BERT & Beyond

CS 490A, Fall 2021

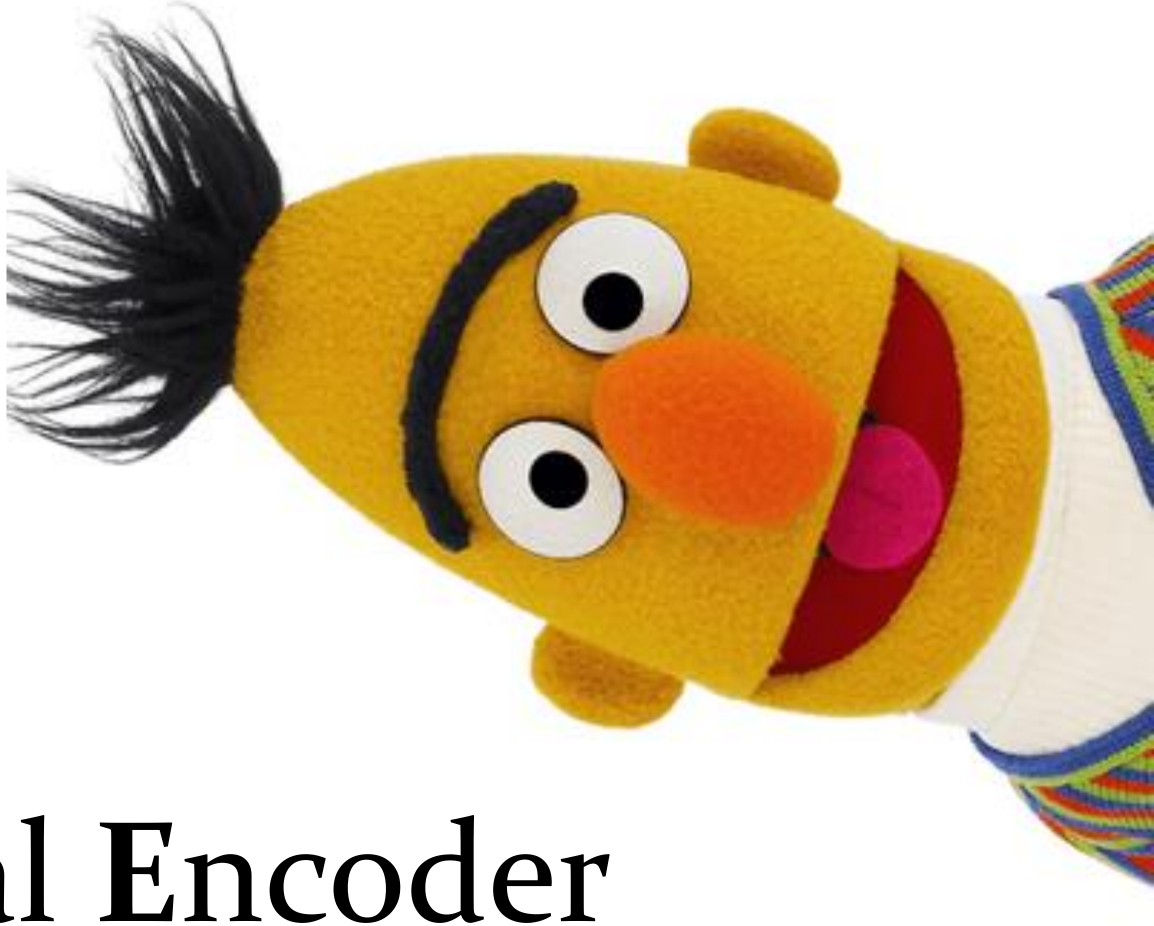
Applications of Natural Language Processing
https://people.cs.umass.edu/~brenocon/cs490a_f21

Brendan O’Connor & Laure Thompson

College of Information & Computer Sciences
University of Massachusetts Amherst

Administrivia

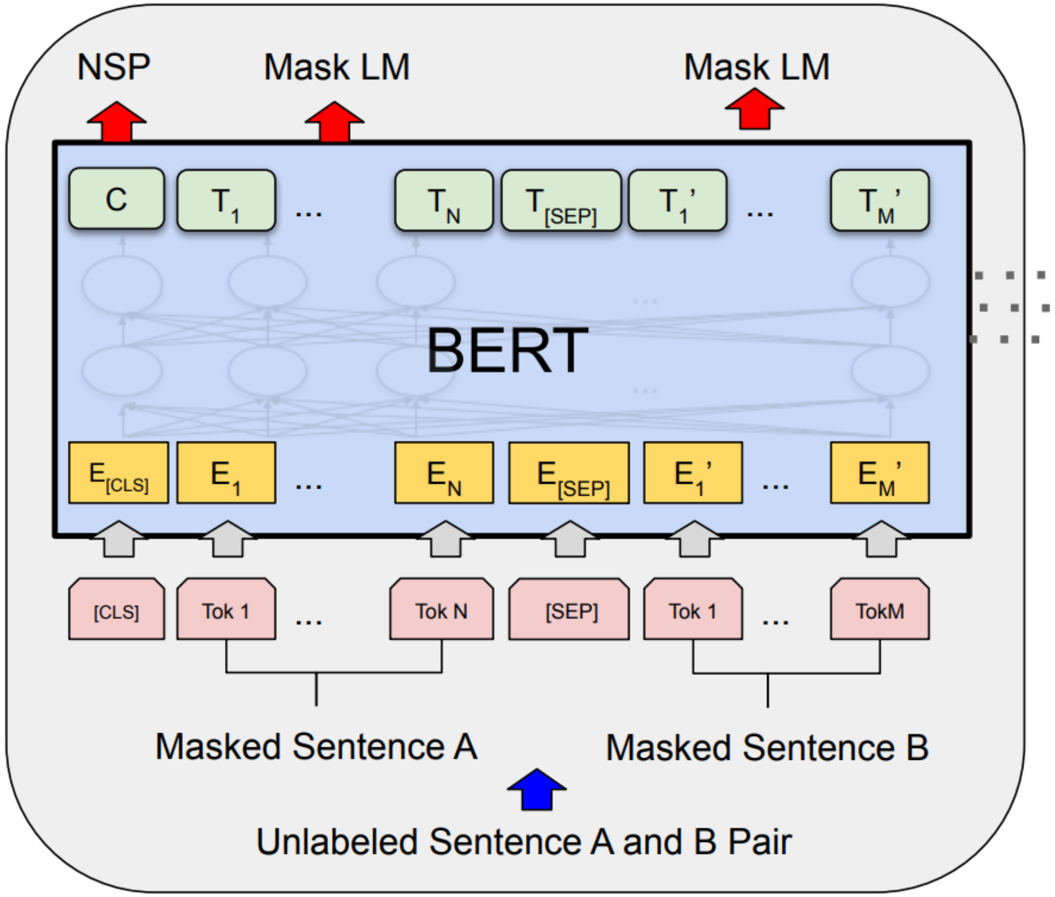
- Project Progress Report due **Monday**, 11/22
- Hugging Face (transformers) tutorial **tomorrow**, 11/19, @11am on Zoom
- HW4 will be released next week



