CS 333: Natural Language Processing

Fall 2023

Prof. Carolyn Anderson
Wellesley College

Announcements

- Francesca Lucchetti will be giving a guest lecture on Tuesday.
- My help hours next week:
  - Monday: 3:30-5
  - Friday: 3:30-4:30

The Deep Learning Pipeline

The Deep Learning Pipeline

Deep learning models can be run in two modes:

- **Training**: update a model's weights to fit new data. This is *supervised learning* because it requires input/ output pairs (labeled data).

- **Inference**: run data through a model to make predictions. This requires only input data. It does not change the model weights.

Transfer Learning

Contemporary machine learning often involves multiple stages of training:

- **Pre-training**: train a large model that will be used by many downstream applications
  Called a foundation model in Bommasani et al. 2021

- **Fine-tuning**: adapting a pre-trained model to a new task or dataset by training it on new data, starting from existing weights.

- **Prompt Engineering**: framing a task so that it can be solved by a pretrained language model.

-

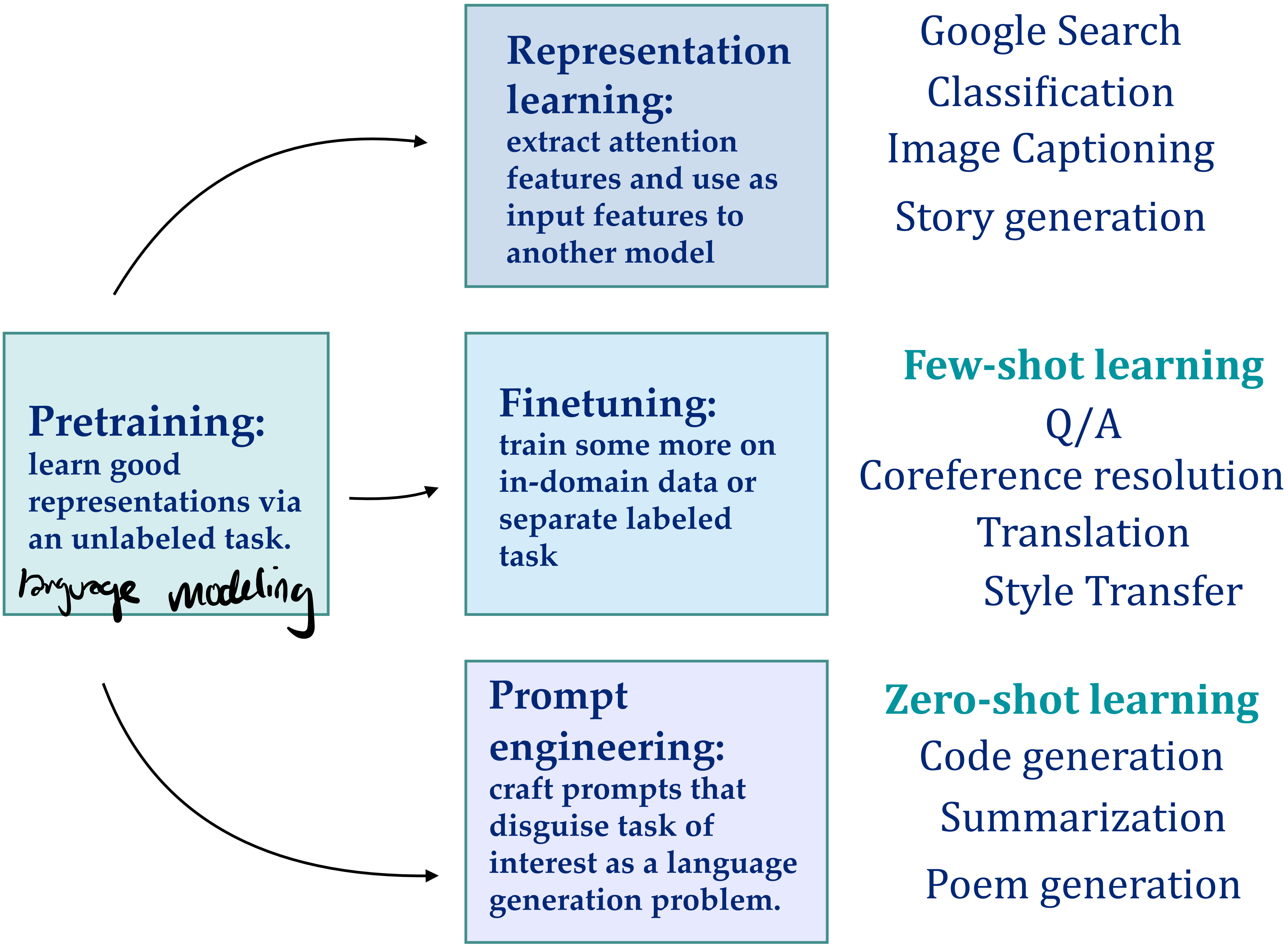
Transfer Learning

Contemporary machine learning models may also build upon other models by **freezing the weights of the original model** and taking some of its components as input.

For instance, the **weights of attention heads** may be re-used as embeddings to be fed in as input to a downstream model.

This is called **feature extraction**. or representation learning

This is what we did in the recipe classifier: we took attention weights from RoBERTa to use as features in our classifier!


Pretraining:
learn good representations via an unlabeled task.
language modeling
Representation learning:
extract attention features and use as input features to another model
Google Search
Classification
Image Captioning
Story generation
Finetuning:
train some more on in-domain data or separate labeled task
Few-shot learning
Q/A
Coreference resolution
Translation
Style Transfer
Prompt engineering:
craft prompts that disguise task of interest as a language generation problem.
Zero-shot learning
Code generation
Summarization
Poem generation

The Recent Past

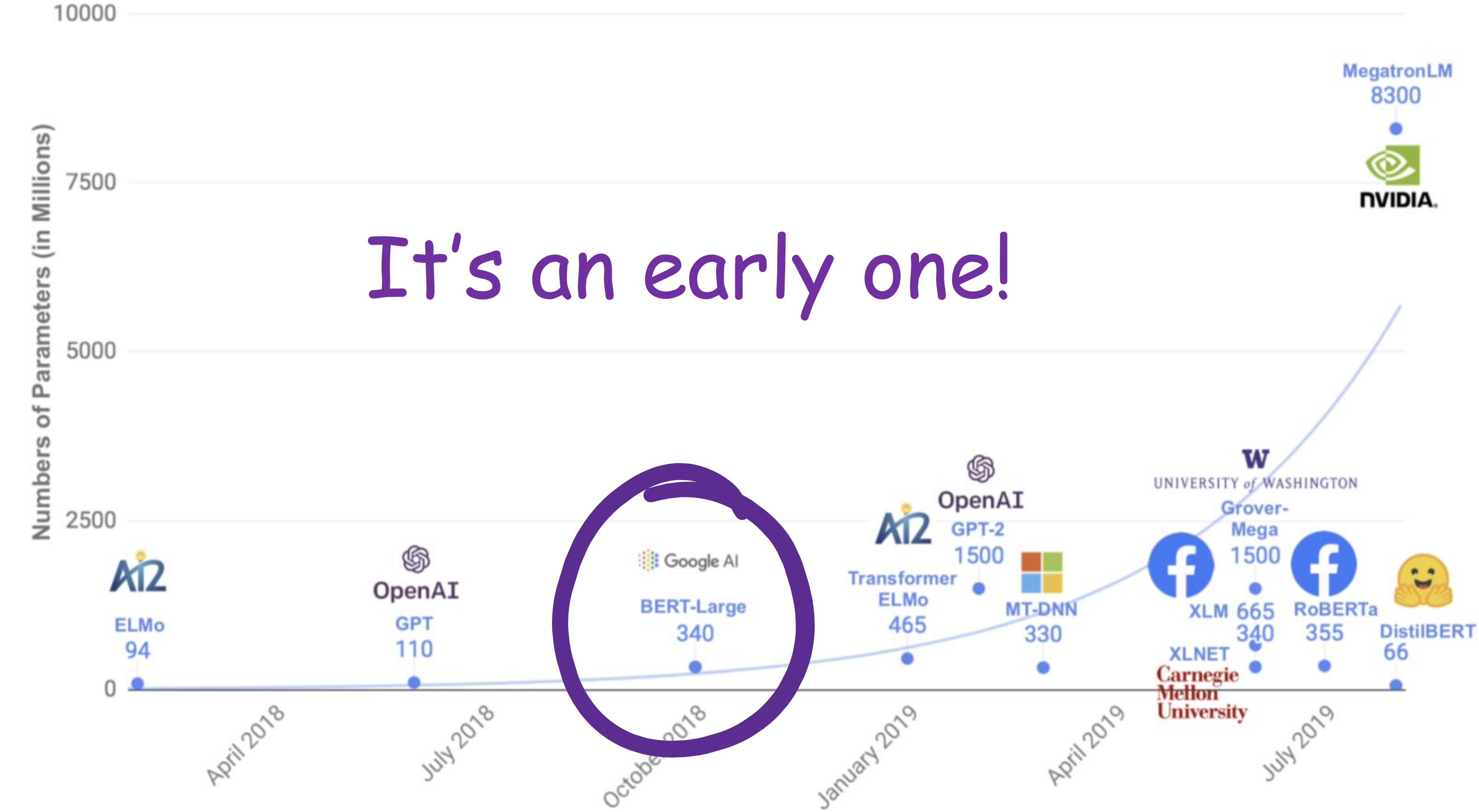

Welcome to Sesame Street

BERT: <u>**B**</u>idirectional <u>**E**</u>ncoder <u>**R**</u>epresentations for <u>**T**</u>ransformers

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

Slides adapted from UMass CS 490A

Why BERT?



