CS60075
Natural Language Processing
Autumn 2020

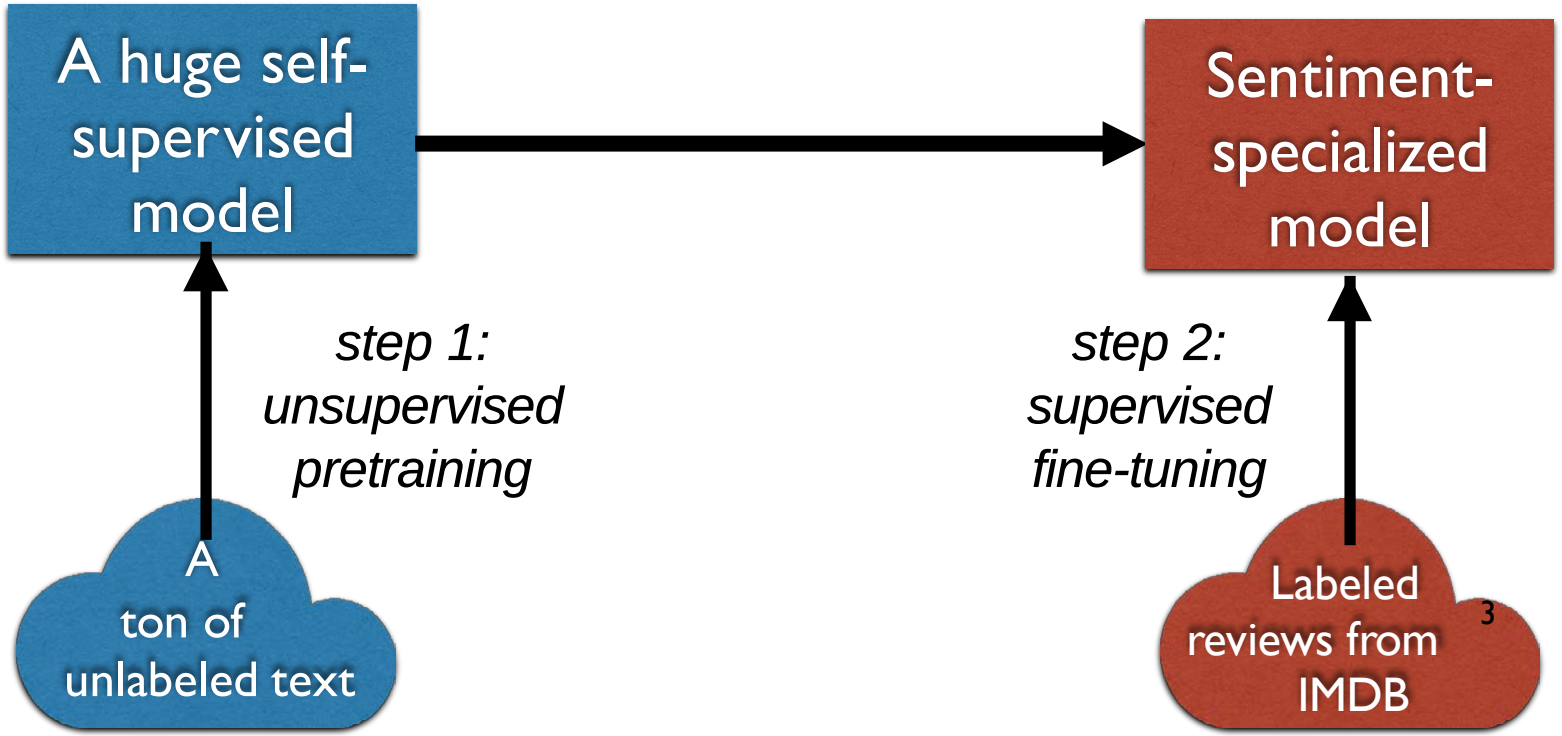
Module 9:

Part 1: BERT

5 November 2020

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
3

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 .

