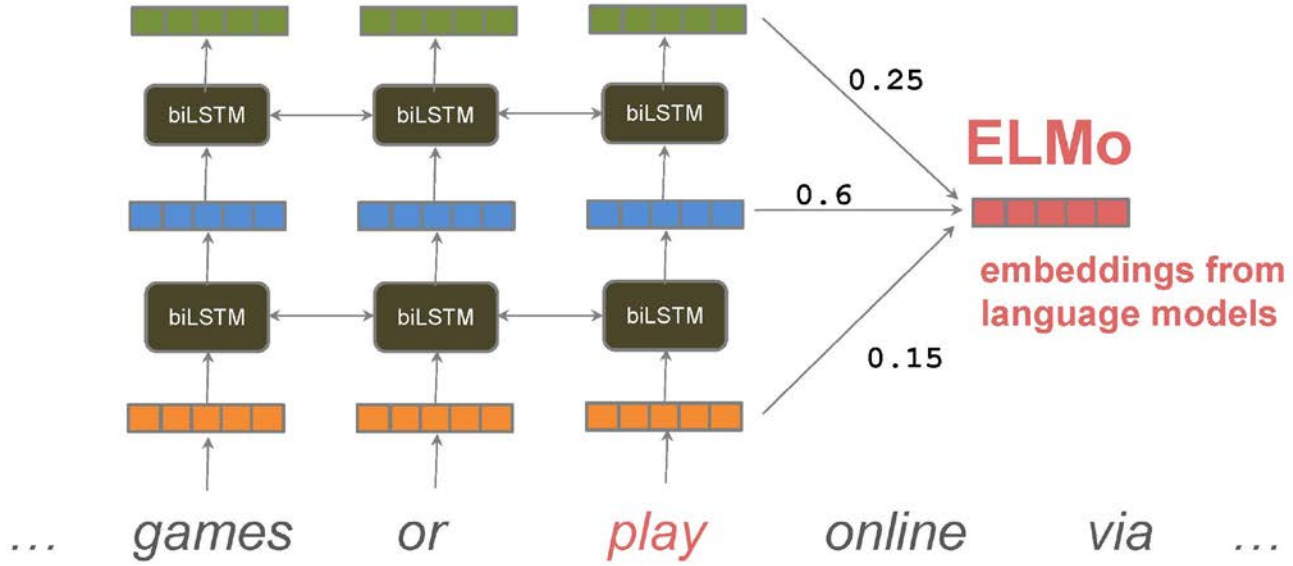
ELMO

Each layer in deep LSTM can generate a latent representation.

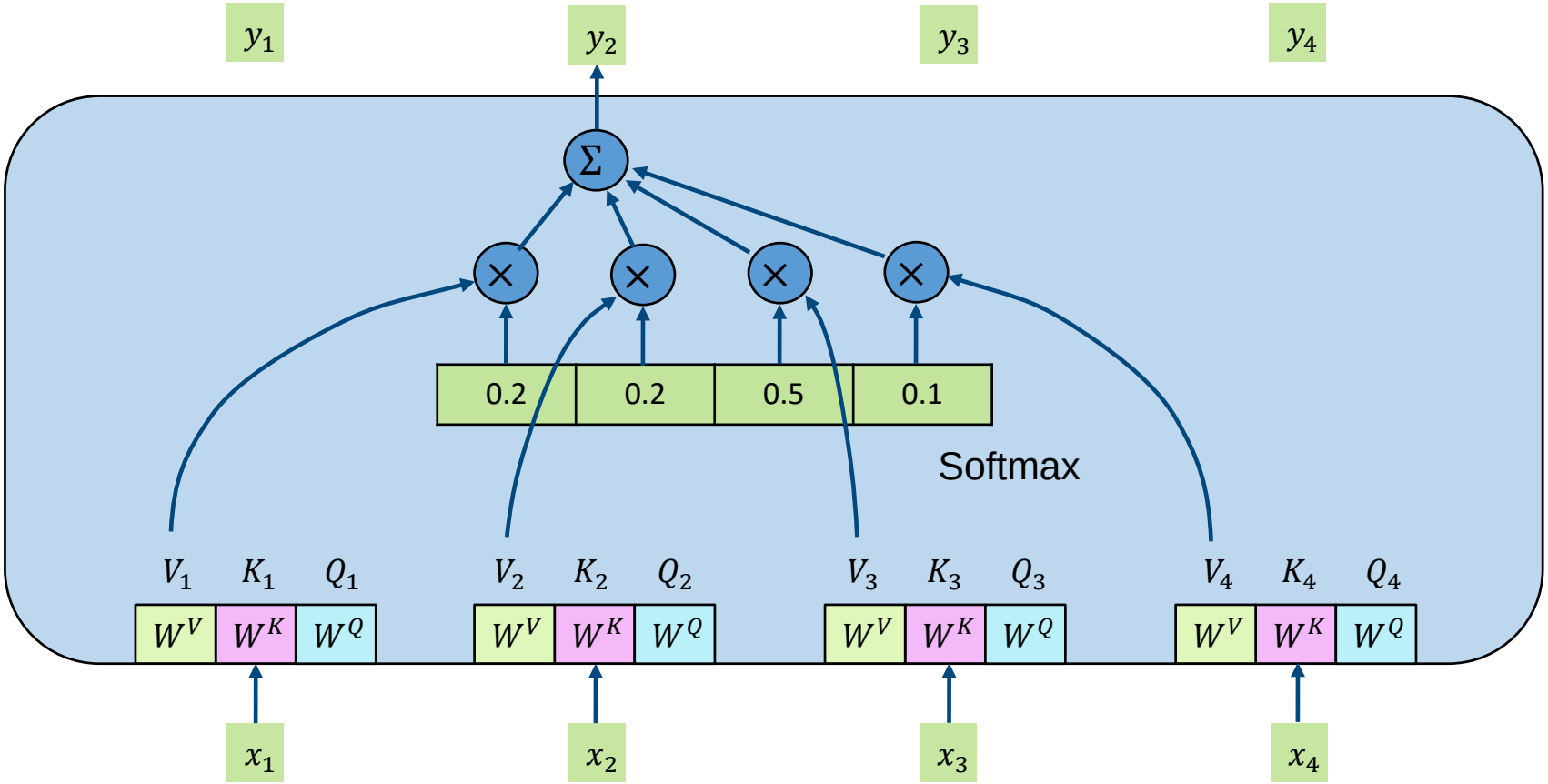


ELMO

use all layers of the language model



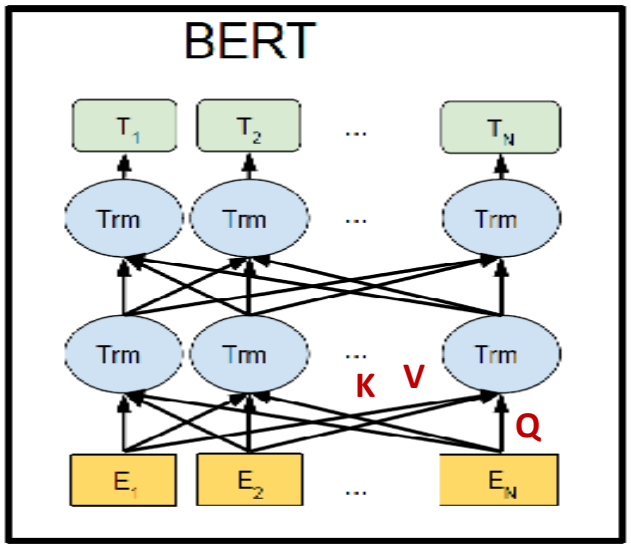
The Transformer – Self-Attention Layer

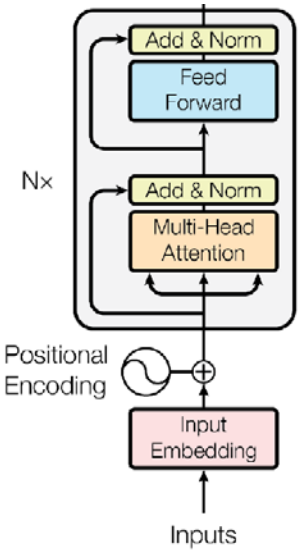


Sequence locations →

BERT

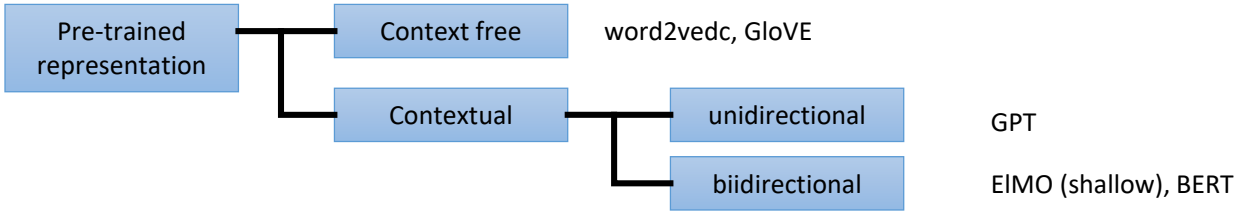
BERT is a bidirectional (Transformer encoder) model.