10000
7500
5000
2500
0
Numbers of Parameters (in Millions)
MegatronLM
8300
NVIDIA.
It's an early one!
Google AI
BERT-Large
340
ELMo
94
OpenAI
GPT
110
Transformer
ELMo
465
OpenAI
GPT-2
1500
MT-DNN
330
UNIVERSITY of WASHINGTON
Grover-
Mega
1500
XLM 665
340
XLNET
Carnegie
Mellon
University
RoBERTa
355
DistilBERT
66
April 2018
July 2018
October 2018
January 2019
April 2019
July 2019

Why BERT?

Highly influential!

25,048 Citations

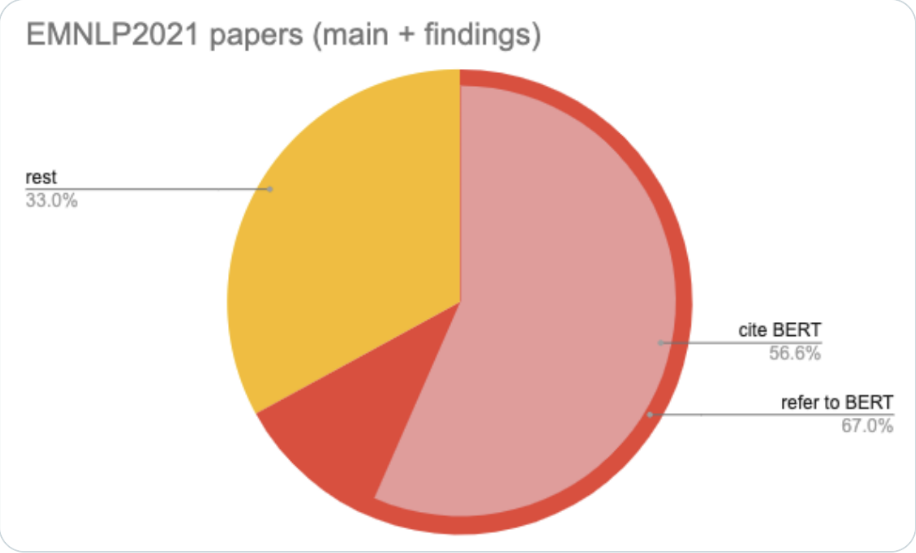
Highly Influential Citations	7,670
Background Citations	10,124
Methods Citations	13,635
Results Citations	463

View All

Gabriel Stanovsky
@GabiStanovsky

I skimmed through many papers from @emnlpmeeting, which got me thinking - what % of papers refer to BERT, and out of those, how many cite it?
Here's the answer*: 67% of papers refer to BERT (!), and 56% cite it.

*computed automatically, exact #'s may vary
#EMNLP2021 #NLProc



12:21 PM · Nov 9, 2021 · Twitter Web App

Sentiment Analysis

Circa 2013: train RNNs on labeled datasets

issues:
- World knowledge
- Syntax
- Semantics

from not very much data (expensive!)

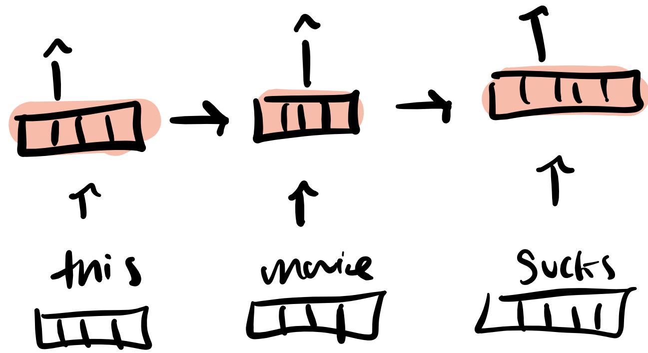
Circa 2017: what if we could reuse a language model for downstream tasks?

Idea: language modeling is cheap! Why not learn first from an LM?

-

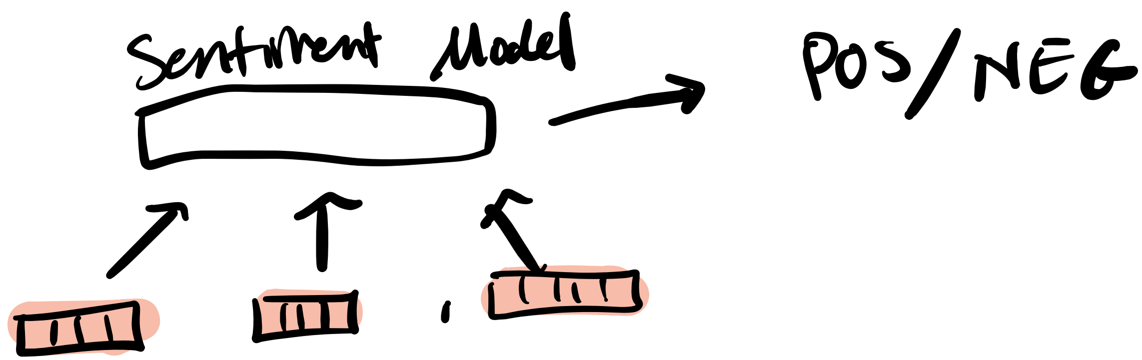
ELMO (2018)

1. Pretrain an RNN LM on lots of data!



2 unidirectional
RNN LMs
Froze LM weights

2. Freeze the LM weights and reuse the hidden states as input to another model



BERT (2019)

1 LM : Transformer

LM trained on a <u>ton</u> of data

Two training objectives:

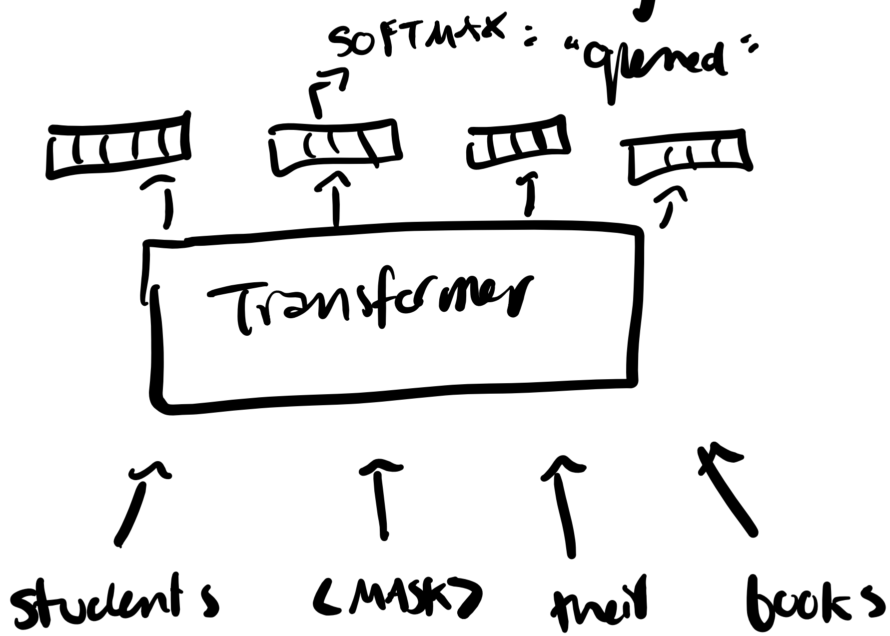
1. Masked language modeling
2. Next sentence prediction

Masked Language Modeling

Input: a sequence where some words are randomly masked:

<MASK>

Goal: predict the identity of masked token

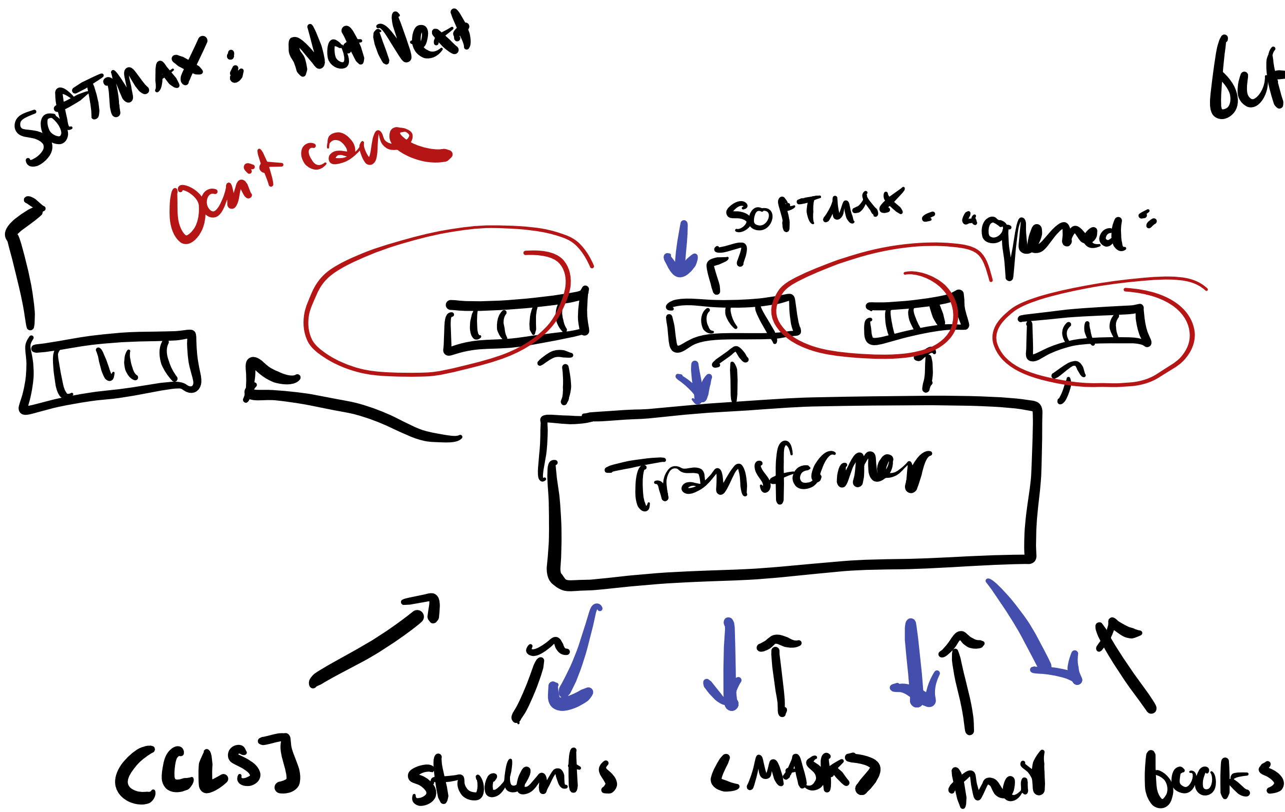


Training

Compute cross-entropy loss

but only for masked

token.