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

Nearest Neighbors

playing
game
games
played
players

VERB
NOUN
ADJ

plays
player
Play
football
multiplayer

Contextual Representations

- Train contextual representations on text corpus

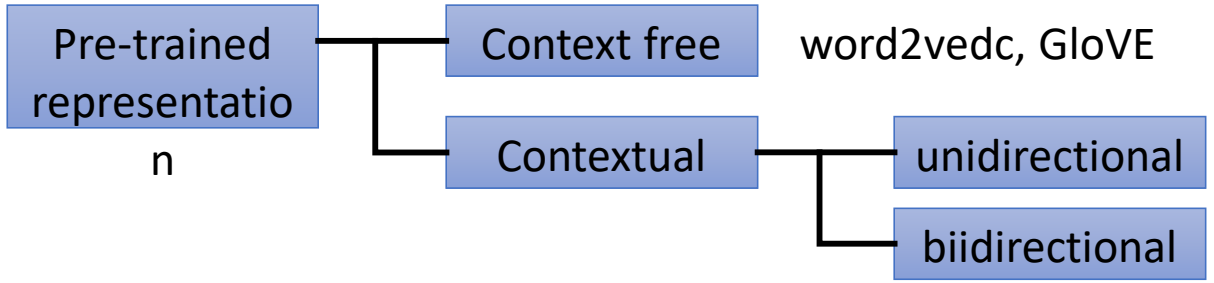
Problem with Previous Methods

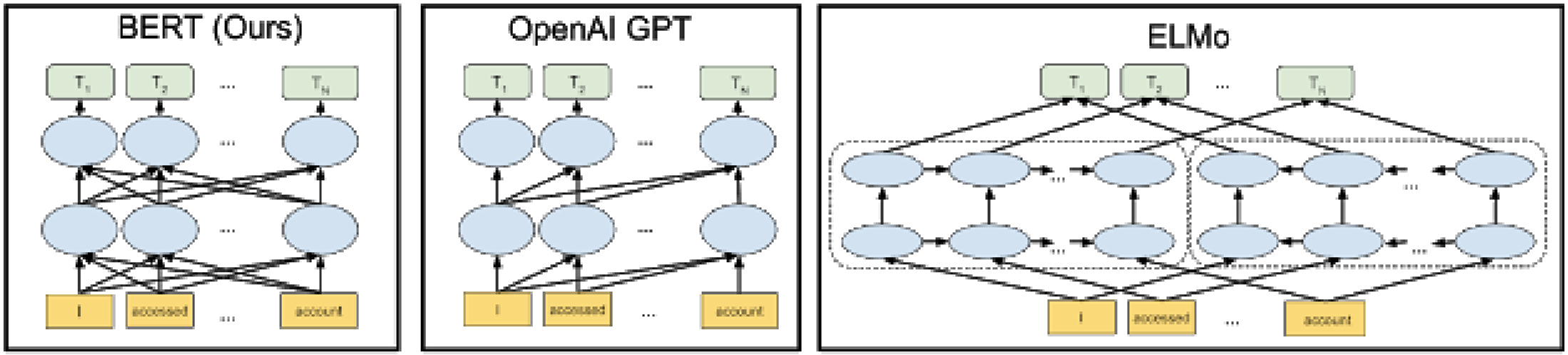
- **Problem**: Language models only use left context *or* right context, but language understanding is bidirectional.
- Why are LMs unidirectional?
  - Reason 1: Directionality is needed to generate a well-formed probability distribution.
  - Reason 2: Words can "see themselves" in a bidirectional encoder.

Slide from Jacob Delvin

What makes BERT different?