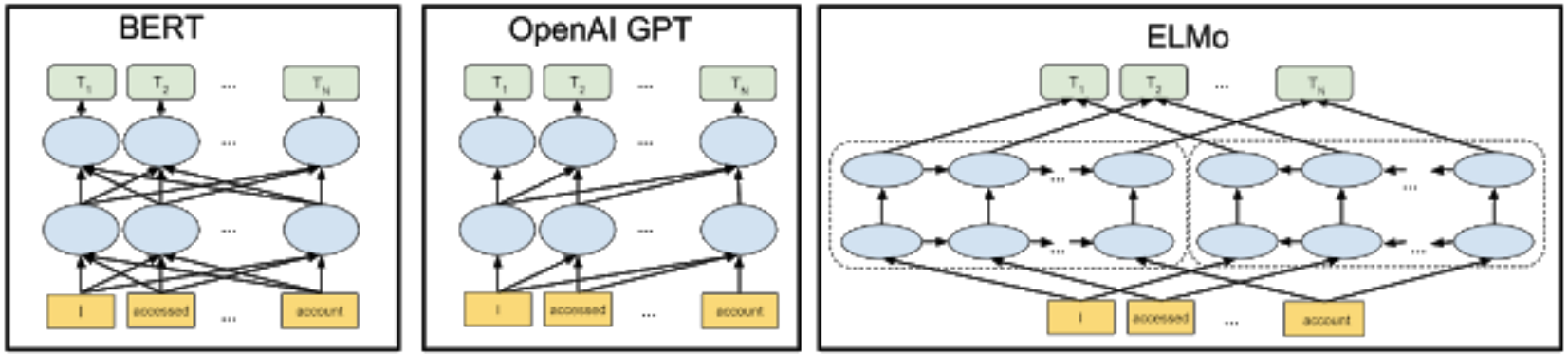




BERT

BERT is a deep bidirectional, unsupervised language representation, pre-trained using only a plain text corpus

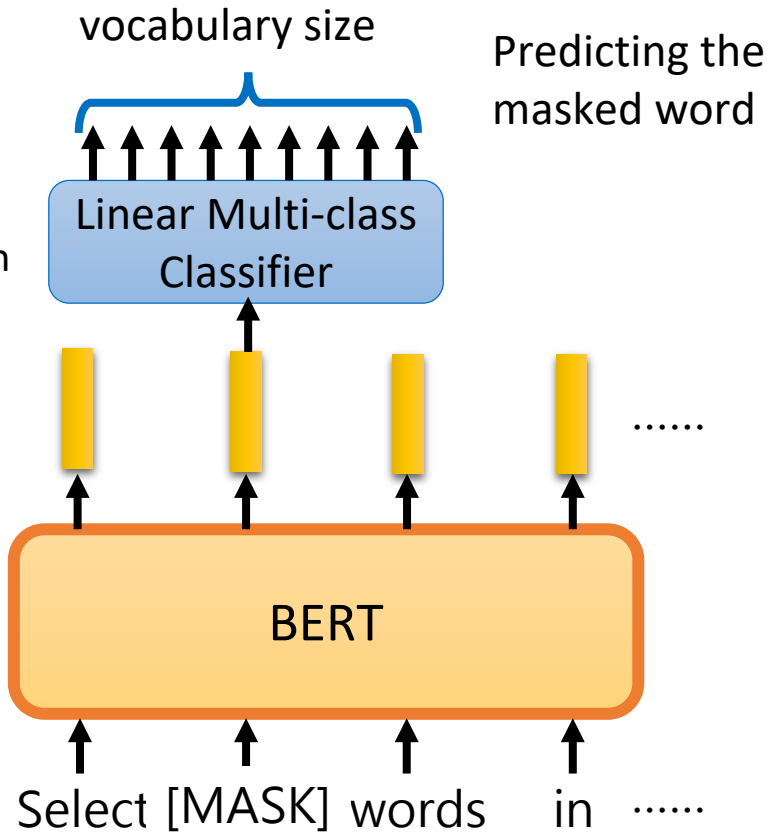




Training of BERT

Approach 1: Masked Language Modeling (MLM)

- Select 15% random words in the input.
- Replace each by a <MASK> token
- On the output re-predict the original words
- There is a bit more detail, see detail in the paper



Training of BERT

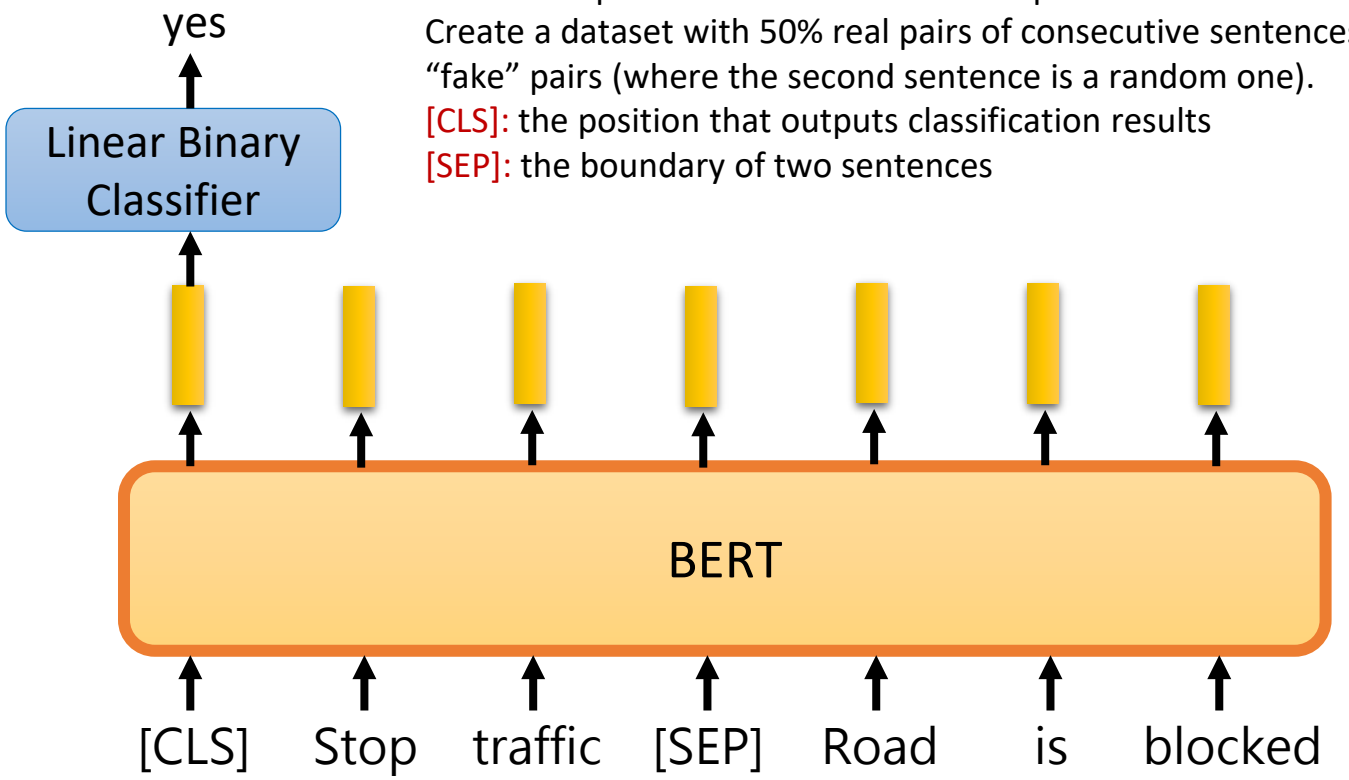
Approach 2: Next Sentence Prediction

Given a corpus of consecutive sentence pairs

Create a dataset with 50% real pairs of consecutive sentences, and 50% "fake" pairs (where the second sentence is a random one).