BERT: **B**idirectional **E**ncoder **R**epresentations for **T**ransformers

Devlin et al. 2019

Pre-Training



Devlin et al. 2019

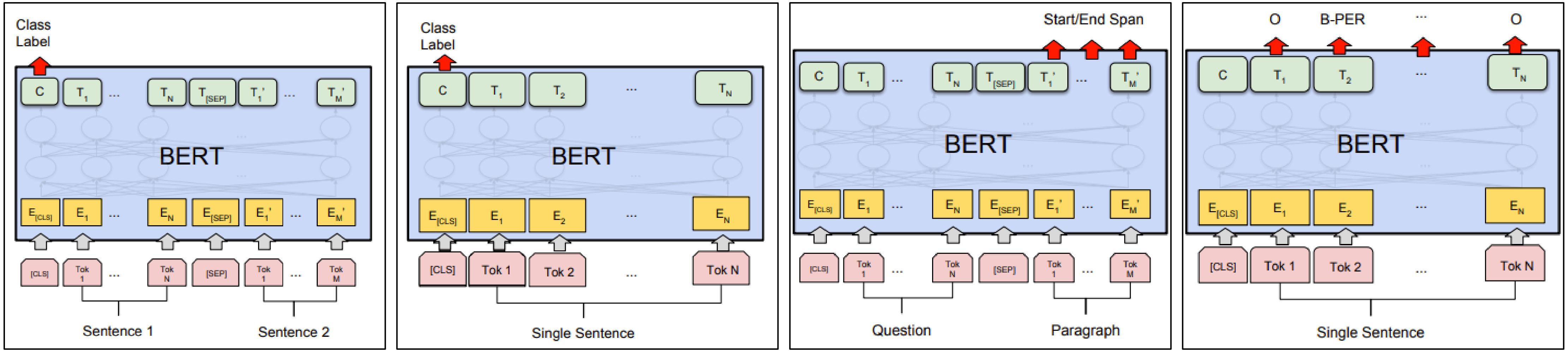
Fine-Tuning

Sentence Pair Classification

Single Sentence Classification

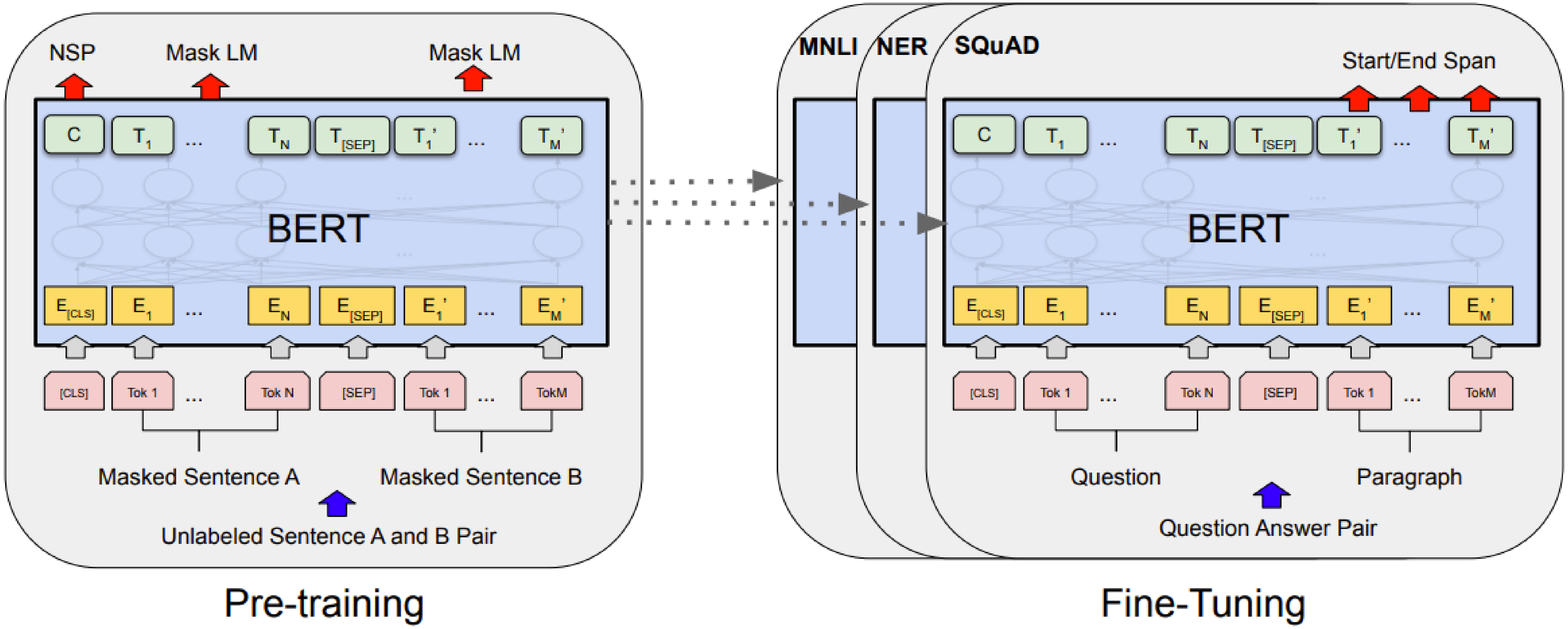
Question Answering

Single Sentence Tagging



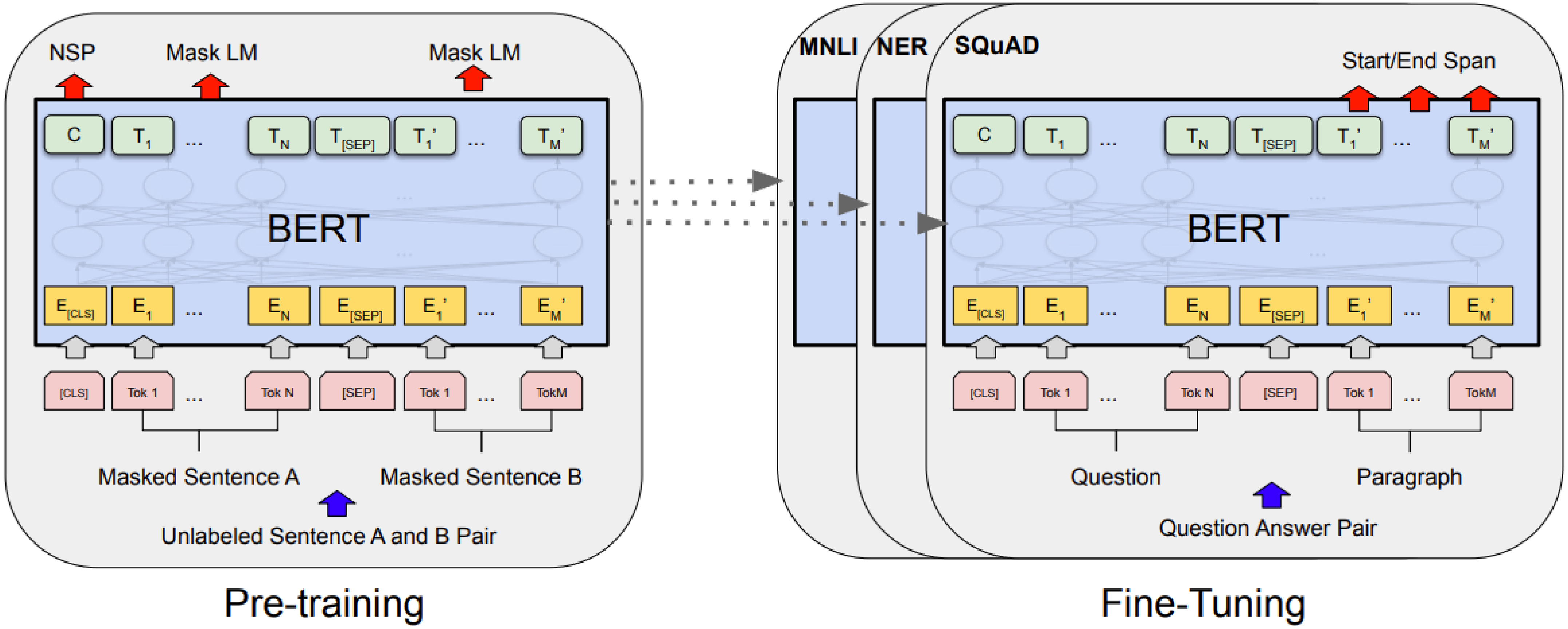
Devlin et al. 2019

Pre-Training vs. Fine-Tuning



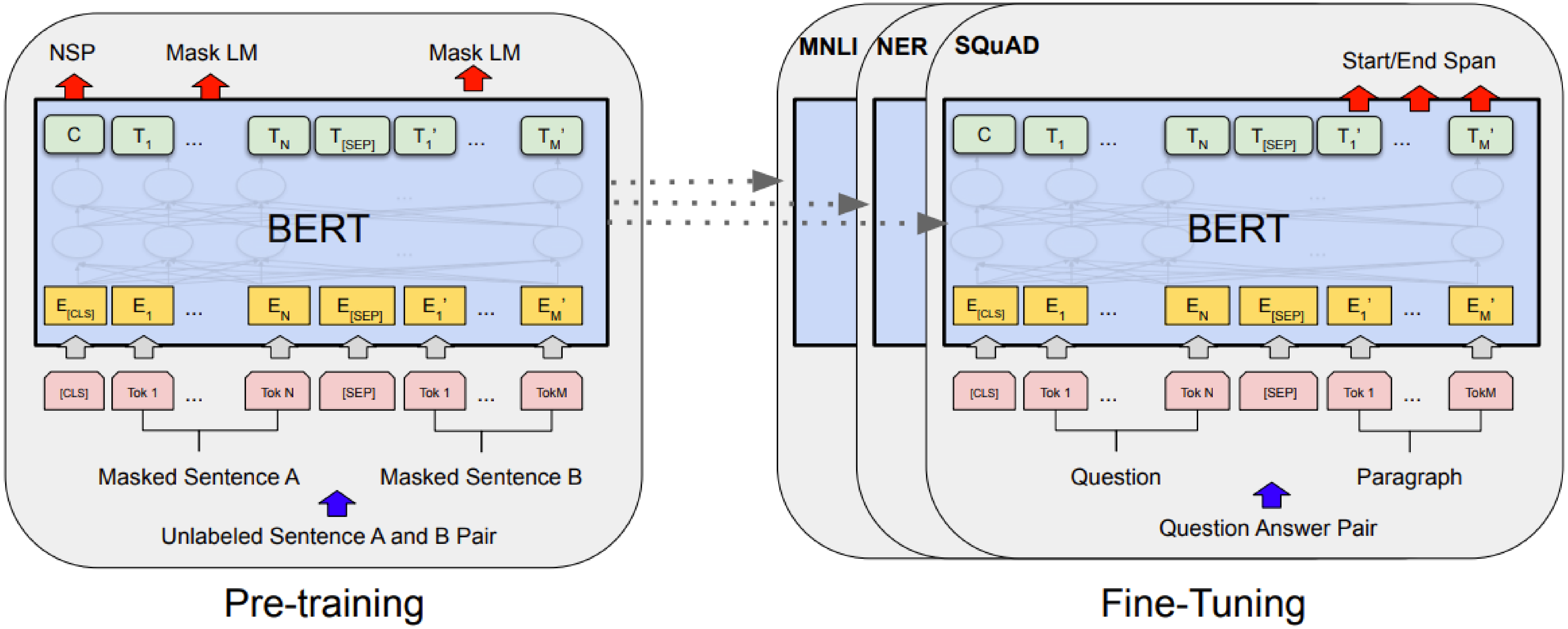
Devlin et al. 2019

Same internal architecture



Devlin et al. 2019

Different output layers & loss functions

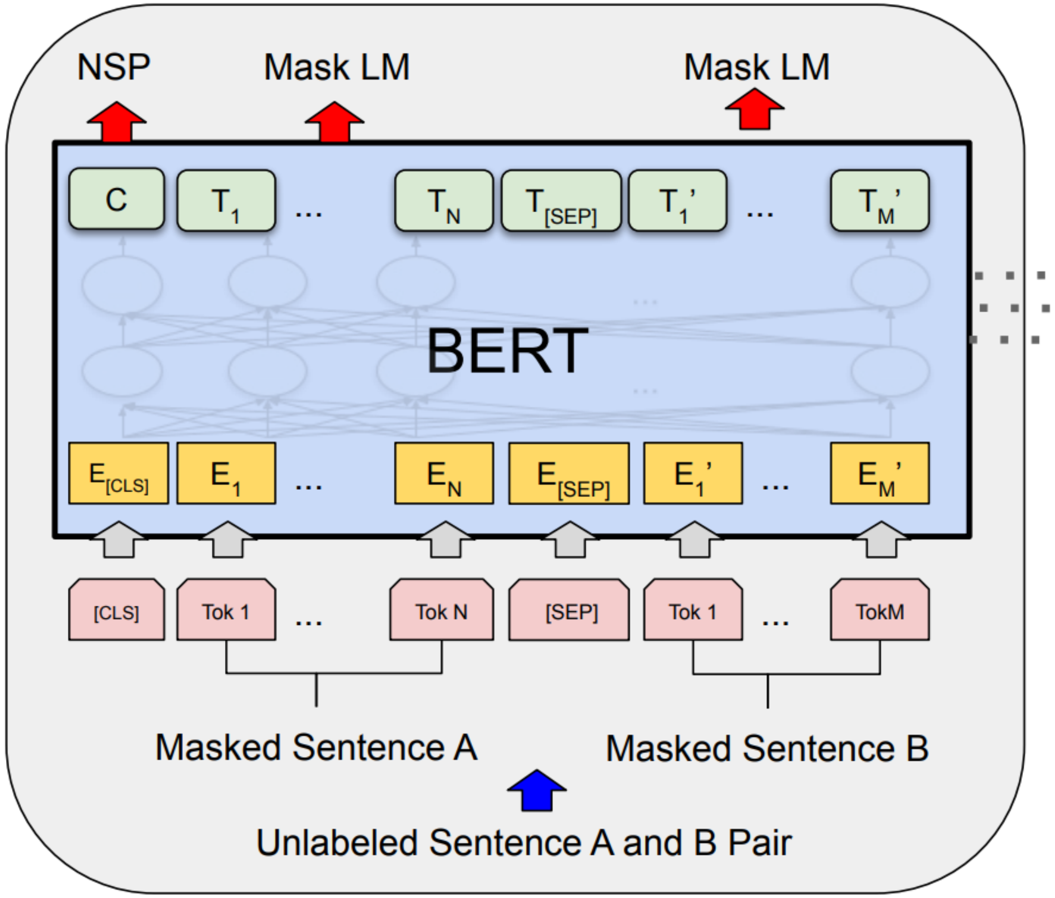


Devlin et al. 2019

Pre-Training BERT Tasks

(1) Masked Language Model

(2) Next Sentence Prediction



Devlin et al. 2019

Masked Language Model Procedure

Example: `my dog is` <u>`hairy`</u>

- 80% of the time: Replace the word with the [MASK] token

  `my dog is [MASK]`