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

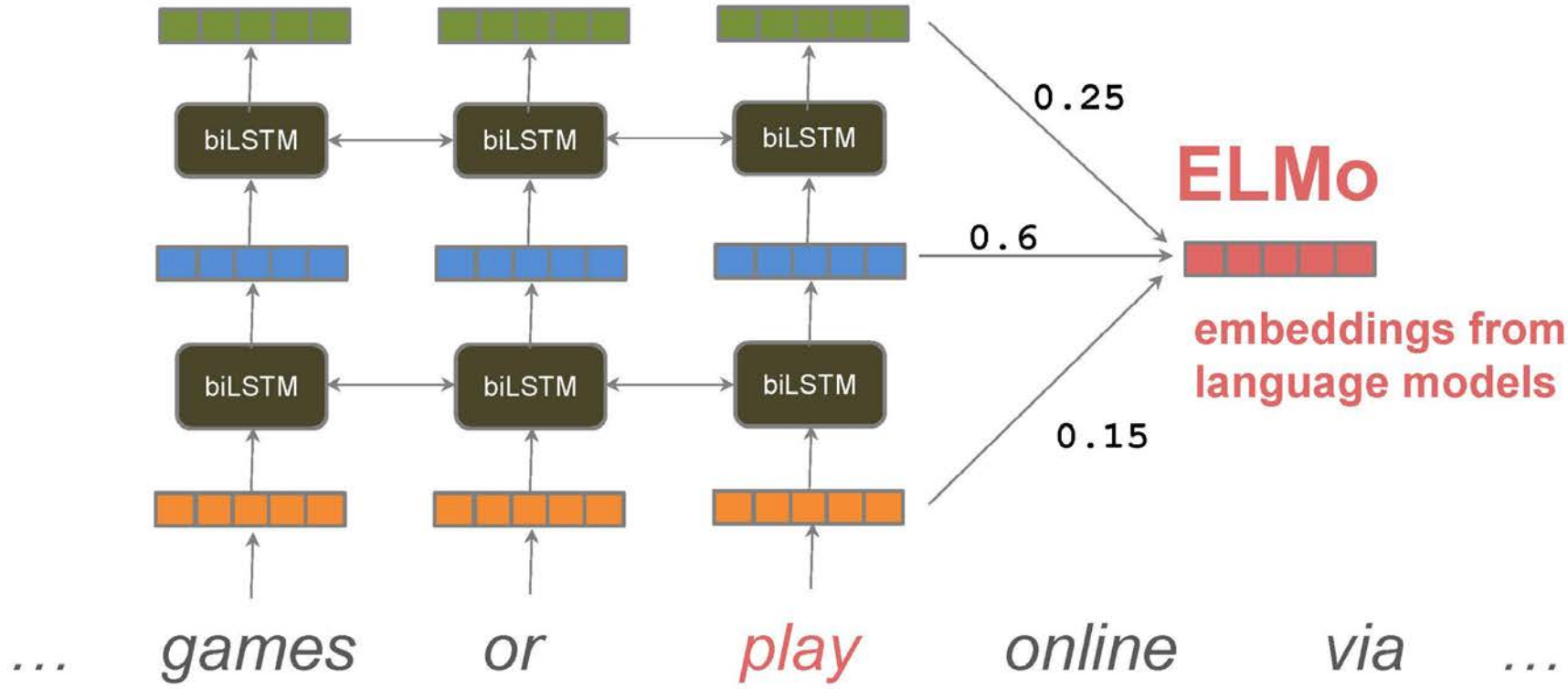




ELMo: Embeddings from Language Models
Pre-trained biLSTM for contextual embedding

ELMO

use all layers of the language model



ELMo: biLSTM for neural word embeddings

ELMo representations are:

- *Contextual*: The representation for each word depends on the entire context in which it is used.
- *Deep*: The word representations combine all layers of a deep pre-trained neural network.
- *Character based*: ELMo representations are purely character based, allowing the network to use morphological clues to form robust representations for out-of-vocabulary tokens unseen in training.

BERT

BERT is a language model (next word predictor) trained on a large dataset of natural language that is fine-tuned for particular tasks.

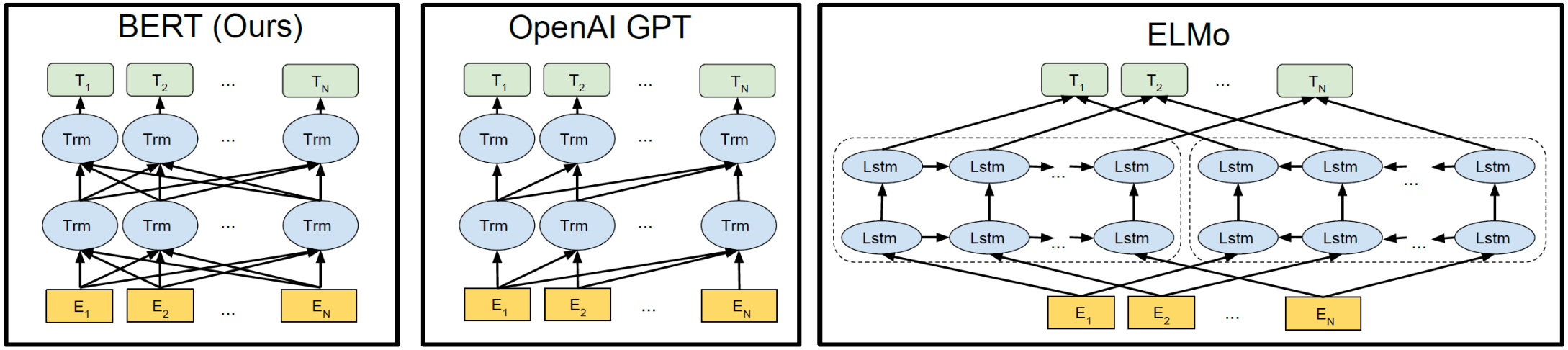
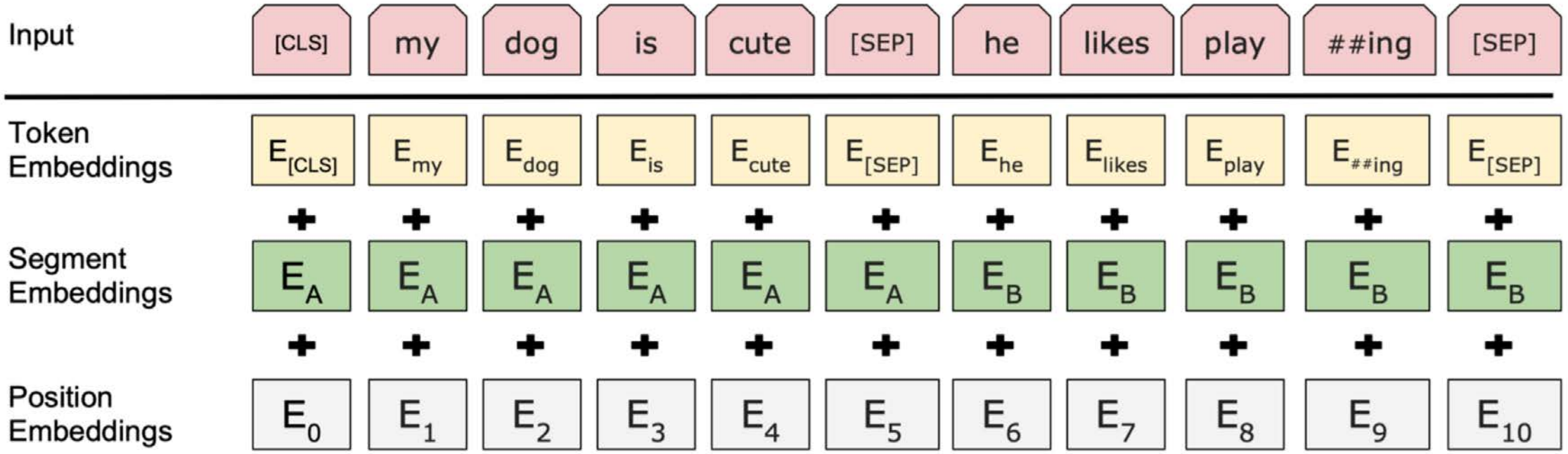