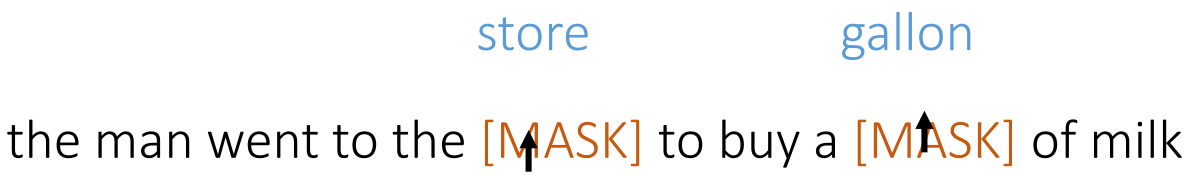
[CLS]: the position that outputs classification results

[SEP]: the boundary of two sentences



Masked LM

- **Solution**: Mask out k% of the input words, and then predict the masked words
  - Typically k = 15%



Too little masking: Too expensive to train

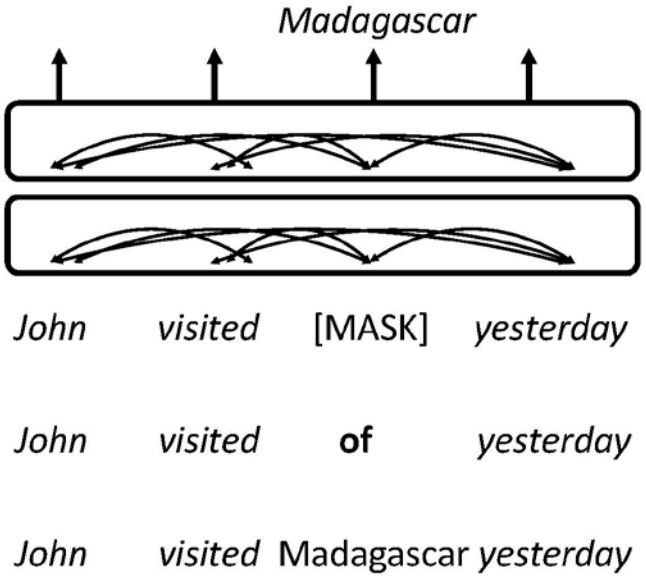
Too much masking: Not enough context

Slide from Jacob Delvin

Masked LM

- Problem: Mask token never seen at fine-tuning
- Solution: 15% of the words to predict, but don't replace with [MASK] 100% of the time. Instead:
- 80% of the time, replace with [MASK]

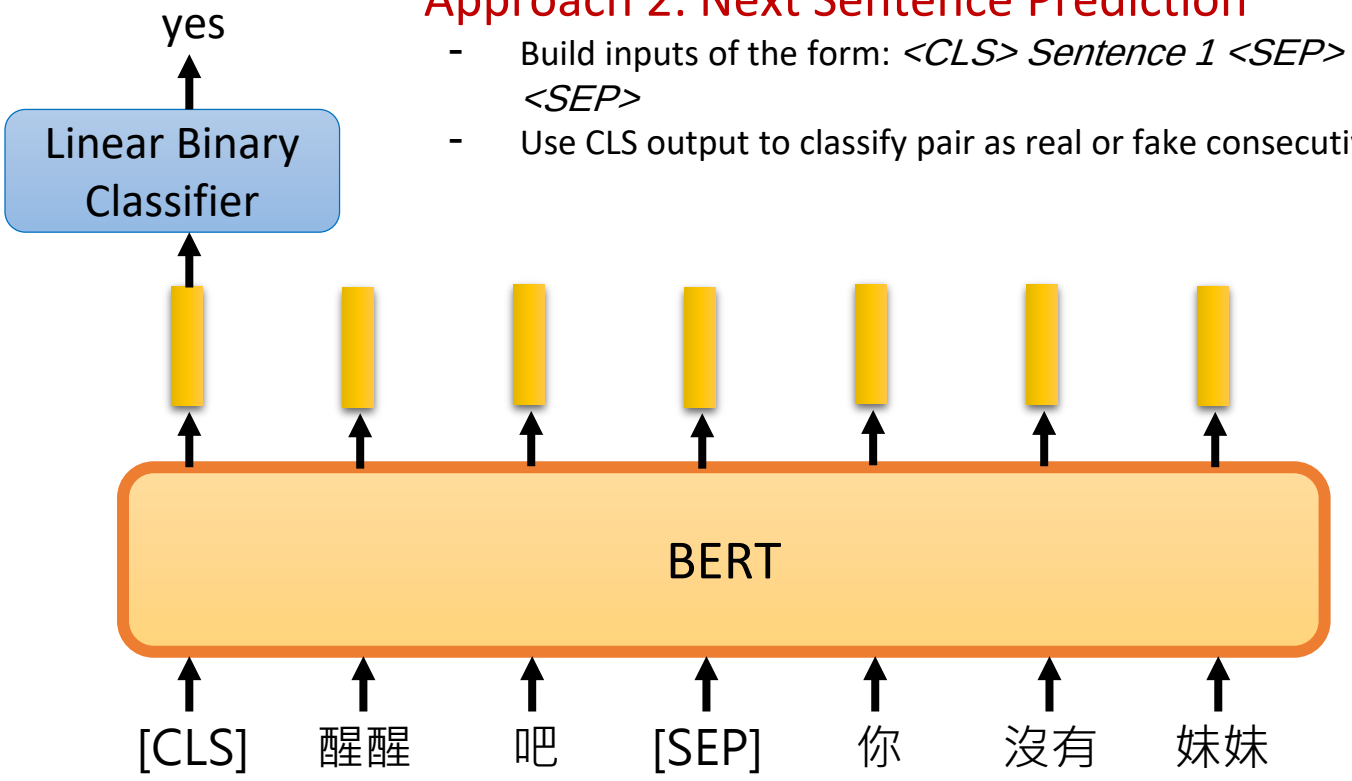
  went to the store → went to the [MASK]
- 10% of the time, replace random word
  went to the store → went to the running
- 10% of the time, keep same
  went to the store → went to the store



Training of BERT

Approach 2: Next Sentence Prediction

- Build inputs of the form: *<CLS> Sentence 1 <SEP> Sentence 2 <SEP>*
- Use CLS output to classify pair as real or fake consecutive pair



Next Sentence Prediction

- To learn *relationships* between sentences, predict whether Sentence B is actual sentence that follows Sentence A, or a random sentence

Sentence A = The man went to the store.
Sentence B = He bought a gallon of milk.
Label = IsNextSentence

Sentence A = The man went to the store.
Sentence B = Penguins are flightless.
Label = NotNextSentence

This is found to be unimportant as can be removed as in RoBERTa

Slide from Jacob Delvin

BERT

BERT is trained with two types of loss:

$$\text{Input} = \texttt{[CLS] the man went to [MASK] store [SEP]}$$
$$\texttt{he bought a gallon [MASK] milk [SEP]}$$
$$\text{Label} = \texttt{IsNext}$$

$$\text{Input} = \texttt{[CLS] the man [MASK] to the store [SEP]}$$
$$\texttt{penguin [MASK] are flight ##less birds [SEP]}$$
$$\text{Label} = \texttt{NotNext}$$

During pretraining, the model must learn to recover what has been MASKed and predict sentence consecutiveness.

What is a... contextual word embedding

Transformer-based models start with a static word embedding (input of layer 1), which is "*contextualized*" by going through stacked layers of self-attention and transformation.

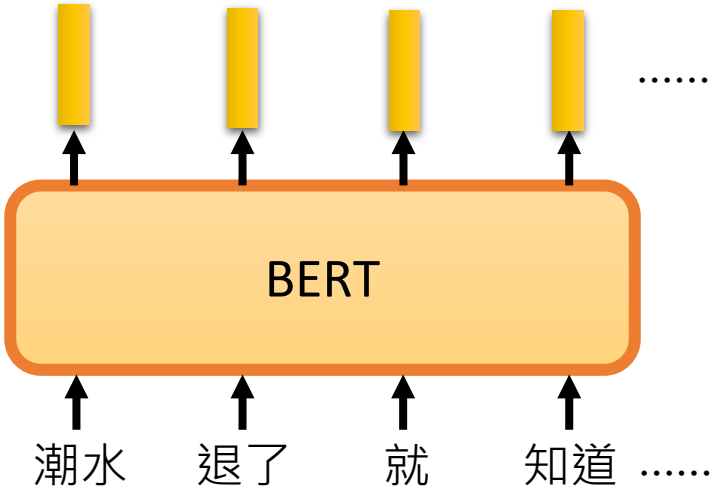
I run my house's kitchen, so if we run out of something, I do a quick run to the market.

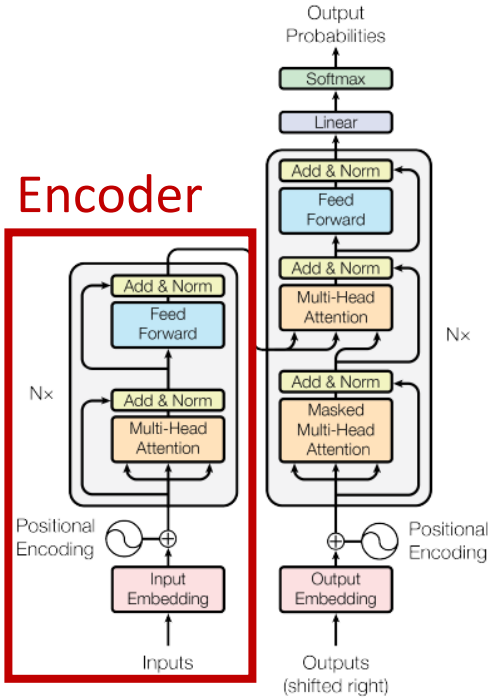
In a Transformer, the model will learn to contextualize and differentiate each instance of "run". In comparison, word2vec has a single vector for "run".