- 10% of the time: Replace the word with a random word

  `my dog is apple`
- 10% of the time: Keep the word unchanged

  `my dog is hairy`

<u>Devlin et al. 2019</u>

Masked Language Model Procedure

Example: `my dog is hairy`

- 80% of the time: Replace the word with the [MASK] token

  Bidirectional language modeling

  `my dog is [MASK]`

- 10% of the time: Replace the word with a random word

  `my dog is apple`

- 10% of the time: Keep the word unchanged

  `my dog is hairy`

Devlin et al. 2019

Masked Language Model Procedure

Example: `my dog is hairy`

- 80% of the time: Replace the word with the [MASK] token

  `my dog is [MASK]`

Bidirectional language modeling

- 10% of the time: Replace the word with a random word

  `my dog is apple`

- 10% of the time: Keep the word unchanged

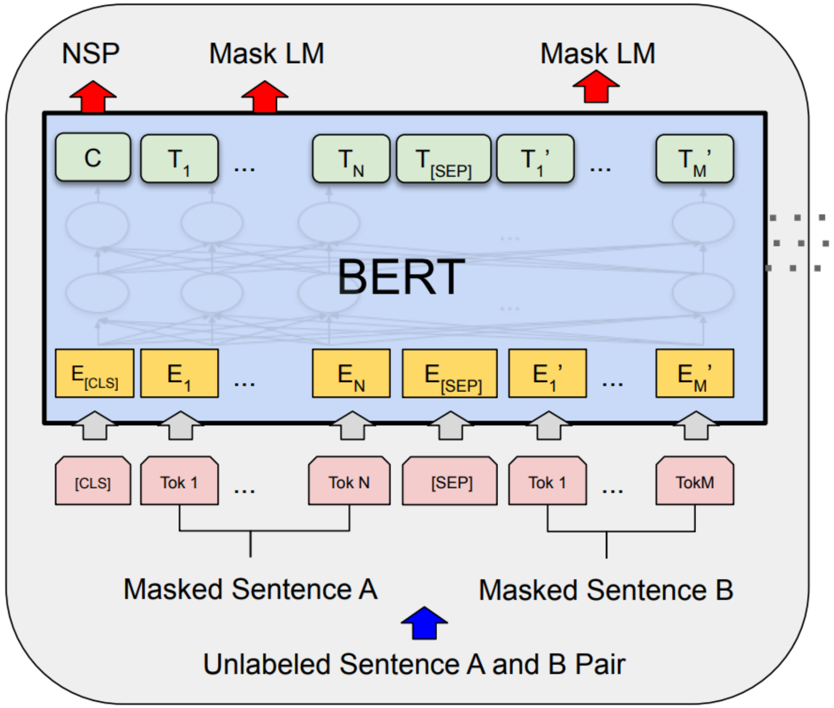
  `my dog is hairy`

Mitigate mismatch between pre-training & fine-tuning

Devlin et al. 2019

Pre-Training BERT: MLM

Idea: Predict vocab ID of masked tokens from final embeddings



Devlin et al. 2019

Masked Language Model Head

1. Transform T_i into a vector with `vocab_size` dimensions

Devlin et al. 2019

Masked Language Model Head

1. Transform T_i into a vector with `vocab_size` dimensions
2. Use softmax to obtain a probability distribution over the vocabulary

Devlin et al. 2019

Masked Language Model Head

1. Transform T_i into a vector with `vocab_size` dimensions
2. Use softmax to obtain a probability distribution over the vocabulary
3. *Use argmax to identify the most probable token*

Devlin et al. 2019

Masked Language Model Head

1. Transform T_i into a vector with `vocab_size` dimensions
2. Use softmax to obtain a probability distribution over the vocabulary
3. *Use argmax to identify the most probable token*

Loss Function: cross-entropy of distribution from 2

Only use masked tokens to calculate loss!