Backprop updates

to all weights,

not just the

masked embeddings.

Why stop at 1 mask?

You can actually mask up to 40%

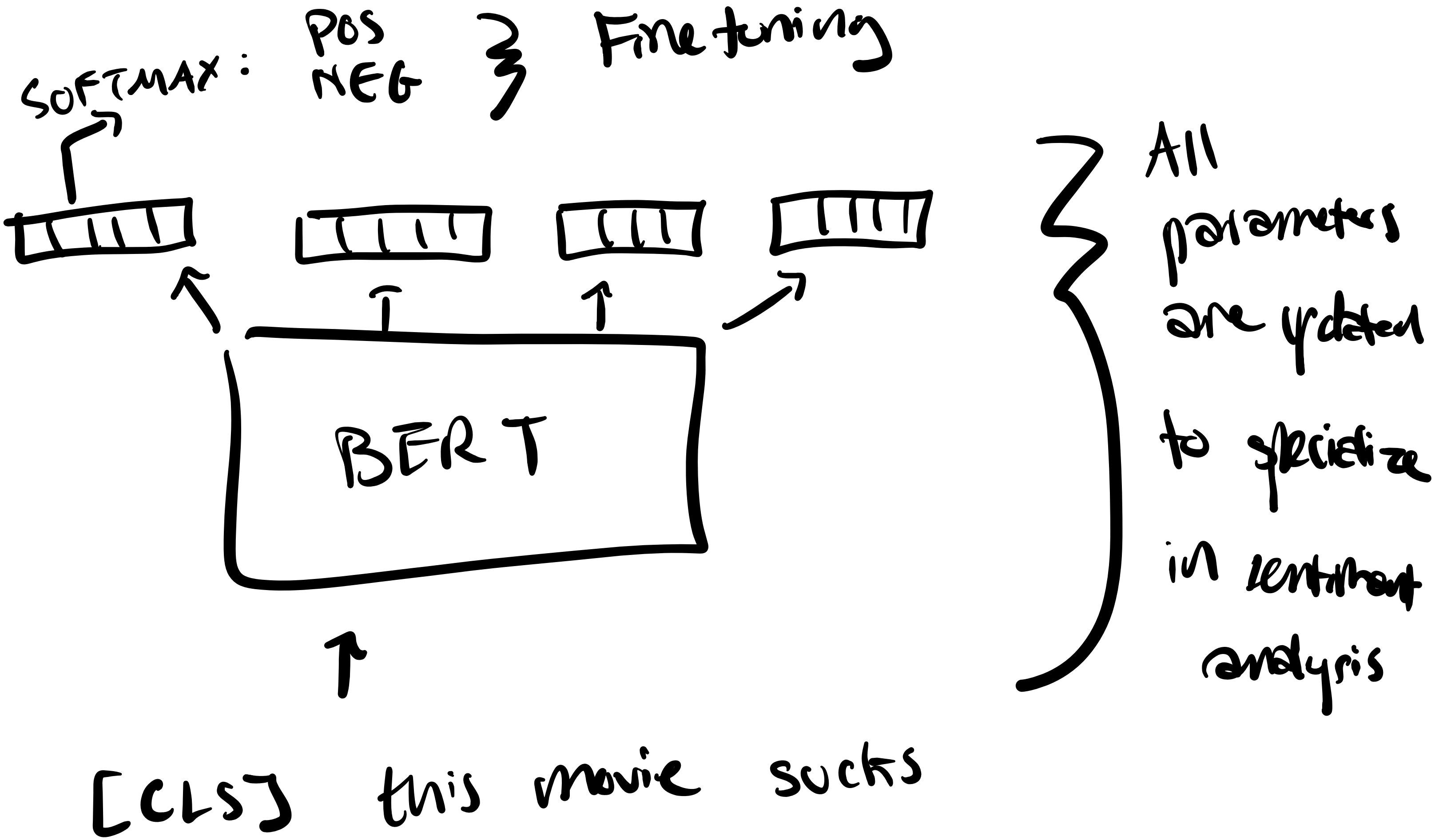
Next Sentence Prediction

Input: [CLS] the man <MASK> to store [SEP] he bought a gallon of [MASK> [SEP]