Bidirectional Encoder Representations from Transformers (BERT)

- BERT = Encoder of Transformer

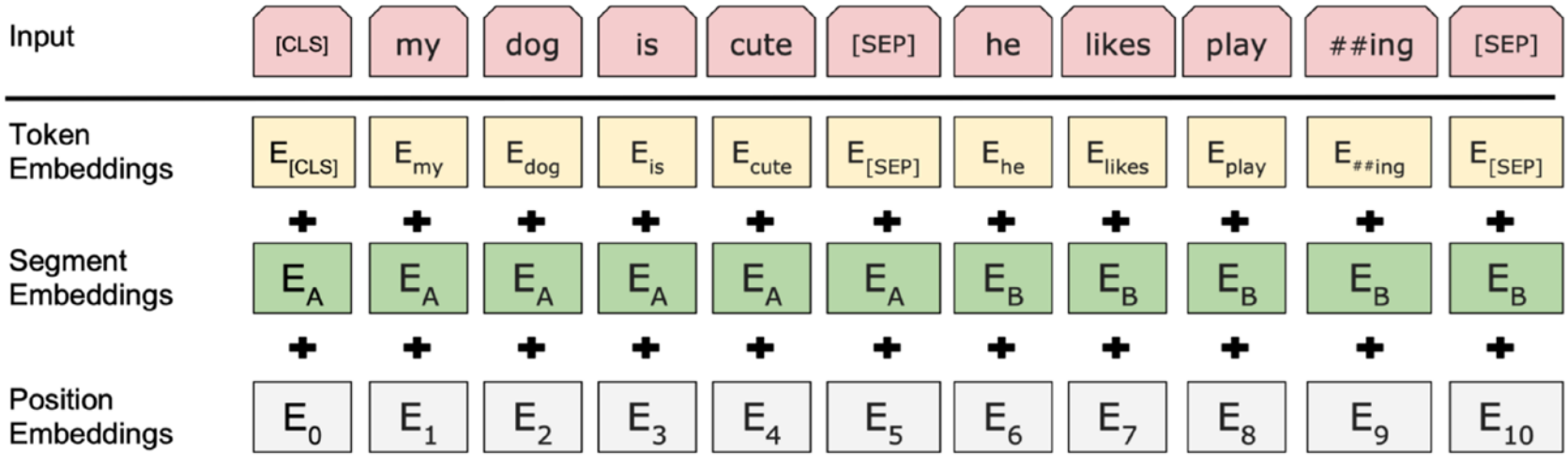
Learned from a large amount of text without annotation





BERT

- Multi-layer self-attention (Transformer)
- Input: a sentence or a pair of sentences with a separator and subword representation



Tokenization Challenges

- Problem: A fixed vocabulary does not account for language evolution.

  "Boeing's new Starliner is built to carry astronauts."

- **Solution 1:** Any word not in the vocabulary is replaced by a special "unknown" token.

>> toks = ["Boeing", " 's", "new", "UNK", "is", "built", "to", "carry", "astronauts", "."]

  Limitation: All unknown words are represented by the same word vector to the model, limiting model understanding.

- **Solution 2**: Increase vocabulary size. NLP models saw vocabulary size grow from 50k to around 200k (until 2017).

>> toks = ["Boeing", " 's", "new", "Starliner", "is", "built", "to", "carry", "astronauts", "."]

Limitation:

- Larger vocabulary means larger model size.
- Model cannot learn good representations for words it sees a few times.
- UNK can still occur (vocab is not infinite)

Sub-word Tokenization

BERT uses Word Piece tokenization

Solution 3: Allow for words to be broken down into "pieces" (e.g. syllables) if they are not present in the vocabulary.

>> toks = ["boeing", " 's", "new", "star", "##liner", "is", "built", "to", "carry", "astronauts", "."] # (output of the BERT uncased tokenizer)

1. Initialize with tokens for all characters
2. While vocabulary size is below the target size:
    1. Build a language model over the corpus (e.g., unigram language model)
    2. Merge pieces that lead to highest improvement in language model perplexity

Need to choose a language model that will make the process tractable. Often a unigram language model

WordPiece

- BERT uses a variant of the wordpiece model
- (Relatively) common words are in the vocabulary:

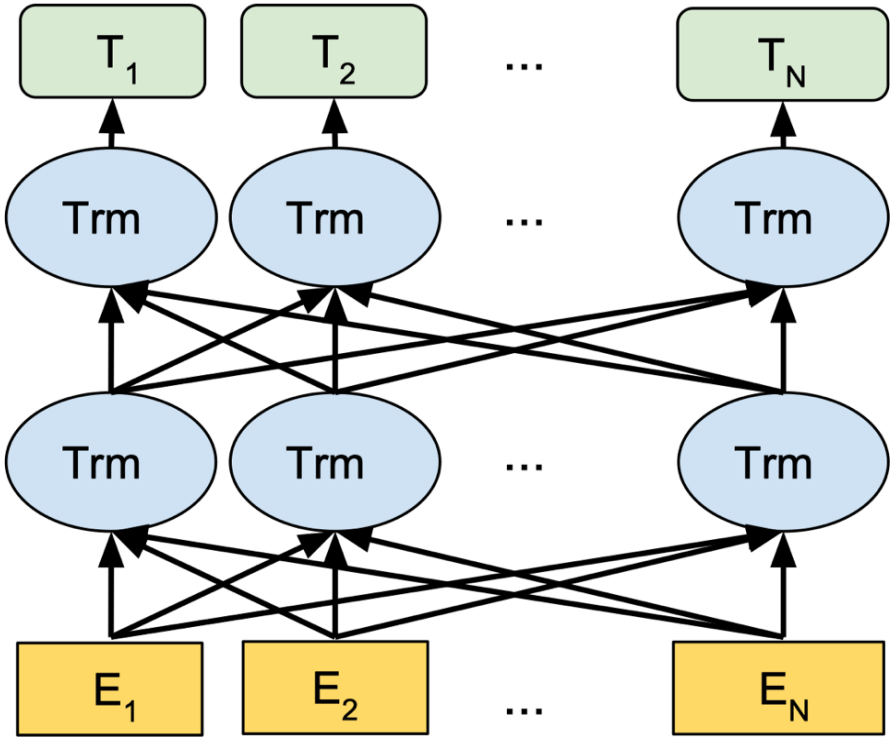
  at, fairfax, 1910s

- Other words are built from wordpieces:

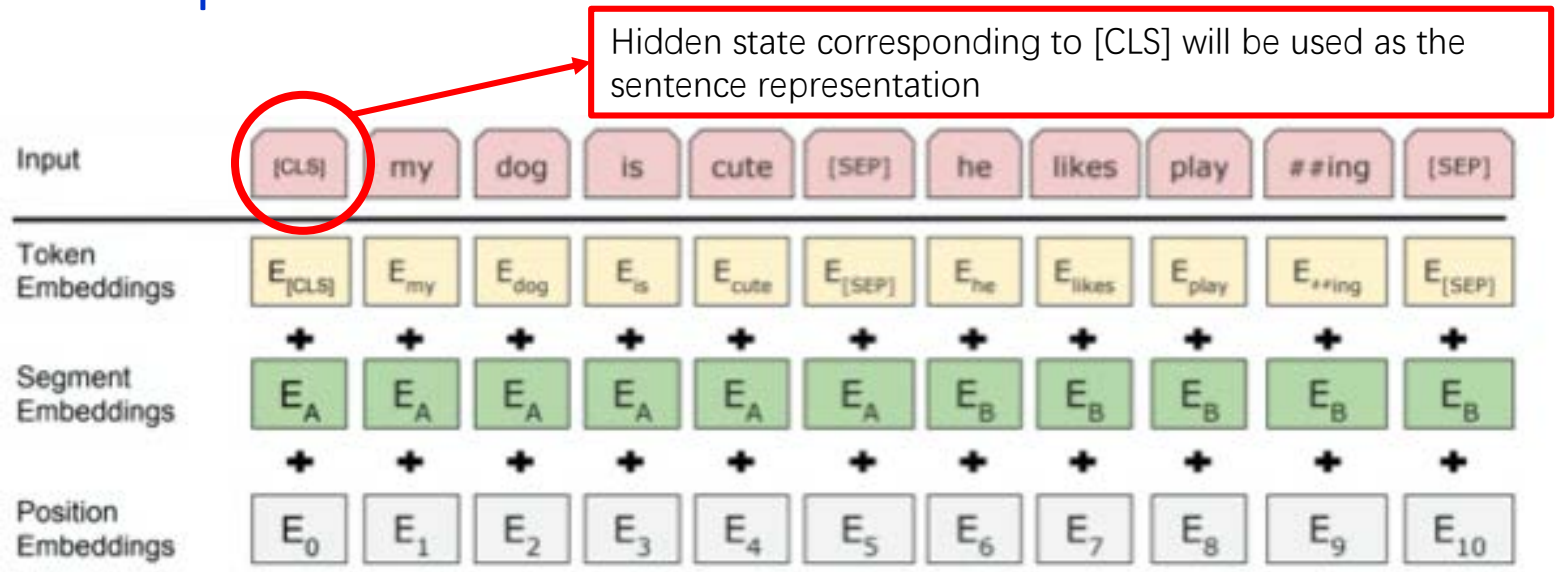
  hypatia = h ##yp ##ati ##a

- Wordpiece Model:
  - Given a training corpus and a number of desired tokens *D*, select *D* wordpieces such that the resulting corpus is minimal in the number of wordpieces when segmented according to the chosen wordpiece model.