Figure 1: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks. Among three, only BERT representations are jointly conditioned on both left and right context in all layers.

But note that only GPT is a generative model.

BERT

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



Sub-word Tokenization

BERT uses Word Piece tokenization

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:

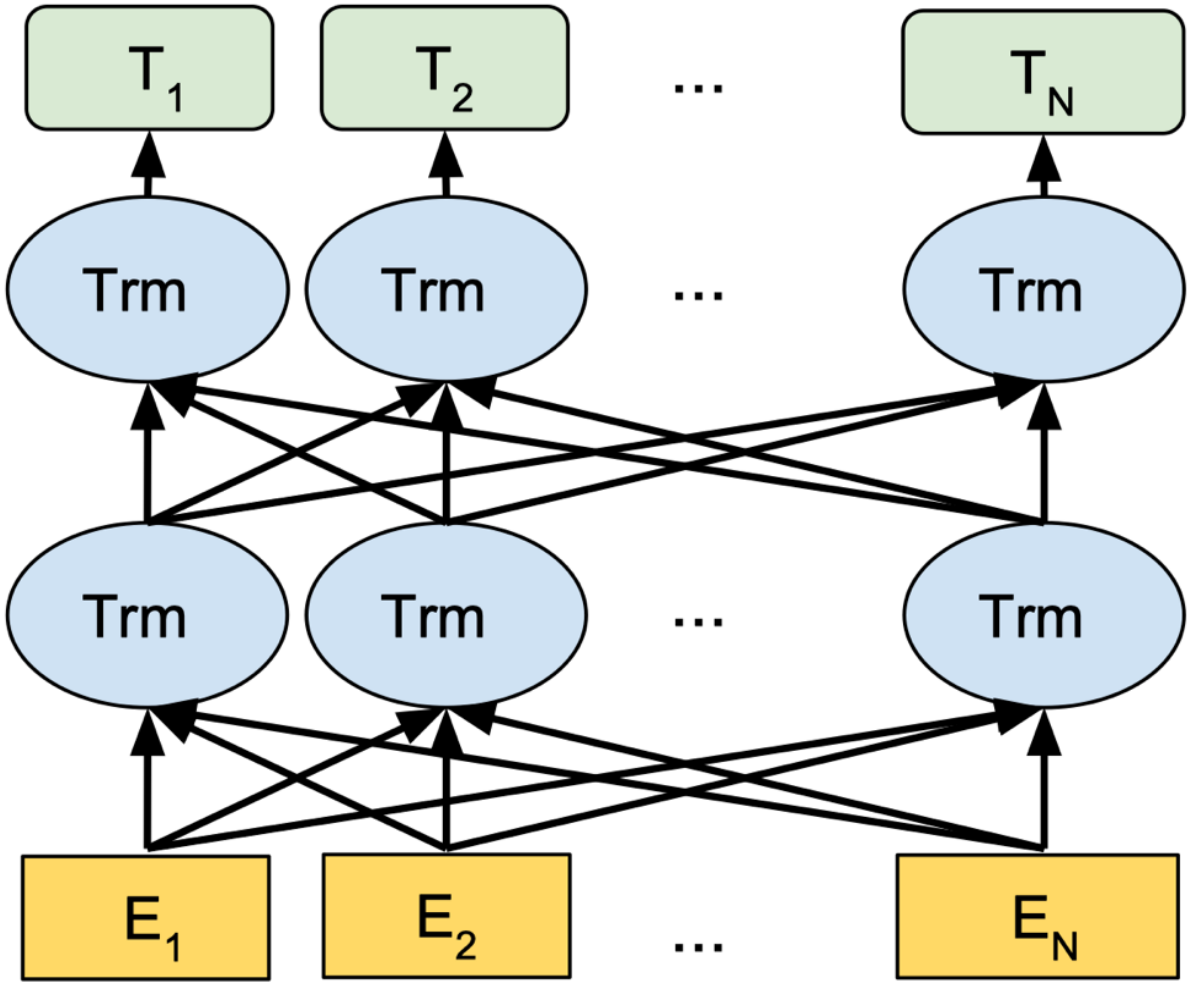
  at, fairfax, 1910s

- 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)