predict: IsNext or NotNext

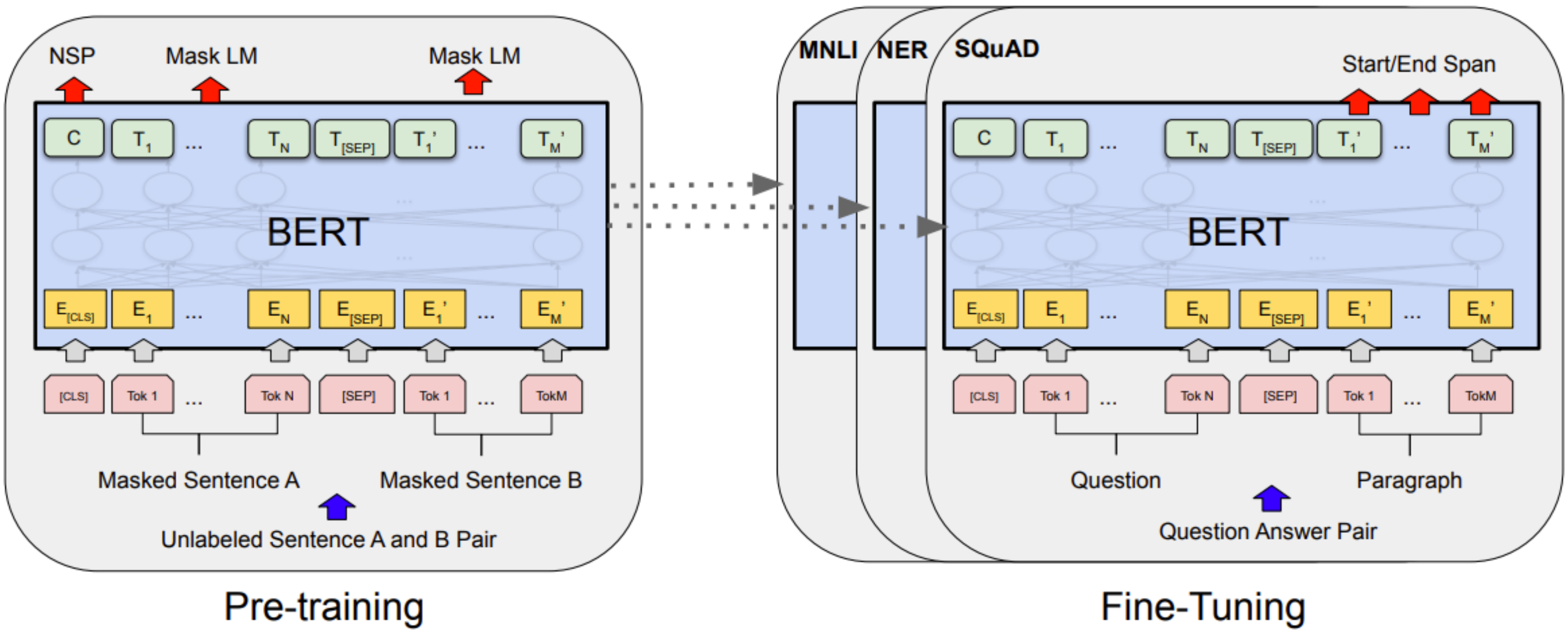
from the [CLS] embedding

How To Use



Fine tune For Sentence Classification

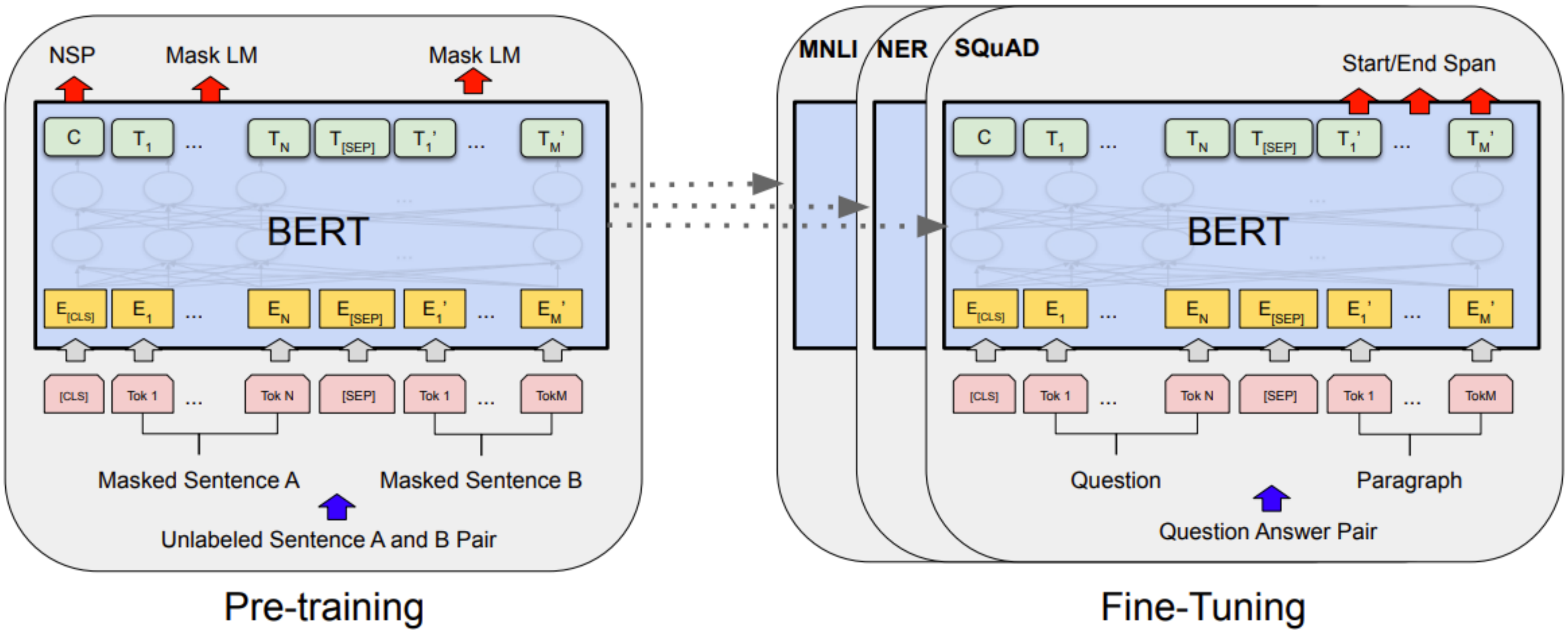
Pre-Training vs. Fine-Tuning



Devlin et al. 2019

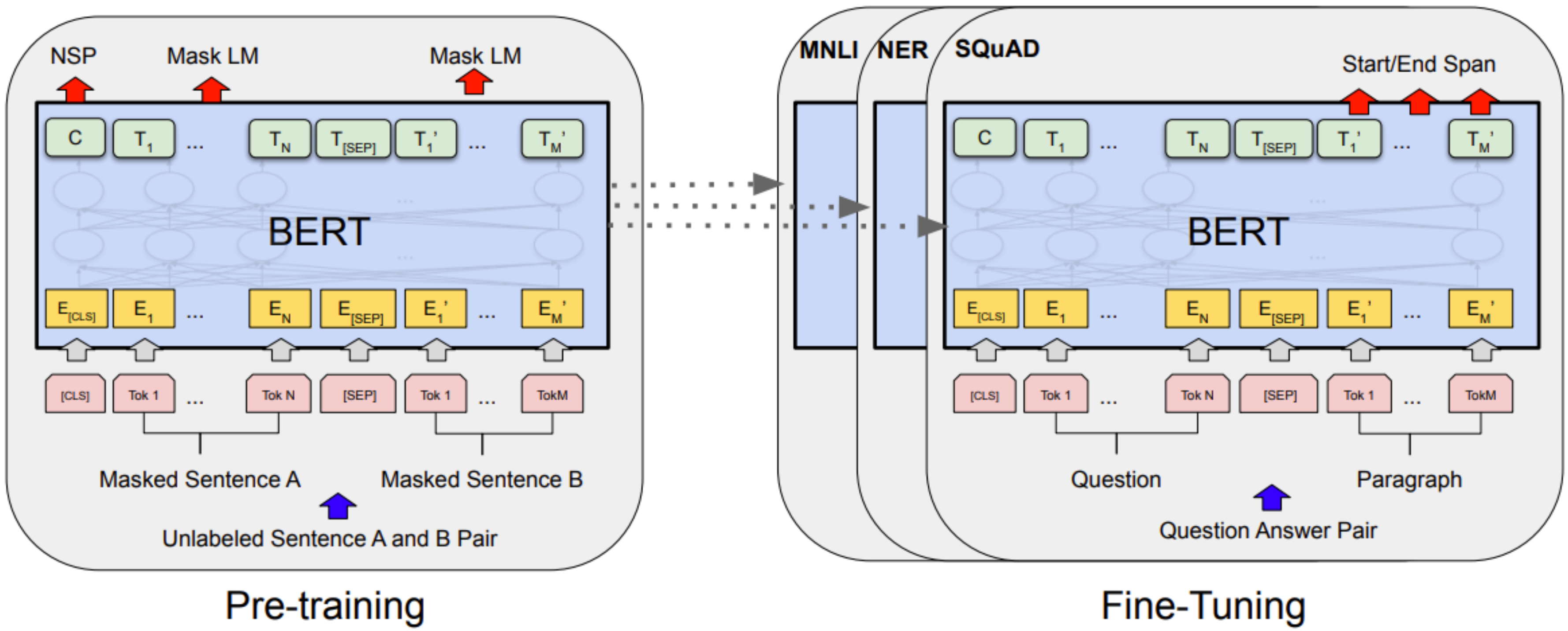
Slides adapted from UMass CS 490A

Same internal architecture



Devlin et al. 2019

Different output layers & loss functions



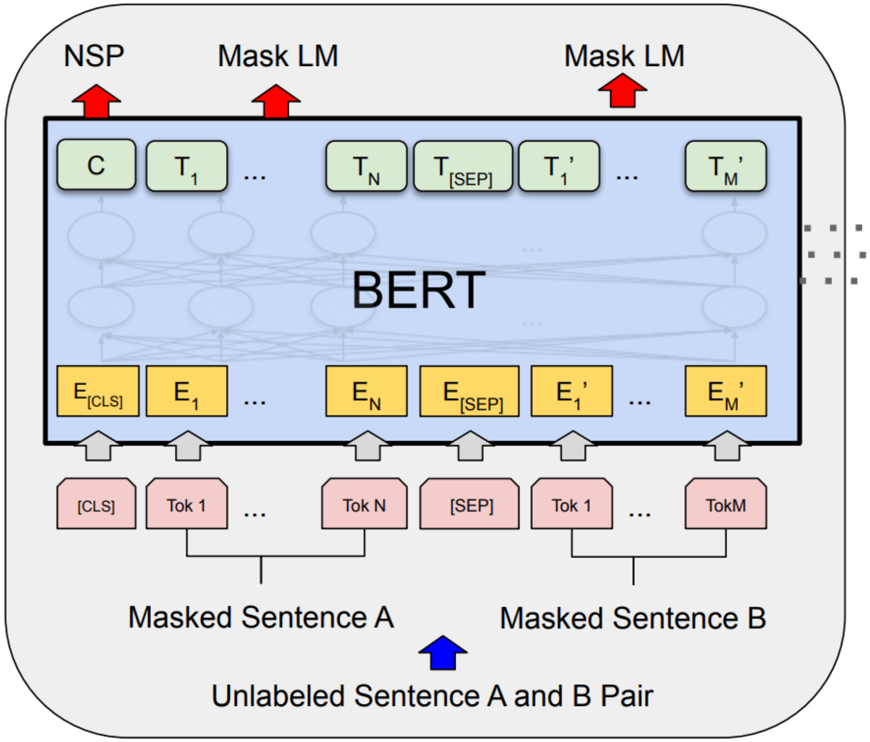
Devlin et al. 2019

Slides adapted from UMass CS 490A

Pre-Training BERT Tasks

(1) Masked Language Model

(2) Next Sentence Prediction



Devlin et al. 2019

Slides adapted from UMass CS 490A

Masked Language Model

Setting: Randomly mask some tokens of the input

Objective: Predict the original word types of each masked token based solely on its context

Masked Language Model Procedure

Apply procedure to 15% of tokens

- 80% of the time: Replace the word with the [MASK] token
- 10% of the time: Replace the word with a random word
- 10% of the time: Keep the word unchanged

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]`

- 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

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

Bidirectional language modeling

`my dog is [MASK]`

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

`my dog is apple`

Mitigate mismatch between pre-training & fine-tuning

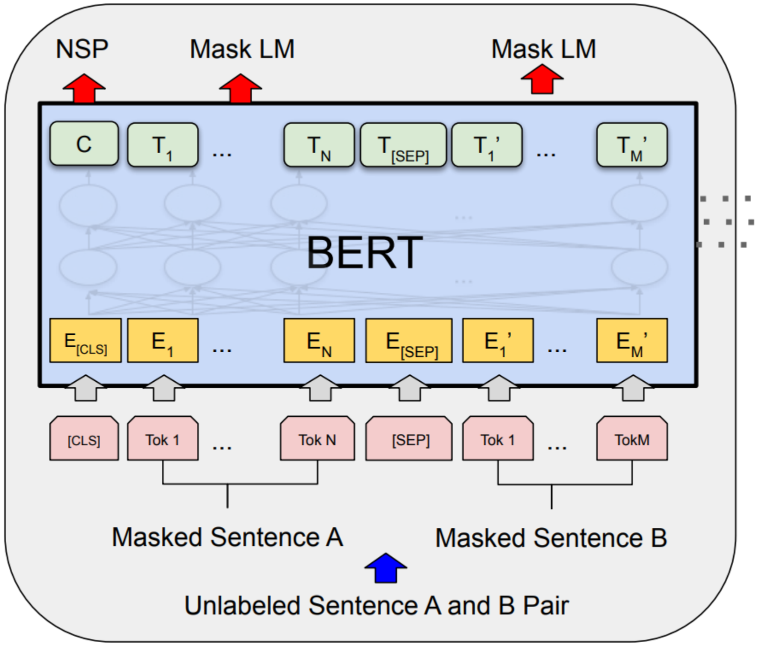
- 10% of the time: Keep the word unchanged