BERT: multi-layer self-attention (Transformer)



Input Representation



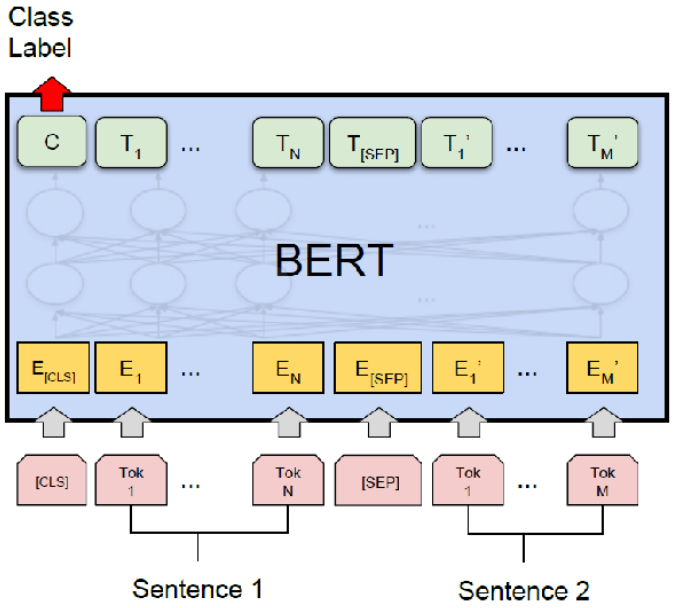
- Use 30,000 WordPiece vocabulary on input.
- Each token is sum of three embeddings
- Single sequence is much more efficient.

Slide from Jacob Delvin

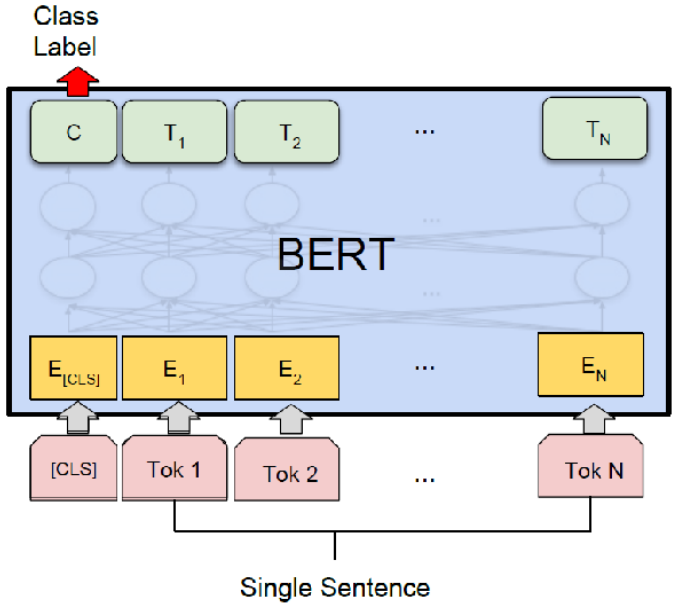
Using BERT

- Use the pre-trained model as the first “layer” of your final model
- Train with fine-tuning using your supervised data
- Fine-tuning recipe: 1-3 epochs, batch size 2-32, learning rate 2e-5 - 5e-5