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

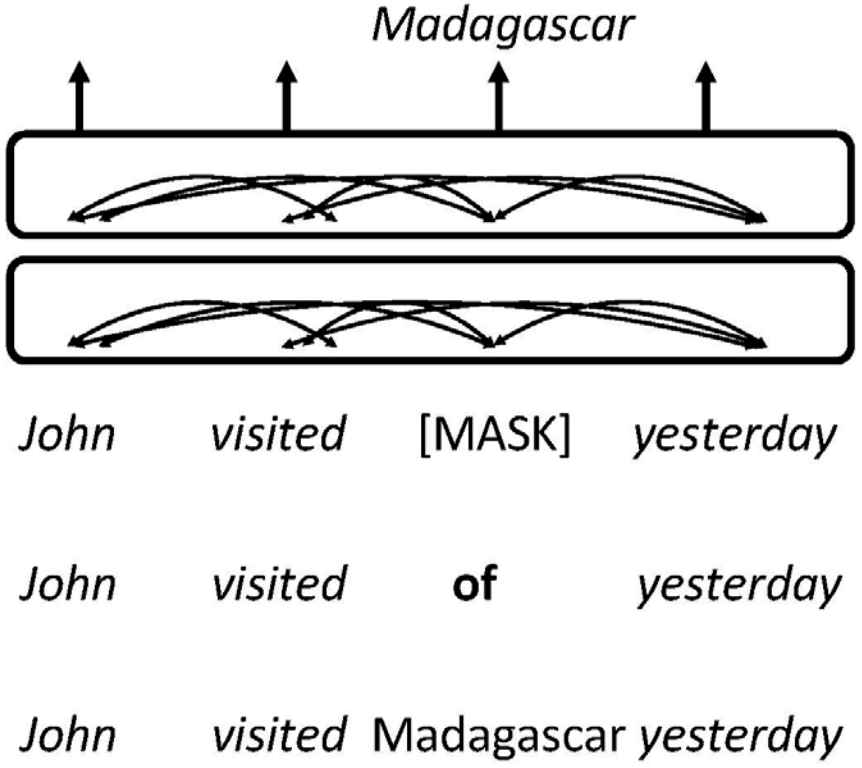
Masked LM

- 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



Slide from Jacob Delvin

Next Sentence Prediction

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

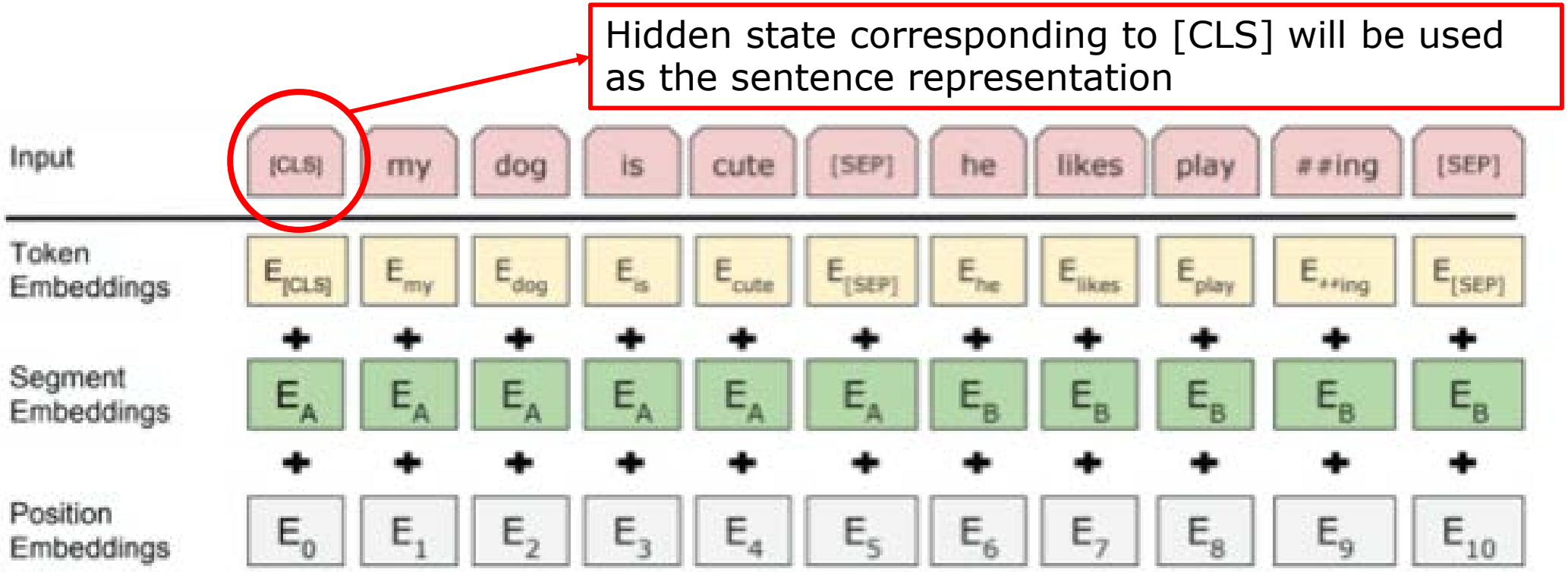
Sentence A = The man went to the store.
Sentence B = He bought a gallon of milk.
Label = IsNextSentence