`my dog is hairy`

Devlin et al. 2019

Pre-Training BERT: MLM

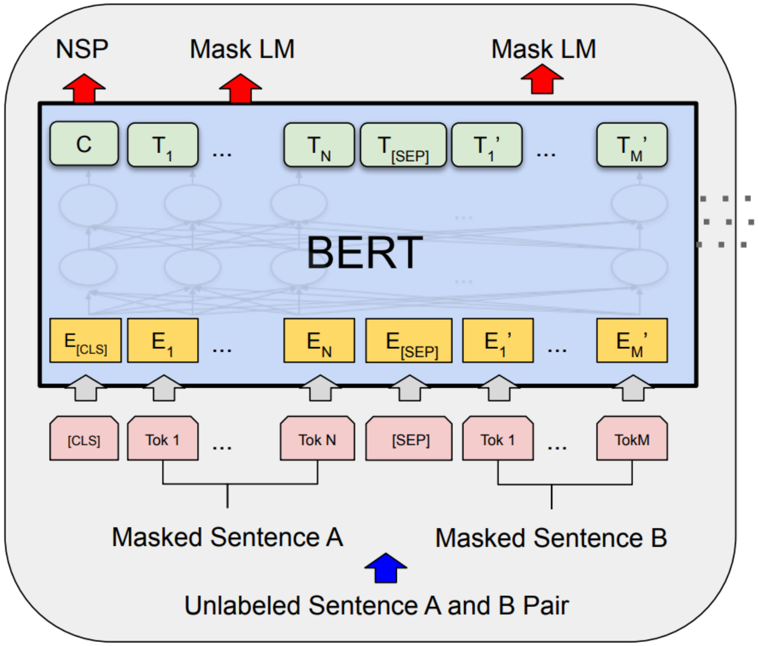
Idea: Predict vocab ID of masked tokens from final embeddings



Devlin et al. 2019

Pre-Training BERT: NSP

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

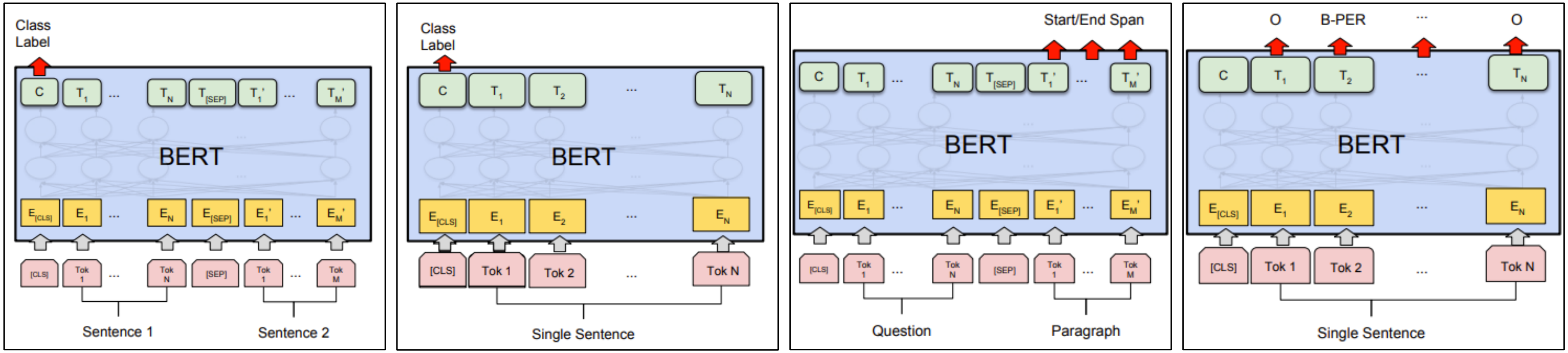


Devlin et al. 2019

Fine-Tuning

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

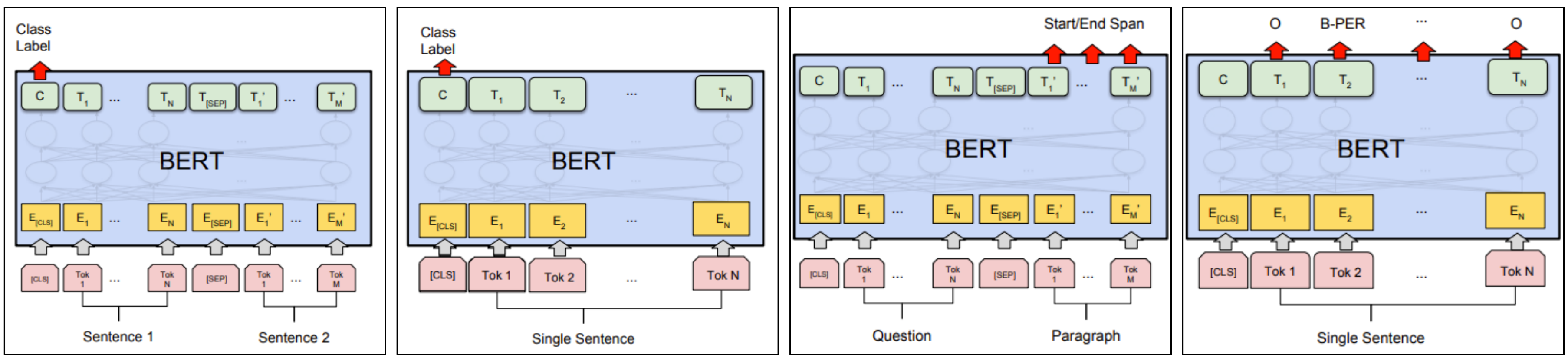
Change pre-training **output layers** of BERT to suit task



[Devlin et al. 2019](#)

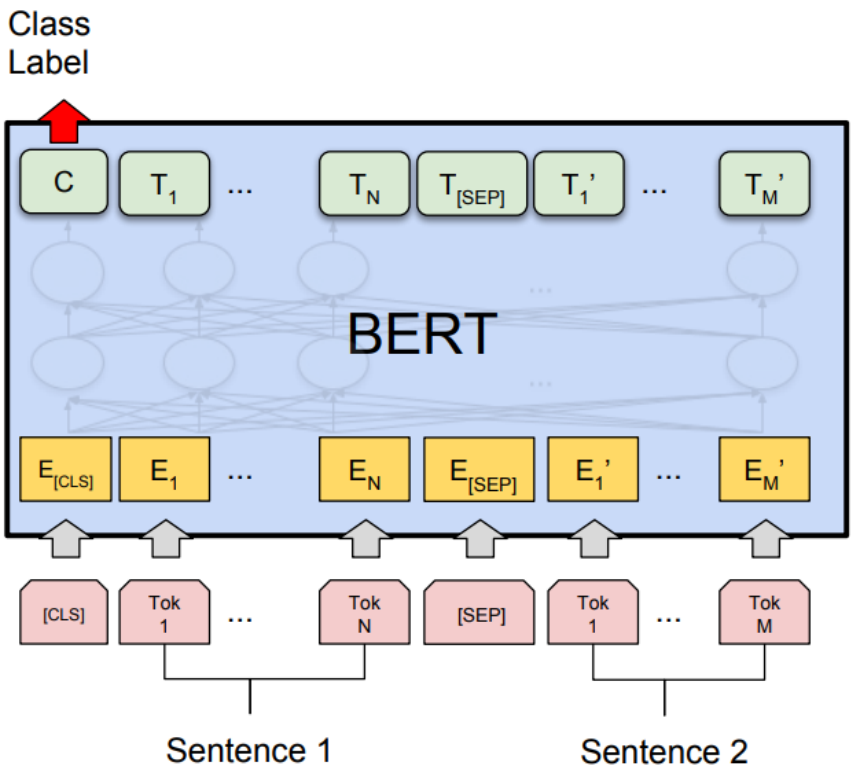
Fine-Tuning

Sentence Pair Classification	Single Sentence Classification	Question Answering	Single Sentence Tagging