BERT Task specialization

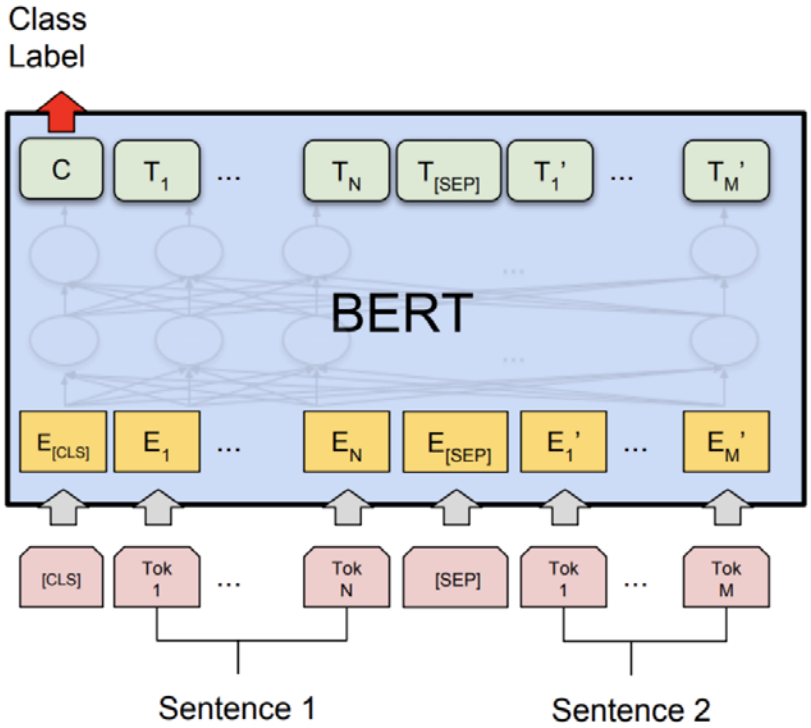


(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

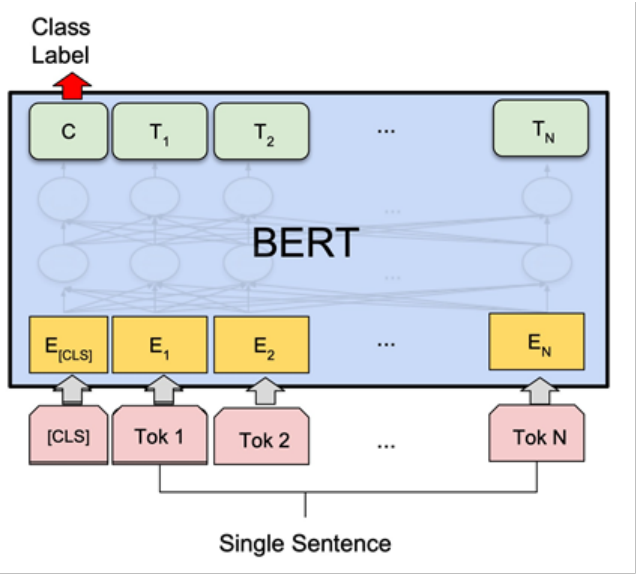


(b) Single Sentence Classification Tasks: SST-2, CoLA

BERT for Classification Task

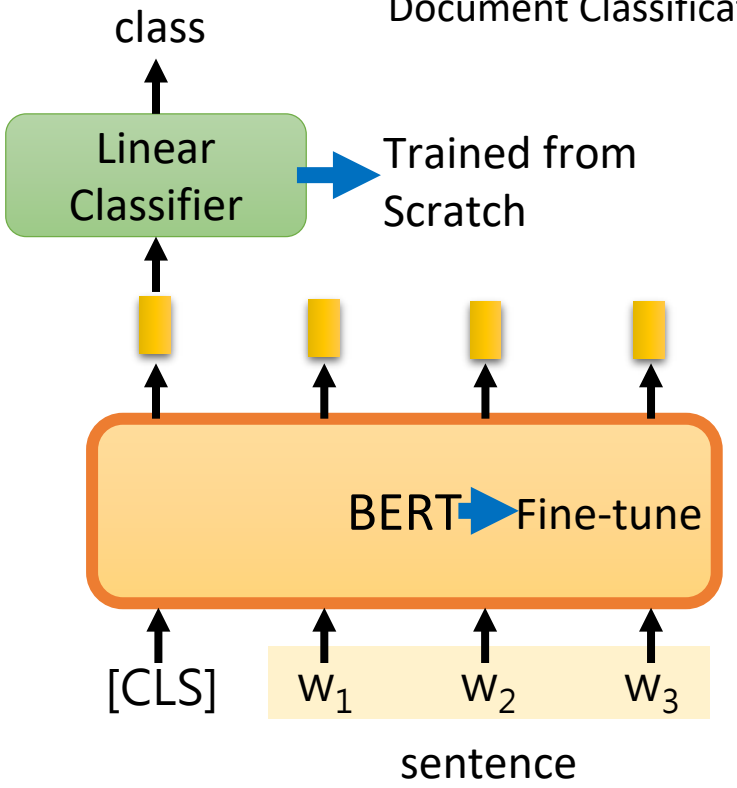


BERT for classification

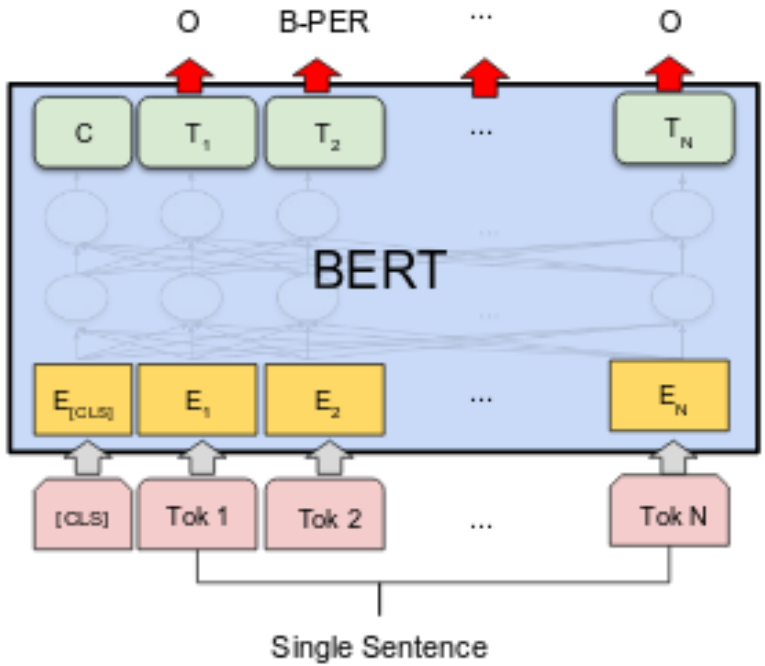


Input: single sentence,
output: class

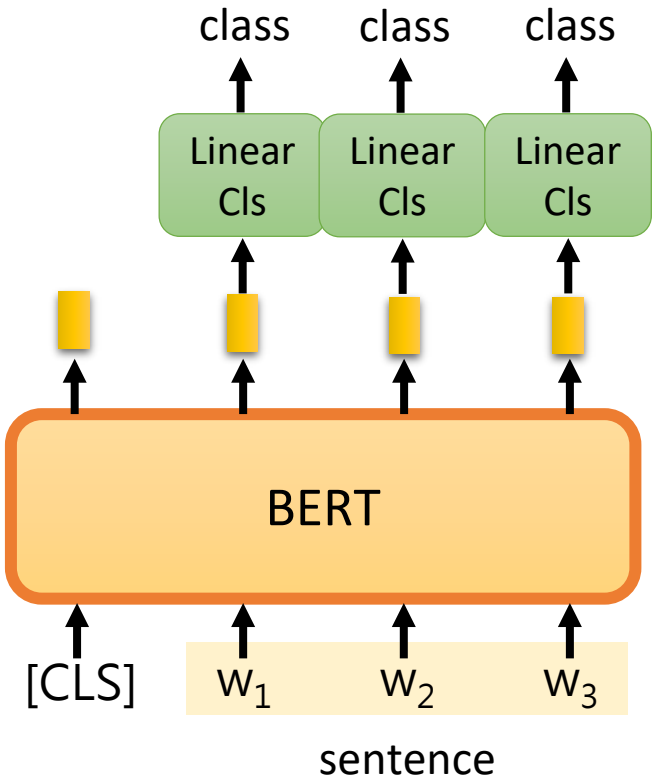
Example:
Sentiment analysis
Document Classification



BERT for tagging tasks

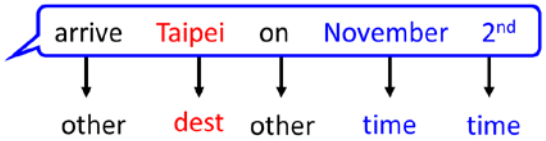


BERT for Tagging



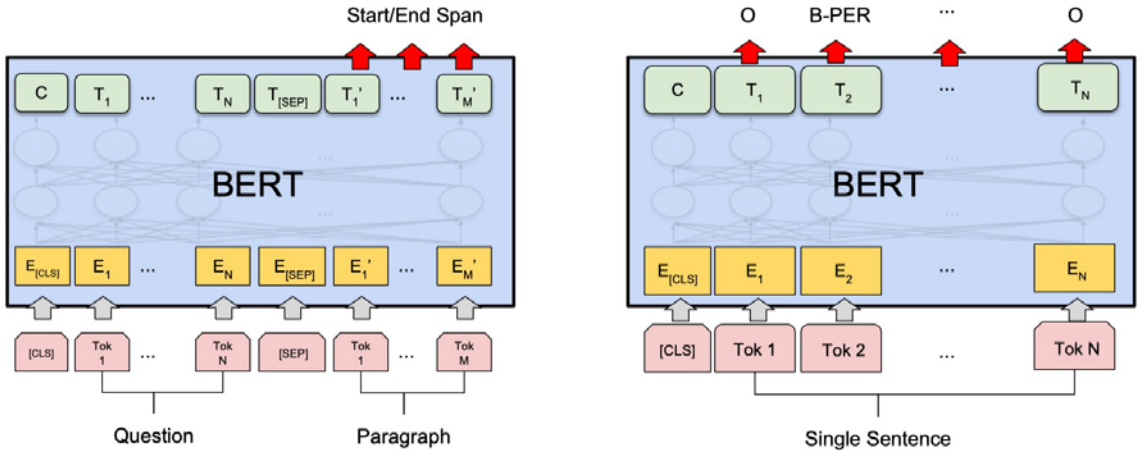
Input: single sentence, output: class of each word

Example: Slot filling

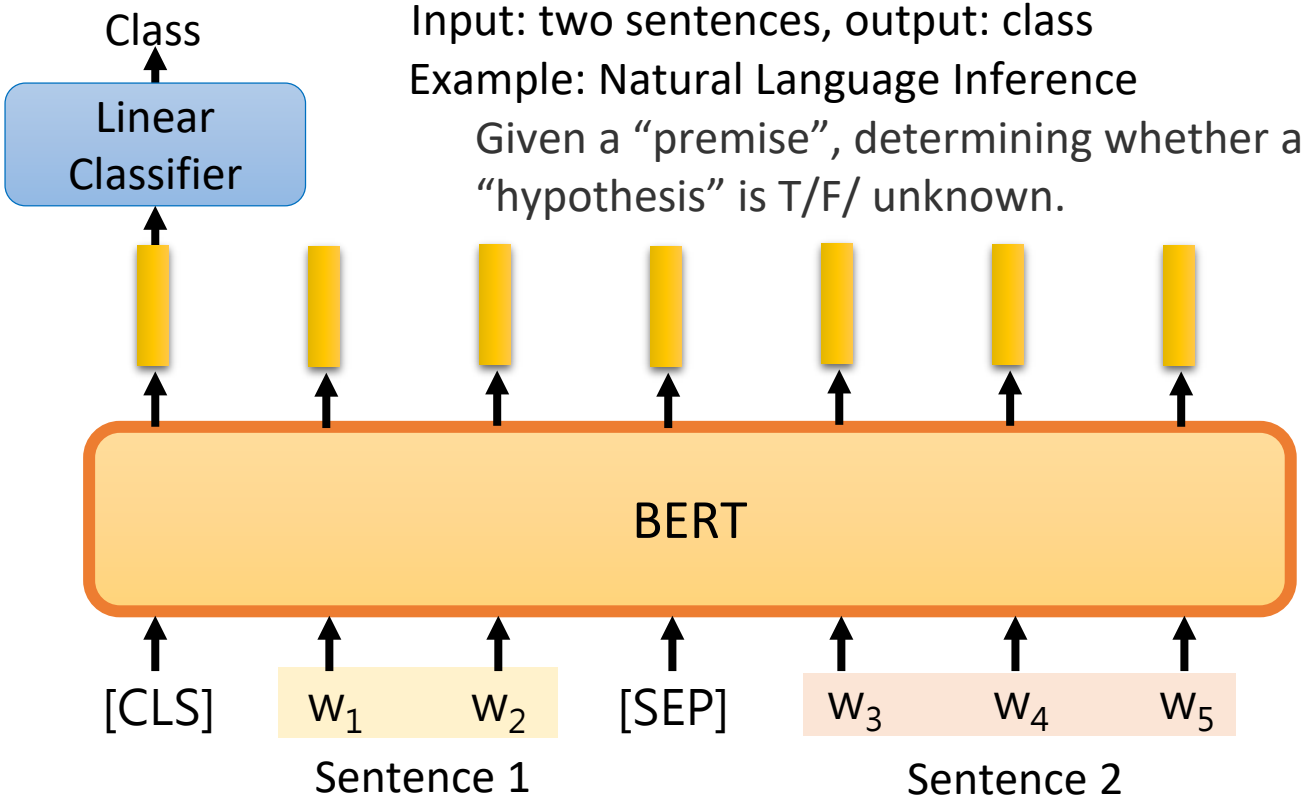


Tagging with BERT

- Can do for a single sentence or a pair
- Tag each word piece
- Example tasks: span-based question answering, name-entity recognition, POS tagging