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

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
- Large changes to weights in top layers (particularly in last layer to route the right information to [CLS])
- Smaller changes to weights lower down in the transformer
- Small learning rate and short fine-tuning schedule mean weights don’t change much

Two Step Development

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

Semi-supervised Learning Step

Model:

Dataset:

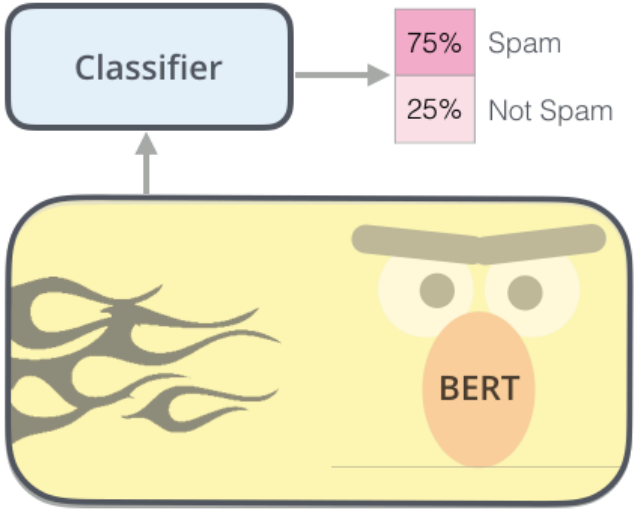
Objective:

Predict the masked word (langauge modeling)

2 - Supervised training on a specific task with a labeled dataset.

Supervised Learning Step

Model: (pre-trained in step #1)

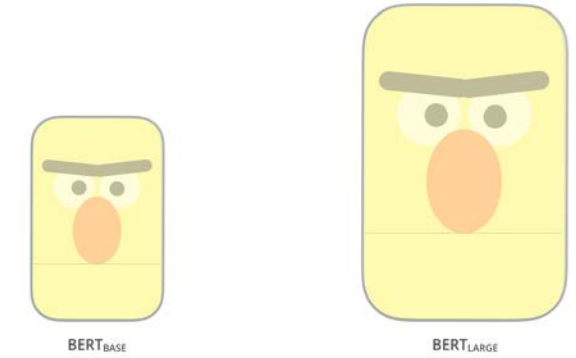


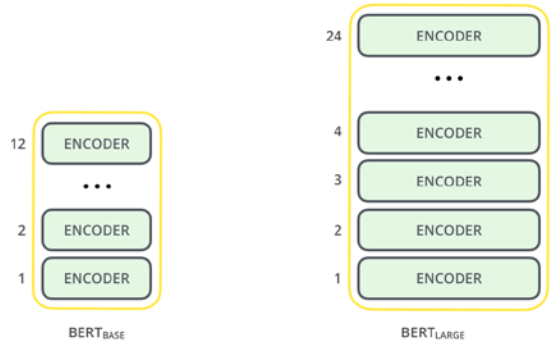
Dataset:

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached…	Not Spam

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





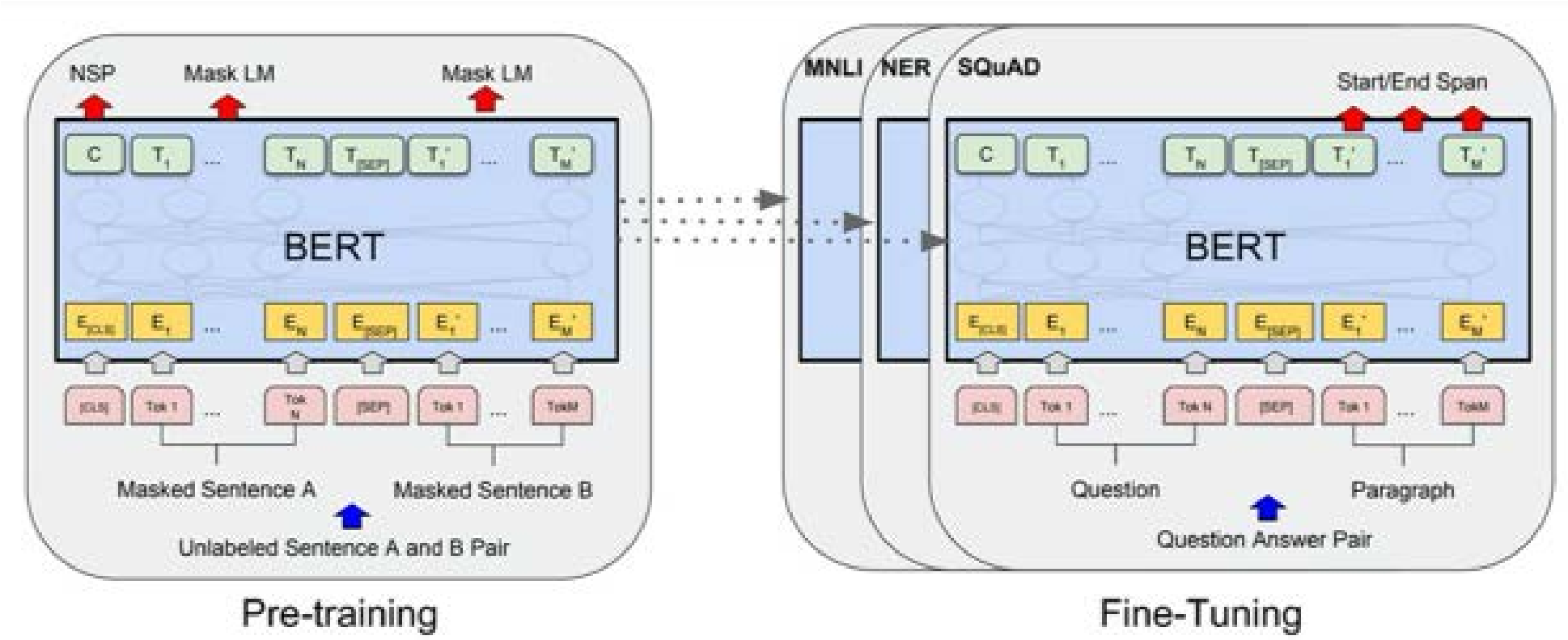
BERT is basically a trained Transformer Encoder stack.