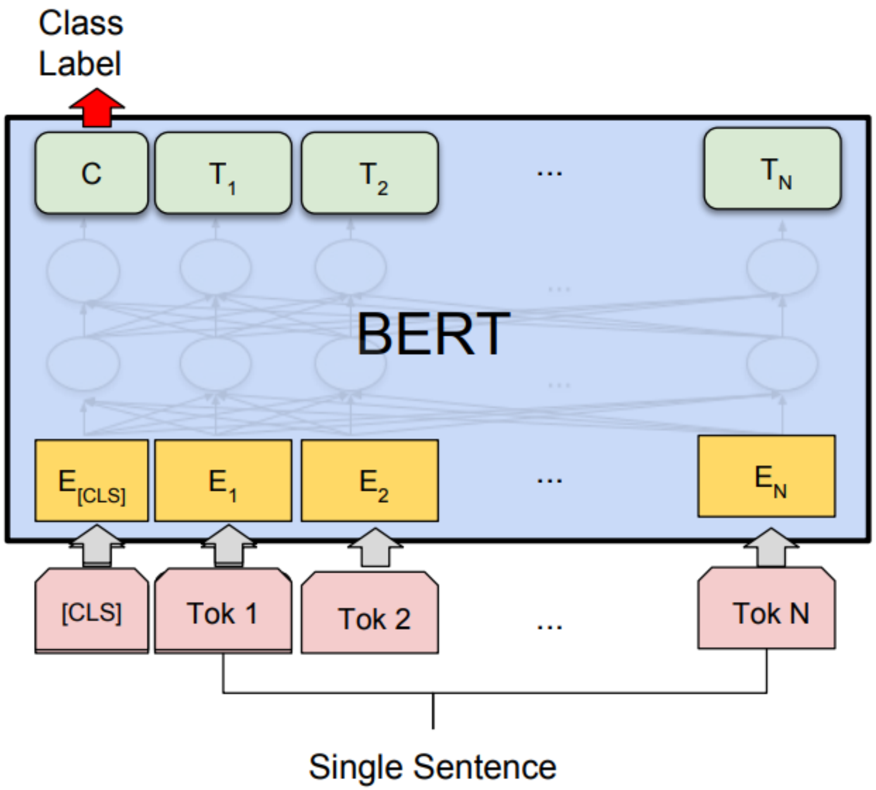
Devlin et al. 2019

Fine-Tuning: Sentence Pair Classification



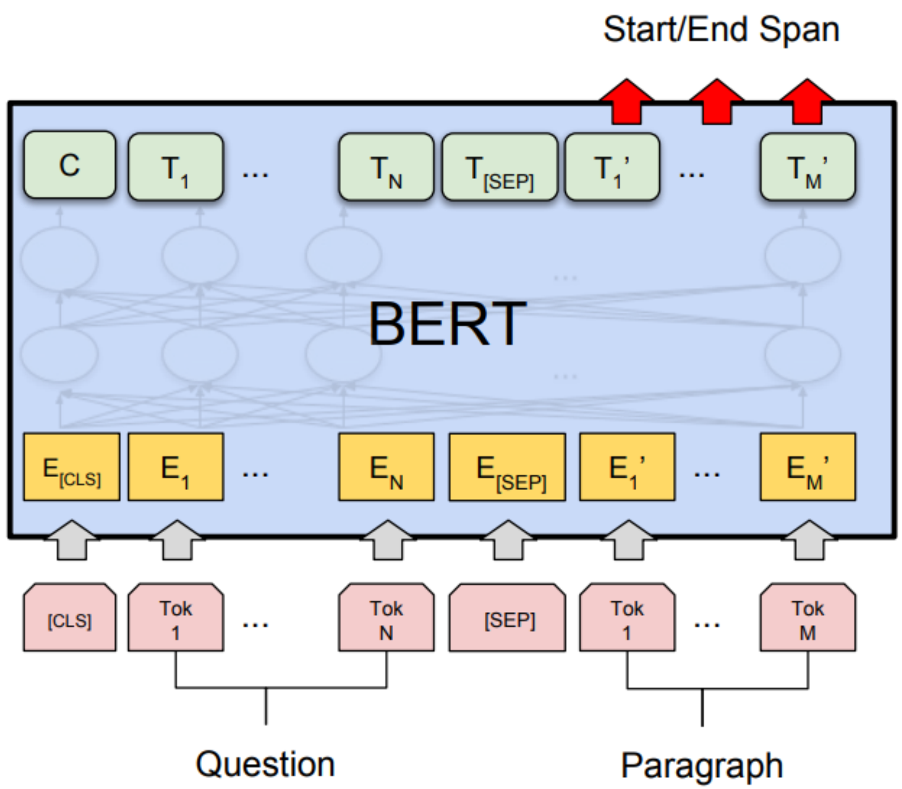
Devlin et al. 2019

Fine-Tuning: Single Sentence Classification



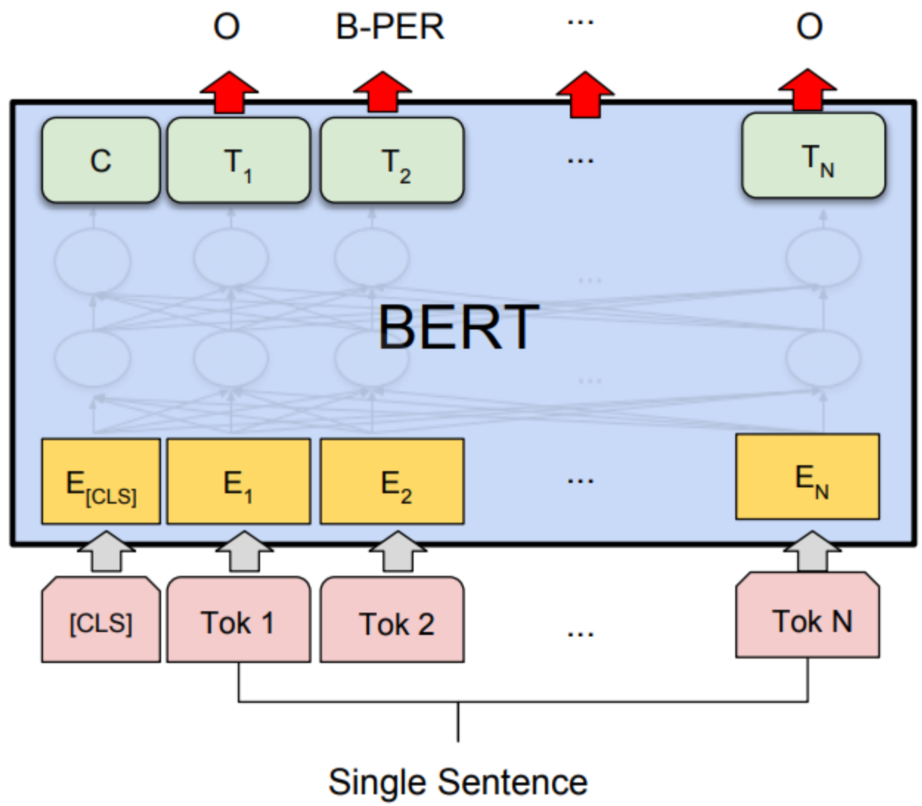
Devlin et al. 2019

Fine-Tuning: Question Answering



Devlin et al. 2019

Fine-Tuning: Single Sentence Tagging



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**

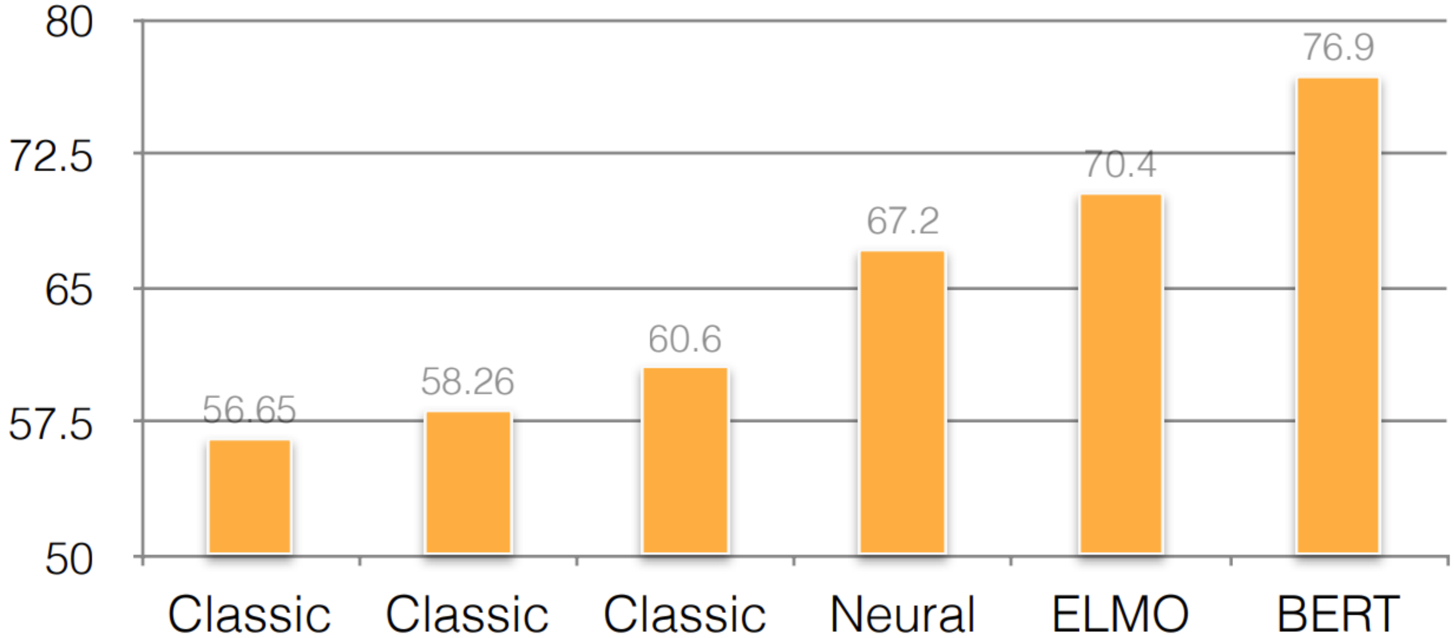
CoLA = Corpus of Linguistic Acceptability

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)

Representation learning:
extract attention features and use as input features to another model

Pretraining:
learn good representations via an unlabeled task.