Devlin et al. 2019

Cross-Entropy

True distribution $\boldsymbol{p}$ and estimating distribution $\boldsymbol{q}$

$$H(p,q) = \sum_{x \in \mathrm{X}} p(x) \log q(x)$$

Cross-Entropy

True distribution $\boldsymbol{p}$ and estimating distribution $\boldsymbol{q}$

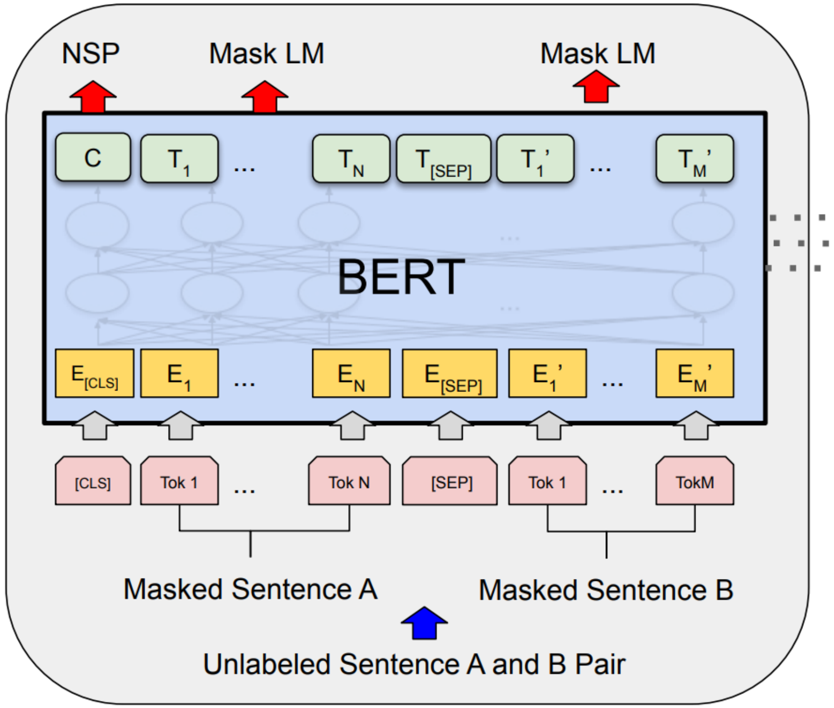
$$H(p,q) = \sum_{x \in \mathrm{X}} p(x) \log q(x)$$

$$H(p,q) = p(x_{true}) \log q(x_{true})$$

$$= \log q(x_{true})$$

Pre-Training BERT: NSP

Idea: Predict whether sentence B follows sentence A using the final embedding of the `[CLS]` token



Devlin et al. 2019

Next Sentence Prediction Head

1. Transform C into a vector with 2 dimensions

Devlin et al. 2019

Next Sentence Prediction Head

1. Transform C into a vector with 2 dimensions
2. Use softmax to obtain a probability distribution over the two classification labels

Devlin et al. 2019

Next Sentence Prediction Head

1. Transform C into a vector with 2 dimensions
2. Use softmax to obtain a probability distribution over the two classification labels
3. *Predict the more probable label*

Devlin et al. 2019

Next Sentence Prediction Head

1. Transform C into a vector with 2 dimensions
2. Use softmax to obtain a probability distribution over the two classification labels
3. *Predict the more probable label*

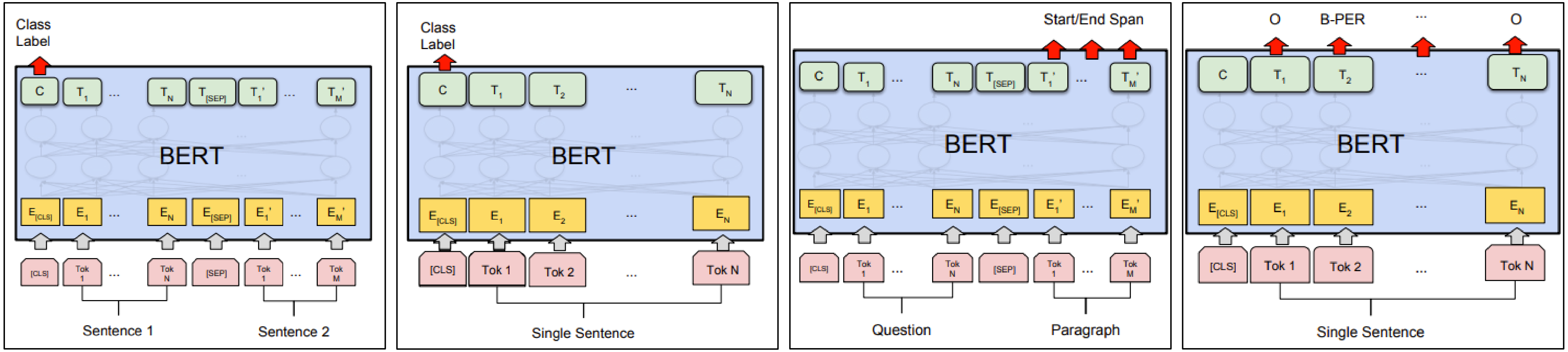
This is just a binary classification task!

Devlin et al. 2019

Fine-Tuning

Use pre-trained **model parameters** for initialization

Change pre-training **output layers** of BERT to suit task



Devlin et al. 2019

Huge gains for many tasks!
GLUE Results

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Devlin et al. 2019

Huge gains for many tasks!
GLUE Results

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

MNLI = Multi-genre Natural Language Inference

Premise	A woman selling bamboo sticks talking to two men on a loading dock.
Entailment	There are **at least** three **people** on a loading dock.
Neutral	A woman is selling bamboo sticks **to help provide for her family.**
Contradiction	A woman is **not** taking money for any of her sticks.