How to use BERT – Case 3



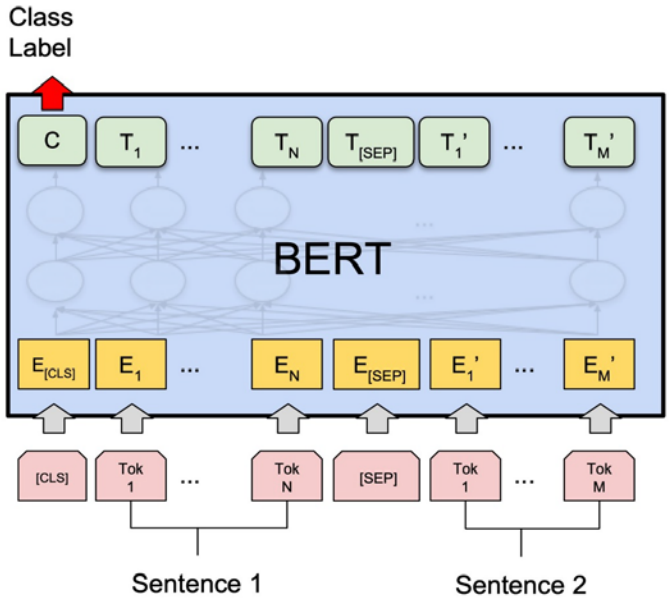
Input: two sentences, output: class

Example: Natural Language Inference

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

Sentence-pair Classification with BERT

- Feed both sentences, and CLS token used for classification
- Example tasks:
  - Textual entailment
  - Question paraphrase detection
  - Question-answering pair classification
  - Semantic textual similarity
  - Multiple choice question answering



Question Answering

SQuAD 1.0: Stanford Question Answering Dataset

Given a paragraph P, a question Q, find the answer as a subset of P (start and end character)

The 1973 oil crisis began in October 1973 when the members of the Organization of Arab Petroleum Exporting Countries (OAPEC, consisting of the Arab members of OPEC plus Egypt and Syria) proclaimed an oil embargo. By the end of the embargo in March 1974, the price of oil had risen from US$3 per barrel to nearly $12 globally; US prices were significantly higher. The embargo caused an oil crisis, or "shock", with many short- and long-term effects on global politics and the global economy. It was later called the "first oil shock", followed by the 1979 oil crisis, termed the "second oil shock."

When did the 1973 oil crisis begin?
Ground Truth Answers: October 1973 | October 1973 | October 1973 | October | 1973

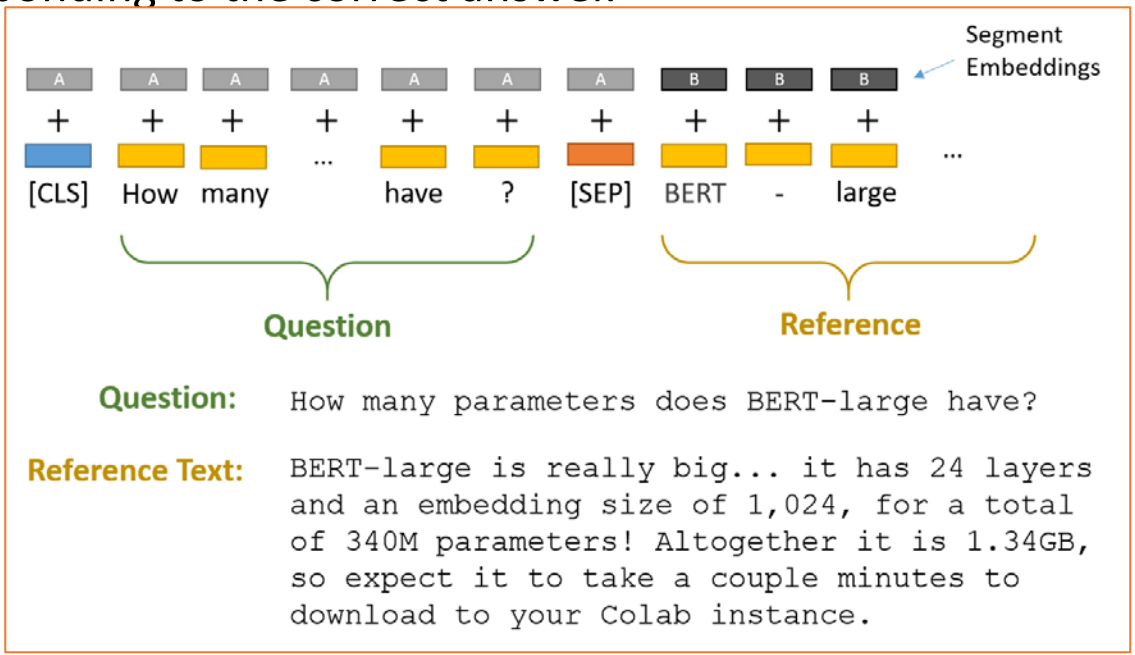
When was the second oil crisis?
Ground Truth Answers: 1979 | 1979 | 1979 | 1979 | 1979

What was another term used for the oil crisis?
Ground Truth Answers: first oil shock | shock | shock | first oil shock | shock

Question Answering with BERT

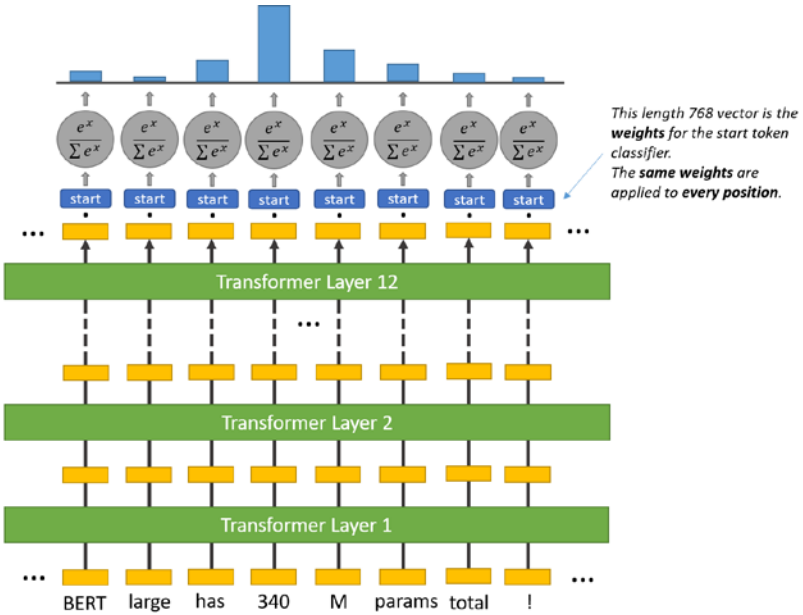
- Stanford Question Answering Dataset (SQuAD).
- Given a question, and *a passage of text containing the answer*, highlight the "span" of text corresponding to the correct answer.

Input Format
Pack the question and the reference text into the input separated by <SEP>
Segment Embedding

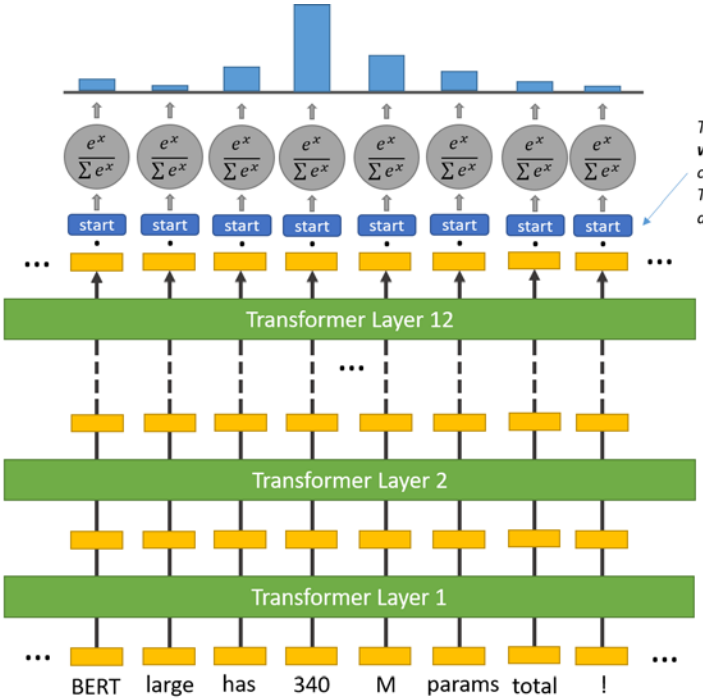


QA: Start and End Token Classification

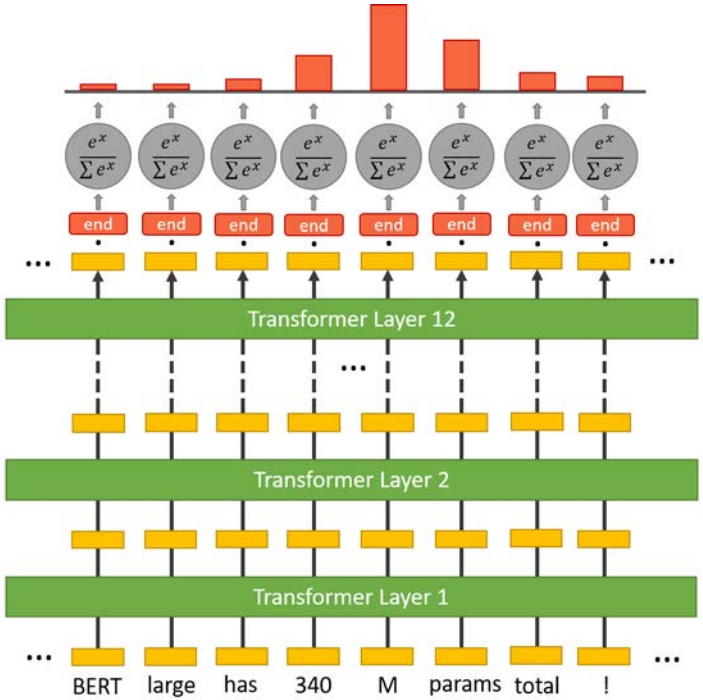
- Highlight a “span” of text containing the answer: Predict which token marks the start of the answer, and which token the end.
- Feed the final embedding of each token into the start token classifier. The classifier only has a single set of weights (represented by the blue “start” rectangle in the above illustration).
- After taking the dot product between the output embeddings and the ‘start’ weights, apply the softmax activation to produce a probability distribution over all of the words. Pick the word with the highest probability of being the start token.
- Repeat this process for the end token–we have a separate weight vector this.