Finetuning:
train some more on in-domain data or separate labeled task

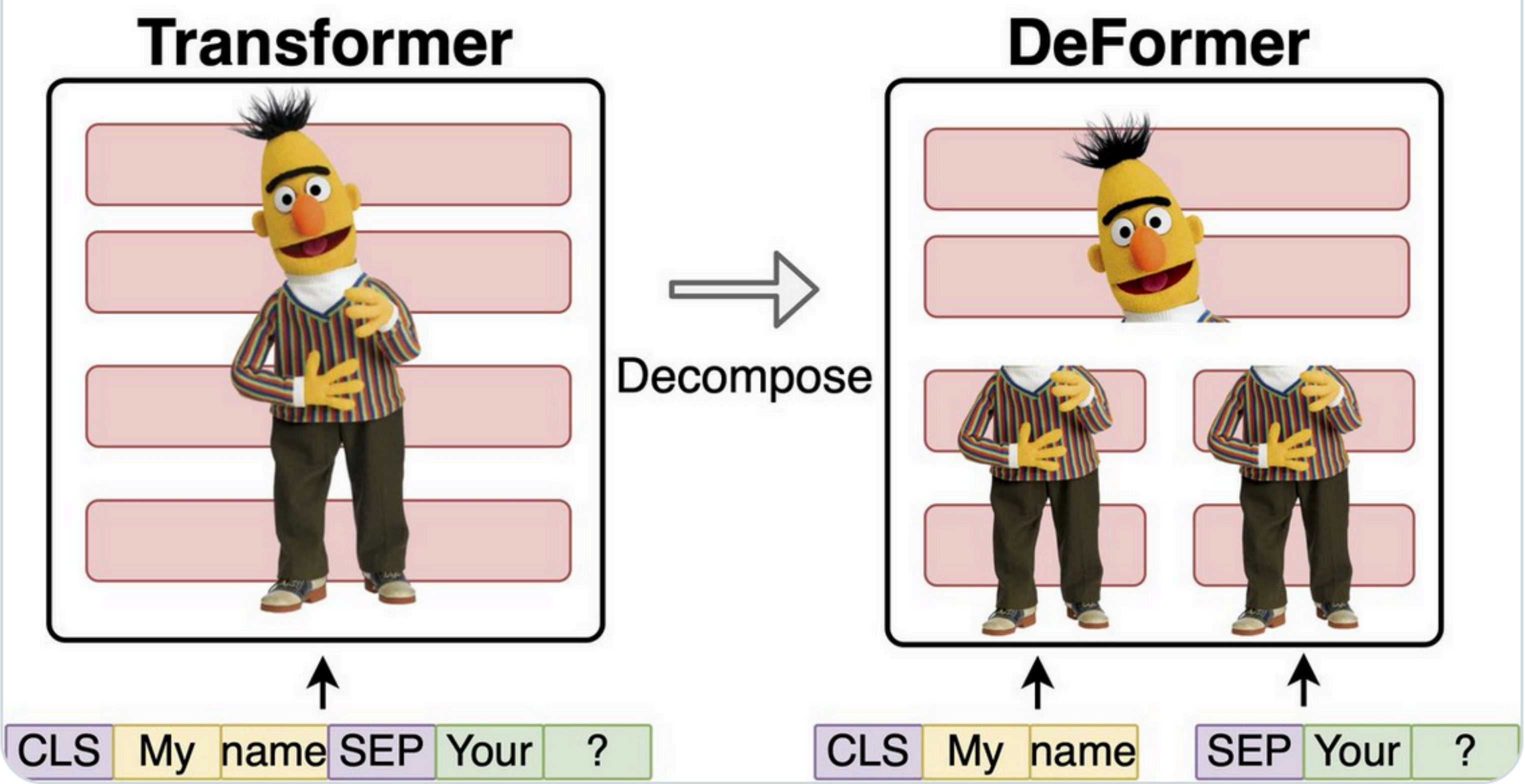
Prompt engineering:
craft prompts that disguise task of interest as a language generation problem.

Lucy Li
@lucy3_li

oh my god they decapitated bert

Qingqing Cao @sysnlp · May 4, 2020

Happy to share my first @aclmeeting paper: "DeFormer: Decomposing Pre-trained Transformers for Faster Question Answering" w/ @harsh3vedi, @aruna__b, and @b_niranjan. arxiv.org/abs/2005.00697. #acl2020nlp #NLP



12:58 AM · May 5, 2020

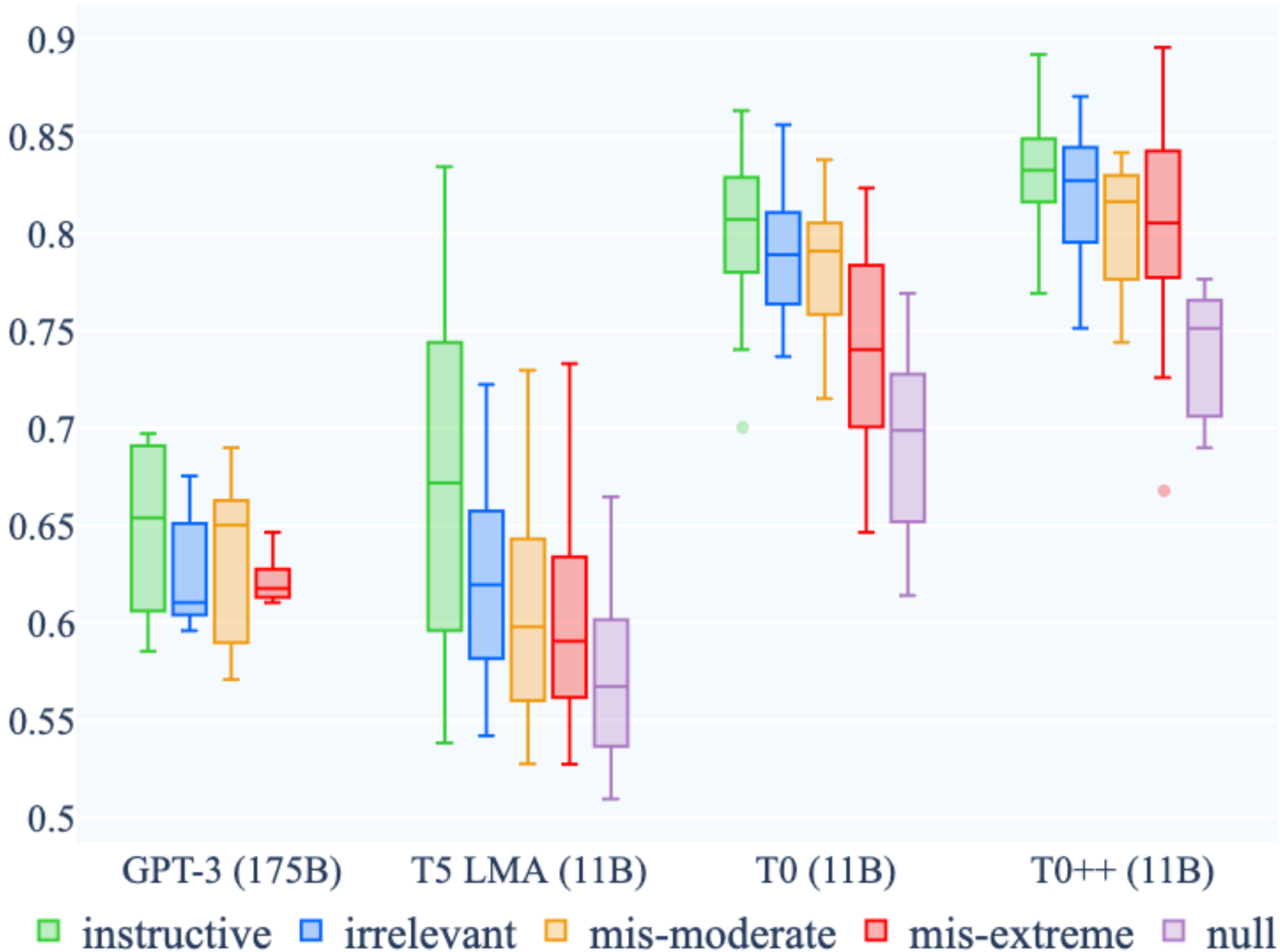
Prompt Engineering

Chain-of-Thought Reasoning

One idea is to make the model generate reasoning before an answer. This guarantees that the answer is conditioned on the reasoning. Some people think this could improve the quality of the answer. However, other work has shown that the answer is not always consistent with the given reasoning.

```
Question: Tom and Elizabeth have a competition to climb a
hill. Elizabeth takes 30 minutes to climb the hill. Tom
takes four times as long as Elizabeth does to climb the
hill. How many hours does it take Tom to climb up the hill?
Answer: It takes Tom 30*4 = <<30*4=120>>120 minutes to
climb the hill.
It takes Tom 120/60 = <<120/60=2>>2 hours to climb the
hill.
So the answer is 2.
===
Question: Jack is a soccer player. He needs to buy two
pairs of socks and a pair of soccer shoes. Each pair of
socks cost $9.50, and the shoes cost $92. Jack has $40.
How much more money does Jack need?
Answer: The total cost of two pairs of socks is $9.50 x 2
= $<<9.5*2=19>>19.
The total cost of the socks and the shoes is $19 + $92 =
$<<19+92=111>>111.
Jack need $111 - $40 = $<<111-40=71>>71 more.
So the answer is 71.
===
Question: Marty has 100 centimeters of ribbon that he must
cut into 4 equal parts. Each of the cut parts must be
divided into 5 equal parts. How long will each final cut
be?
```

What Are Prompts Really Doing?