Devlin et al. 2019; Gururangan et al. 2017

Huge gains for many tasks!
GLUE Results

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

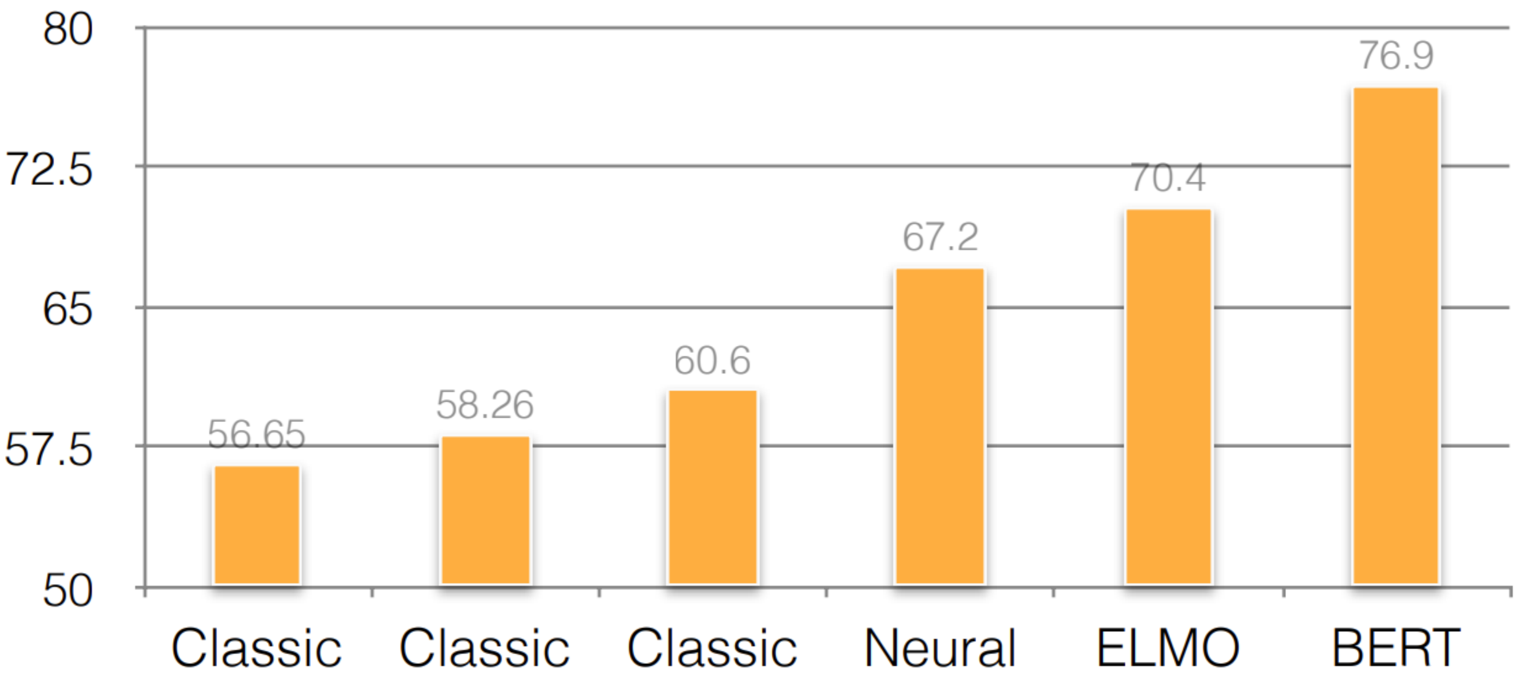
CoLA = Corpus of Linguistic Acceptability

Included	Morphological Violation	(a)	*Maryann should leaving.
	Syntactic Violation	(b)	*What did Bill buy potatoes and _?
	Semantic Violation	(c)	*Kim persuaded it to rain.
Excluded	Pragmatical Anomalies	(d)	*Bill fell off the ladder in an hour.
	Unavailable Meanings	(e)	*He$_i$ loves John$_i$. (*intended*: John loves himself.)
	Prescriptive Rules	(f)	Prepositions are good to end sentences with.
	Nonce Words	(g)	*This train is arrivable.

Devlin et al. 2019; Warstadt et al. 2019

Huge gains for many tasks! Coreference Resolution

"**I** voted for **Nader** because **he** was most aligned with **my** values," **she** said.



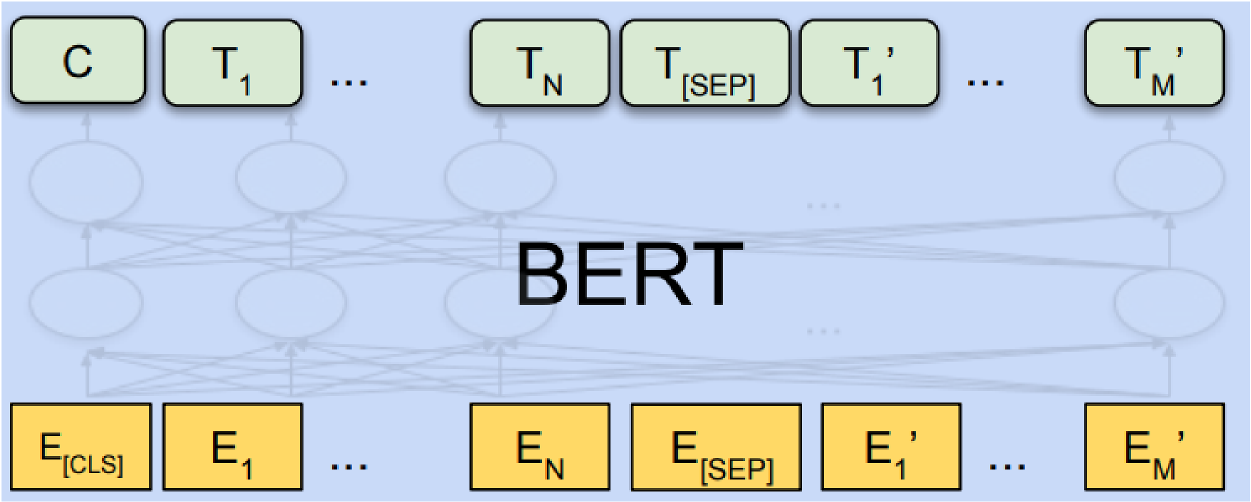
From slide of Bamman (2021)