Start and End Token Classification



*This length 768 vector is the **weights** for the start token classifier.*
*The **same weights** are applied to **every position**.*



BERT Performance

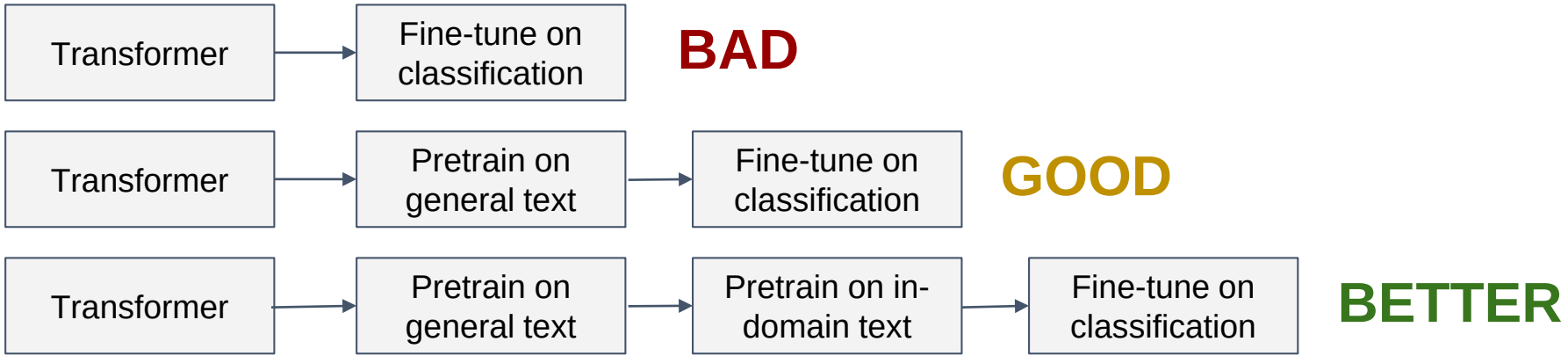
SQuAD

System	Dev		Test	
	EM	F1	EM	F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	86.6
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Training a classifier with Transformers

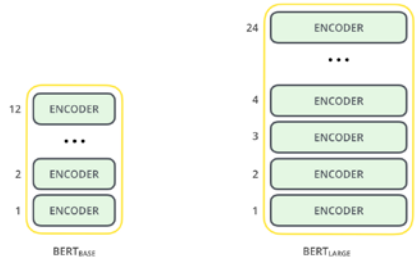
Practical tip:

When training a classifier, pretrain the model on data as close to in-domain data.
For in detail analysis: ULMFit paper 2018



Model Architecture

- BERT BASE
  - 12 layers, 768-dim per word-piece token
  - 12 heads.
  - Total parameters = 110M
- BERT LARGE
  - 24 layers, 1024-dim per word-piece token
  - 16 heads.
  - Total parameters = 340M



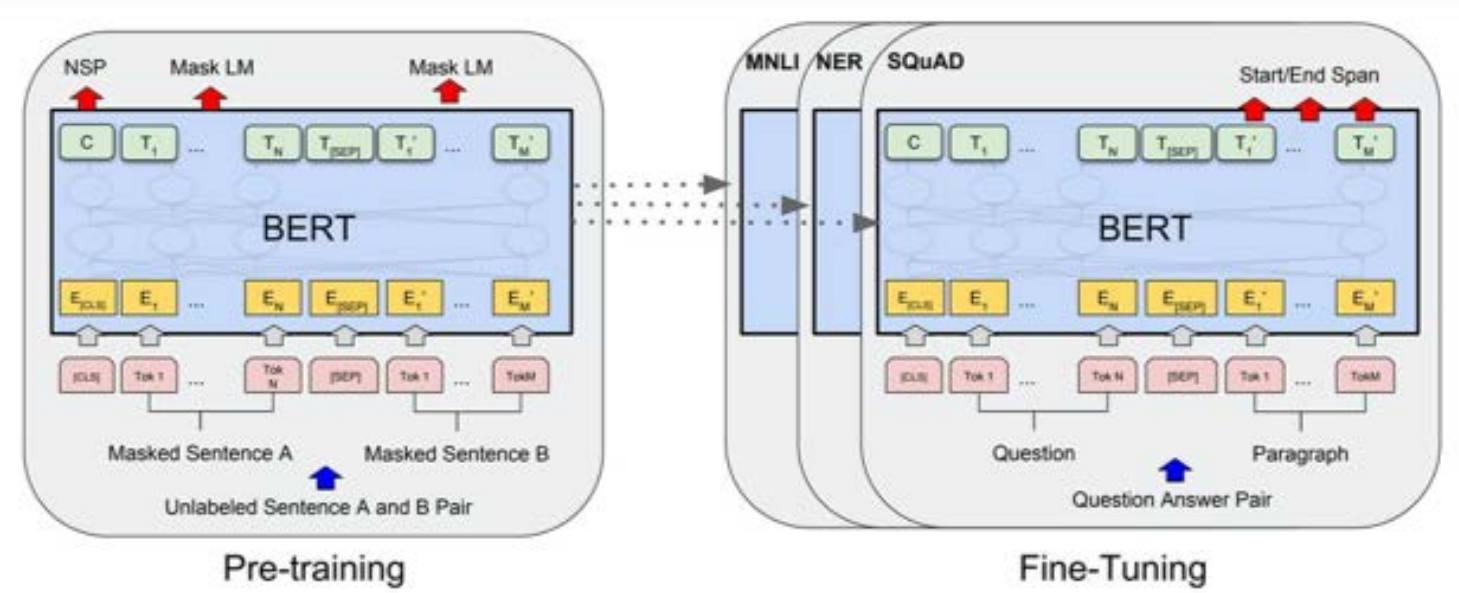
BERT is basically a trained Transformer Encoder stack.

Model Details

- Data: Wikipedia (2.5B words) + BookCorpus (800M words)
- Batch Size: 131,072 words (1024 sequences * 128 length or 256 sequences * 512 length)
- Training Time: 1M steps (~40 epochs)
- Optimizer: AdamW, 1e-4 learning rate, linear decay
- BERT-Base: 12-layer, 768-hidden, 12-head
- BERT-Large: 24-layer, 1024-hidden, 16-head
- Trained on 4x4 or 8x8 TPU slice for 4 days

Slide from Jacob Delvin

Fine Tuning Procedure



Results

- Fine-tuned BERT outperformed previous state of the art on 11 NLP tasks
- Since then was applied to many more tasks with similar results
- The larger models perform better, but even the small BERT performs better than prior methods
- Variants quickly outperformed human performance on several tasks, including span-based question answering — but what does this mean is less clear
- Started an arms race (between industry labs) on bigger and bigger models

Where to get BERT?

- The Transformers library: https://github.com/huggingface/transformers

- Provides state-of-the-art implementation of many models, including BERT and RoBERTa

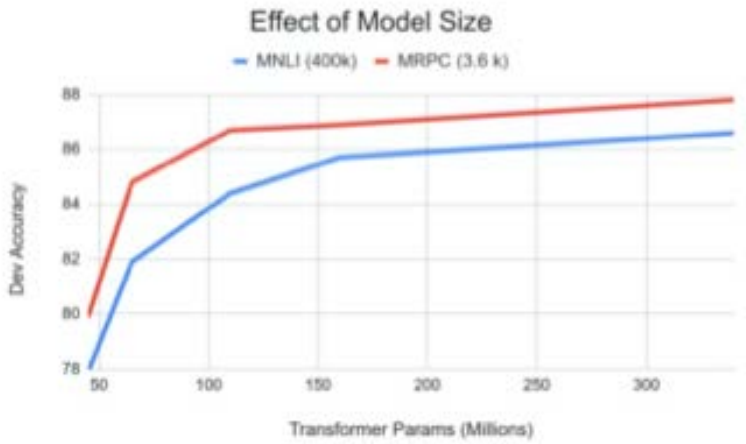
- Including pre-trained models

Hard to do with BERT

- BERT cannot generate text (at least not in an obvious way)

- Masked language models are intended to be used primarily for "analysis" tasks

Effects of Model Size



- Big models help *a lot*
- Going from 110M -> 340M params helps even on datasets with 3,600 labelled examples
- Improvements have *not* asymptoted

Slide from Jacob Delvin

References

- The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
- BERT