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

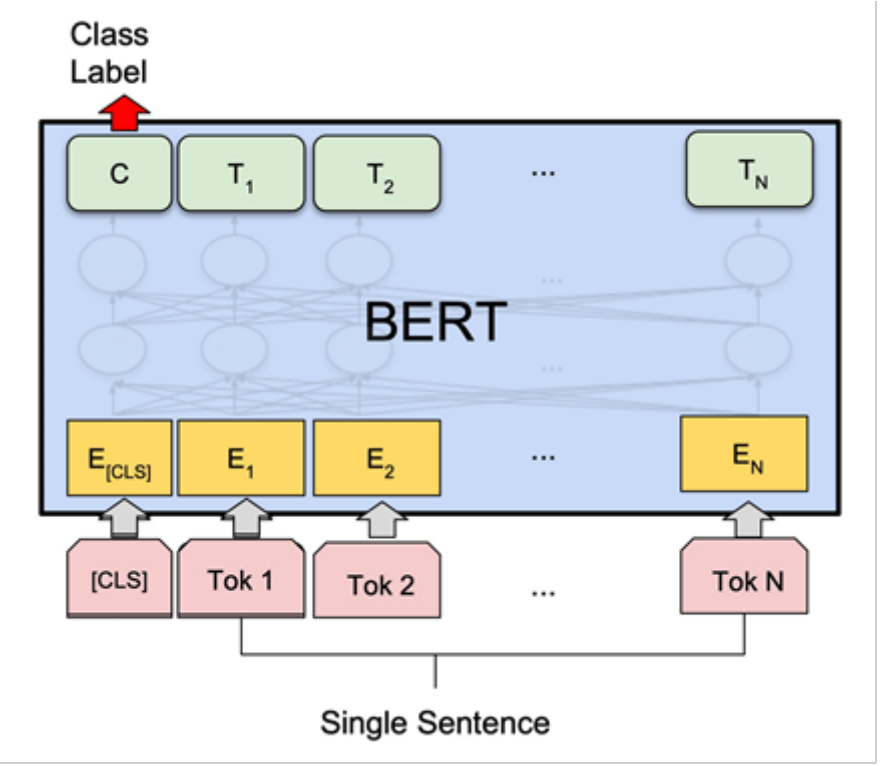
Slide from Jacob Delvin

Fine Tuning Procedure



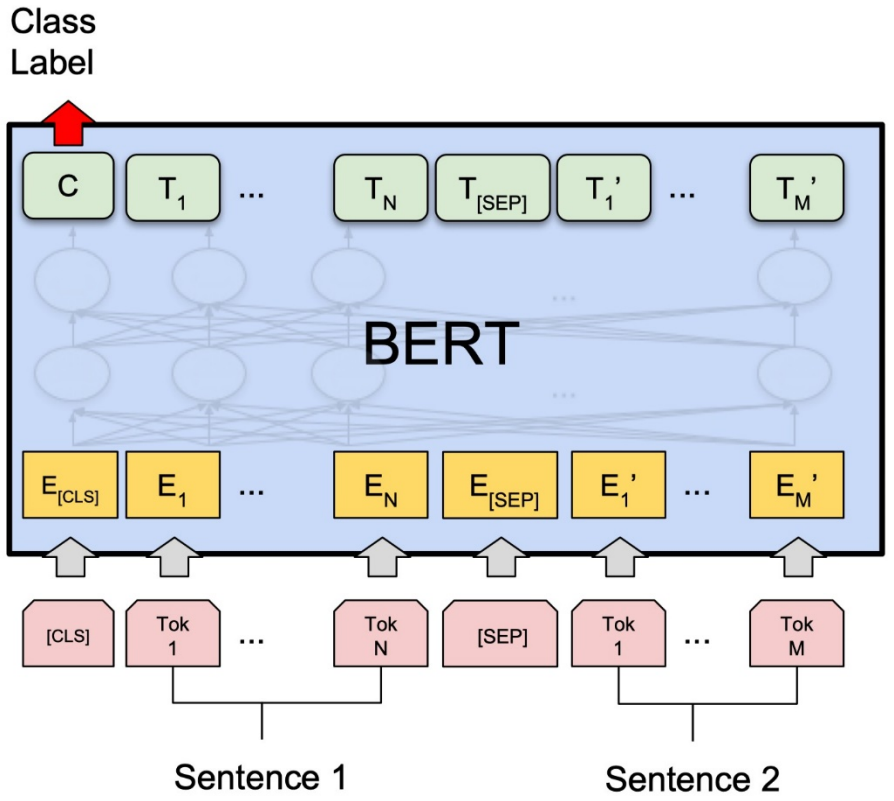
Example: Sentence Classification

CLS token is used to provide classification decision



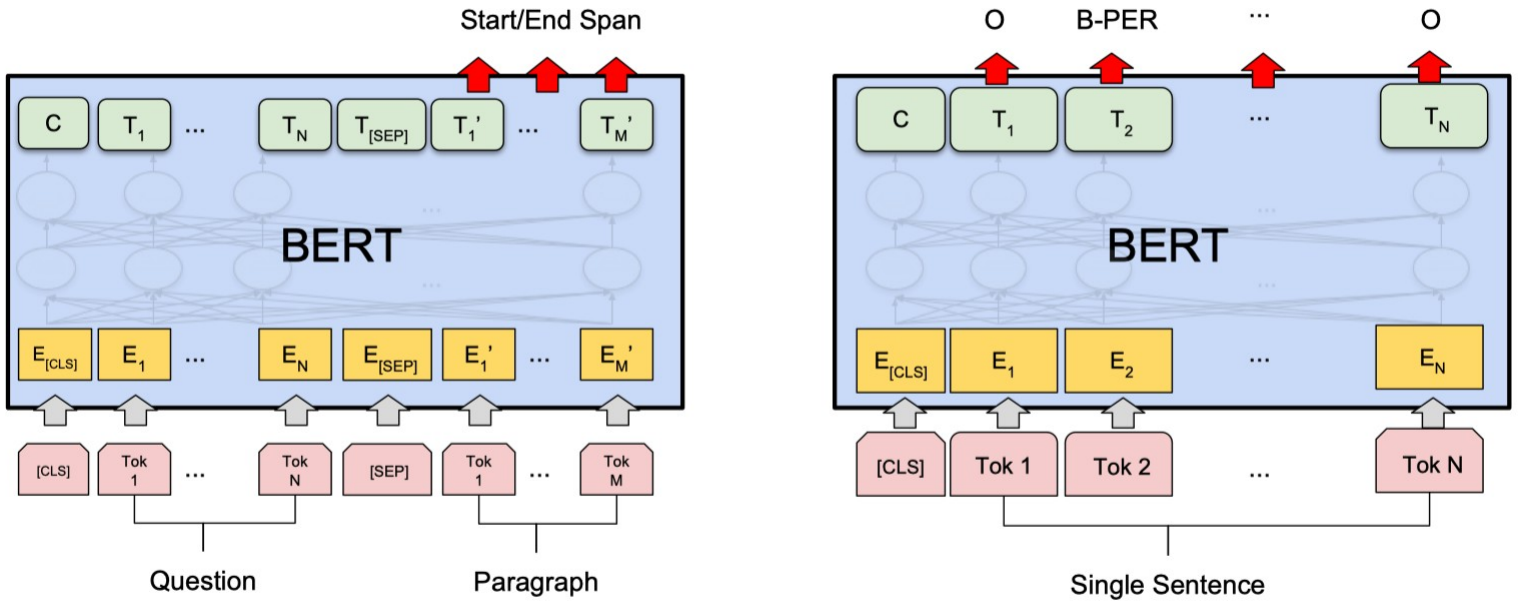
Sentence-pair Classification with BERT

- 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



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



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

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

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