Using BERT

BERT “Internal” Features

Internal token-level embeddings are 768 dimensions

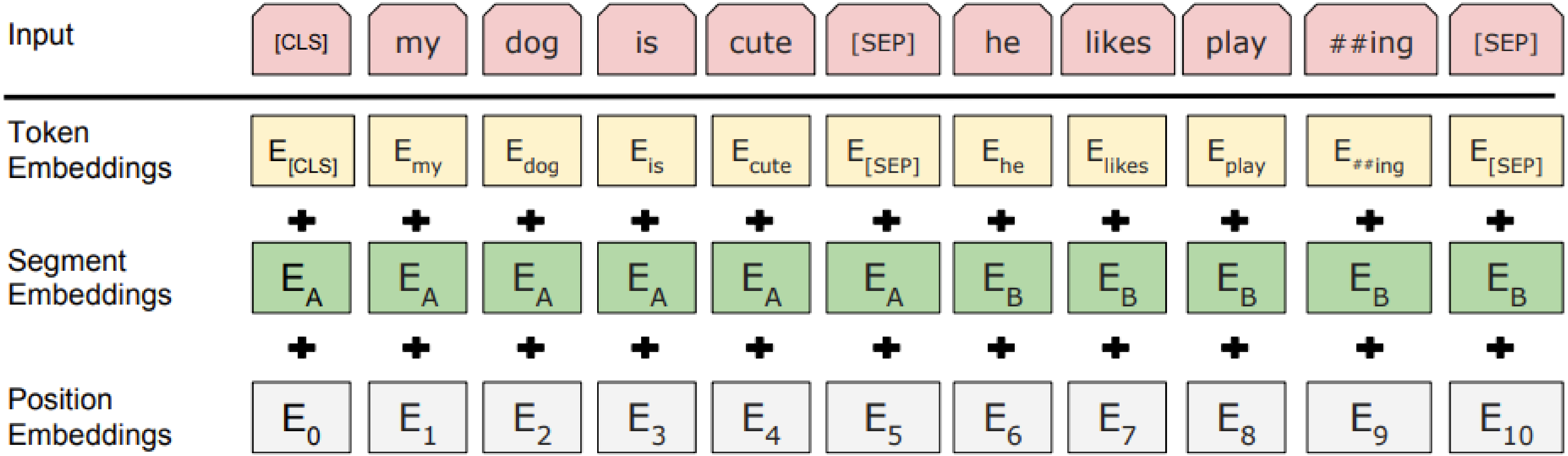
One for encoding layer, one for each hidden layer (12)



Devlin et al. 2019

Building Initial Token Embeddings

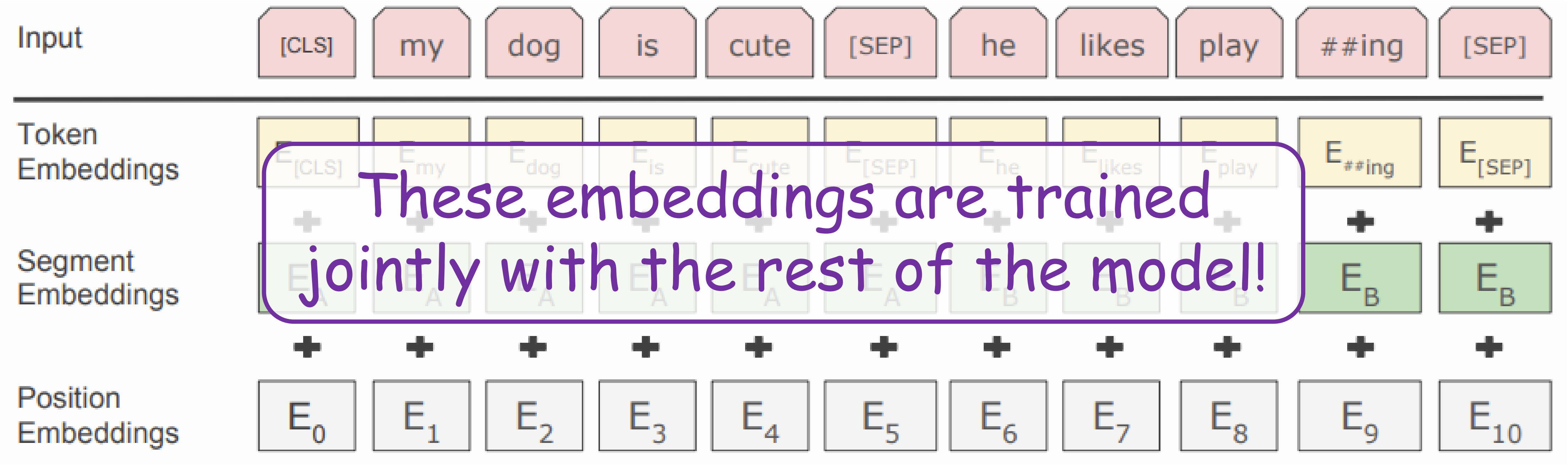
The input token embeddings $\boldsymbol{E_i}$ are the sum of 3 embeddings encoding token, segment, and position information



Devlin et al. 2019

Building Initial Token Embeddings

The input token embeddings $\boldsymbol{E_i}$ are the sum of 3 embeddings encoding token, segment, and position information



Devlin et al. 2019

BERT (Sub)Tokens

BERT tokens do **not** strictly correspond to word tokens

Devlin et al. 2019

Subword-Based Tokenization

the dog fetched the stick

Subword-Based Tokenization

the dog fetched the stick

Word-Based: `the`, `dog`, `fetched`, `the`, `stick`

Subword-Based Tokenization

the dog fetched the stick

Word-Based: `the`, `dog`, `fetched`, `the`, `stick`

Character-Based: `t`, `h`, `e`, `_`, `d`, `o`, `g`, `_`, `f`, `e`, `t`, `c`, `h`, `e`, `d`, `_`, `t`, `h`, `e`, `_`, `s`, `t`, `i`, `c`, `k`

Subword-Based Tokenization

the dog fetched the stick

Word-Based: `the`, `dog`, `fetched`, `the`, `stick`

Token-Based: `the`, `dog`, `fetch`, `##ed`, `the`, `stick`

Character-Based: `t`, `h`, `e`, `_`, `d`, `o`, `g`, `_`, `f`, `e`, `t`, `c`, `h`, `e`, `d`, `_`, `t`, `h`, `e`, `_`, `s`, `t`, `i`, `c`, `k`

Class Activity

colab.research.google.com/drive/1v3iustM3huxMVSItknowzWGwdrQpZoco

WordPiece

Goal: Given a training corpus and number of desired tokens **D**, select **D** wordpieces (i.e., subtokens) so that the training corpus is minimally segmented

Wu et al. 2016

WordPiece

Goal: Given a training corpus and number of desired tokens **D**, select **D** wordpieces (i.e., subtokens) so that the training corpus is minimally segmented

Top-Down: Break the starting vocabulary into smaller components until there are only **D**

Wu et al. 2016

Alternative: Byte Pair Encoding (BPE)

Used by GPT-2 and RoBERTa

Goal: Using a training corpus build a set of **D** subtokens to tokenized the training corpus

Sennrich et al. 2016

Alternative: Byte Pair Encoding (BPE)

Initial: The symbol vocabulary is the set of characters in the training corpus.

Do: For the most frequent 2-symbol sequence (A, B) in the training corpus, create a new symbol AB and replace all instances of with (A, B) with AB.

End: When there are D symbols.

Sennrich et al. 2016

After BERT: RoBERTa

Same BERT architecture, but with different pre-training

- Drop the Next Sentence Prediction pre-training task
- Use a BPE-based subtokenization method
- Pre-train with more data for a longer duration

Liu et al. 2019

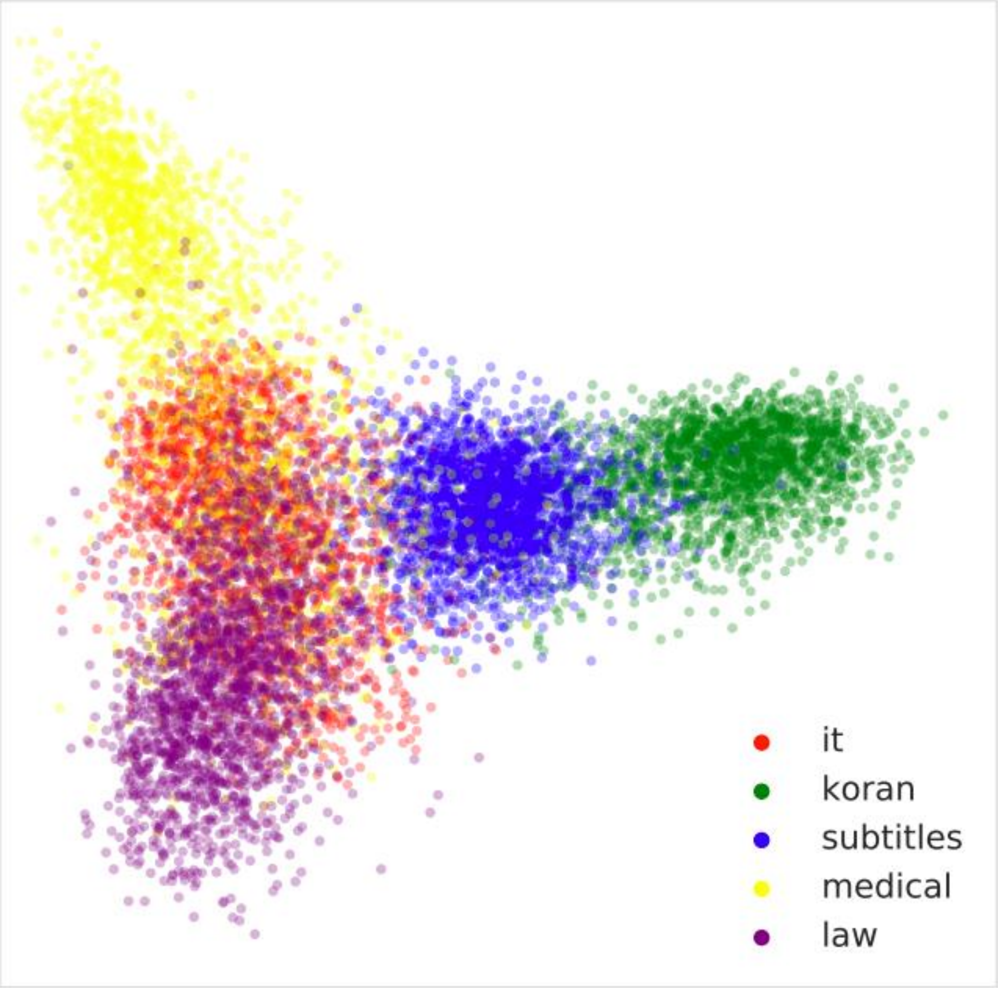
After BERT: RoBERTa

Same BERT architecture, but with different pre-training

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
$\text{XLNet}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

Liu et al. 2019

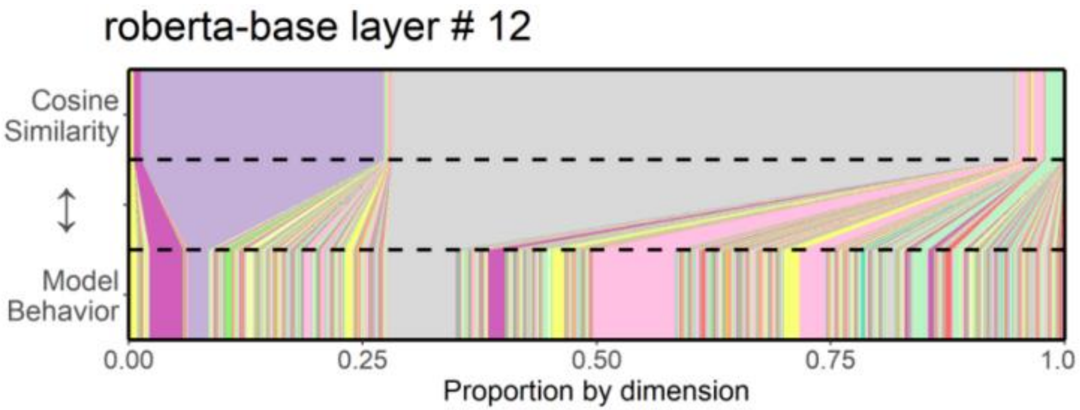
BERT representations reflect domain

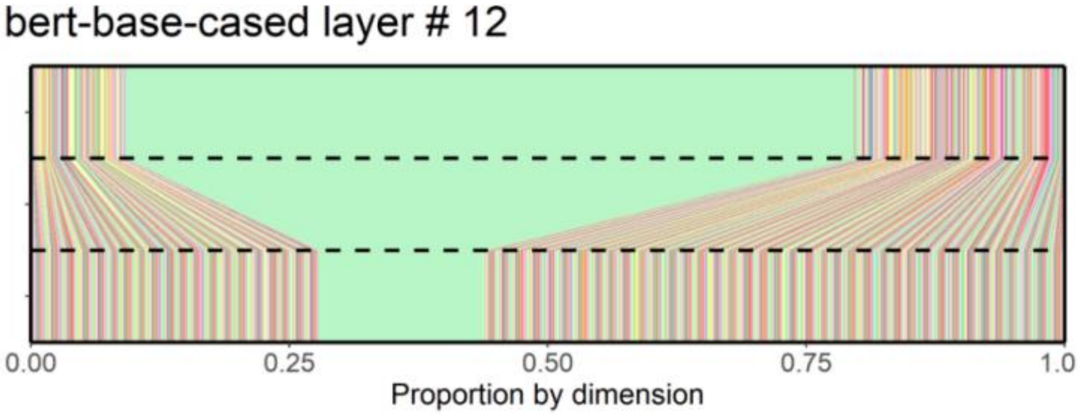


Aharoni & Goldberg 2020

Trouble with raw embeddings

A few dimensions will dominate similarity measures such as cosine similarity and Euclidean distance





Timkey & van Schijndel 2021