Results from Webson & Pavlick (2022)

Does CoT Help?

Solving and Generating NPR Sunday Puzzles with Large Language Models

Jingmiao Zhao and Carolyn Jane Anderson
Computer Science Department
Wellesley College
Wellesley, MA 02482 USA
carolyn.anderson@wellesley.edu

Abstract

We explore the ability of large language models to solve and generate puzzles from the NPR Sunday Puzzle game show using PUZZLEQA, a dataset comprising 15 years of on-air puzzles. We evaluate four large language models using PUZZLEQA, in both multiple choice and free response formats, and explore two prompt engineering techniques to improve free response performance: chain-of-thought reasoning and prompt summarization. We find that state-of-the-art large language models can solve many PUZZLEQA puzzles: the best model, GPT-3.5, achieves 50.2% loose accuracy. However, in our few-shot puzzle generation experiment, we find no evidence that models can generate puzzles: GPT-3.5 generates puzzles with answers that do not conform to the generated rules. Puzzle generation remains a challenging task for future work.

Puzzle Description: Today's puzzle involves "consonyms," which are words that have the same consonants in the same order but with different vowels. Every answer is the name of a country.
Question: MINGLE
Answer: MONGOLIA

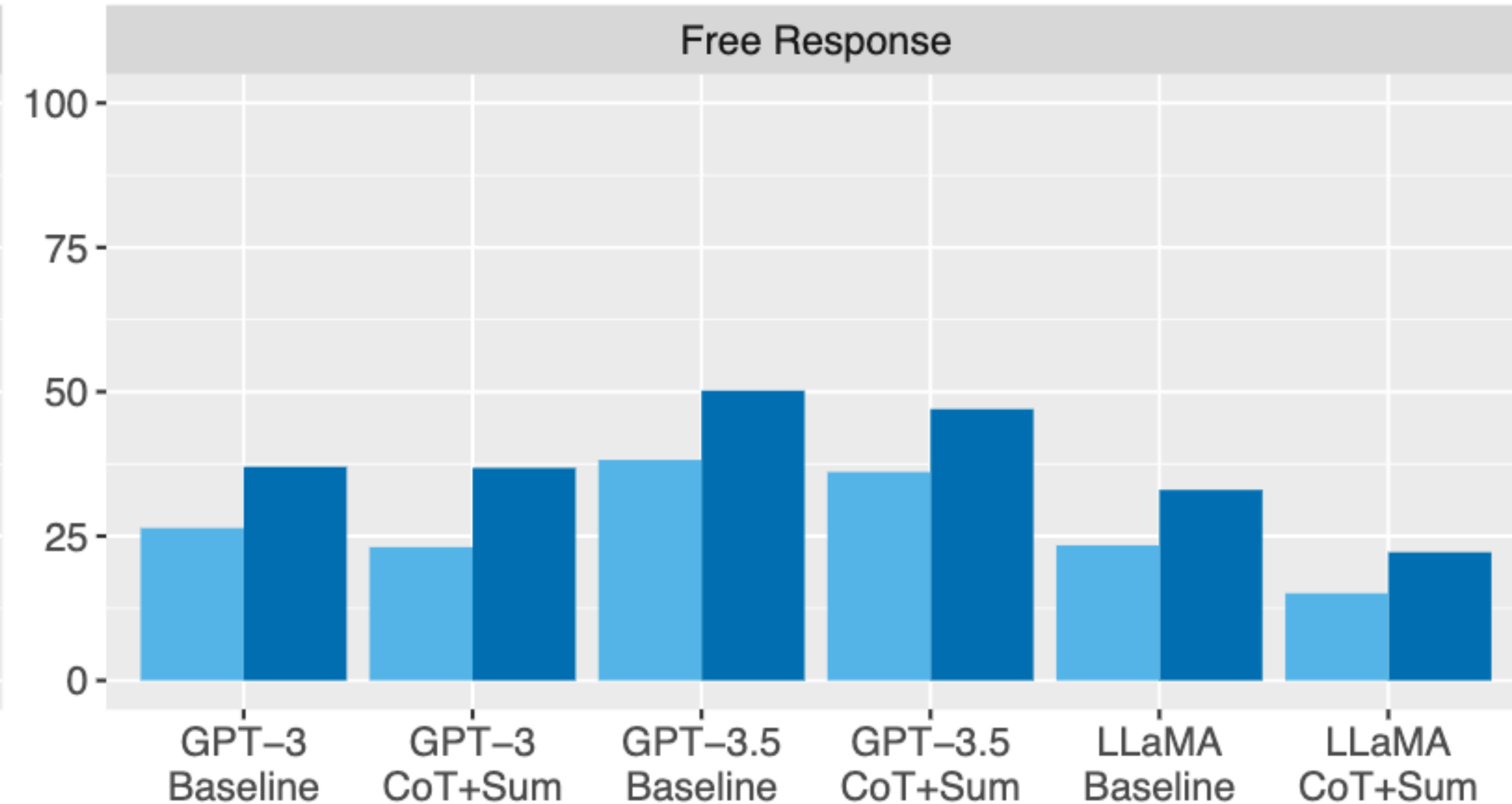
Figure 1: NPR Sunday Puzzle from March 12, 2023

Benchmarking AI through Games

Our work continues the tradition of evaluating AI progress through puzzles and games (Ferrucci 2012; Rodriguez et al. 2021; Rozner, Potts, and Mahowald 2021; Sobieszek and Price 2022). Contemporary LLMs have demonstrated strong

Does CoT Help?

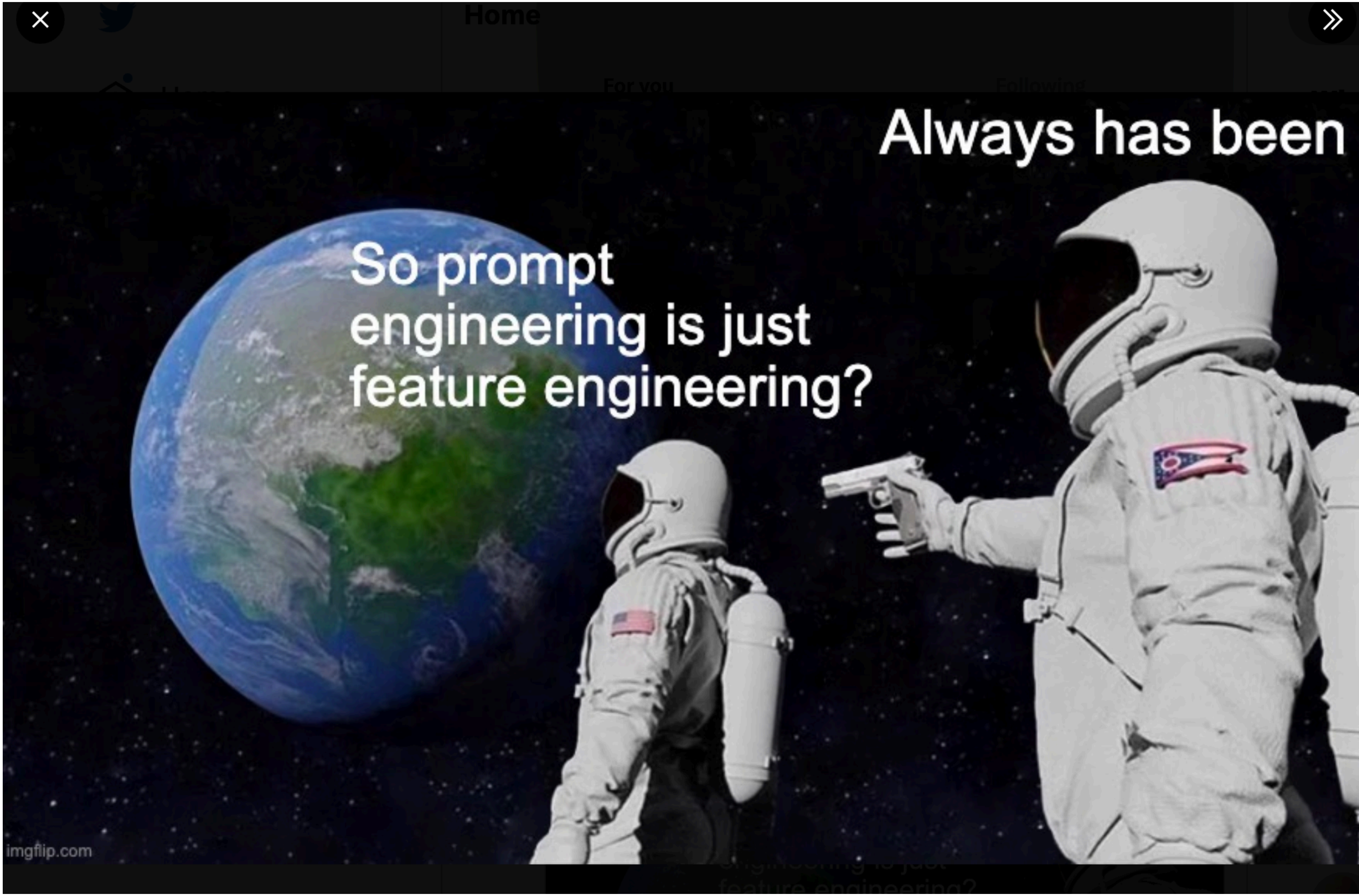
Maybe not?



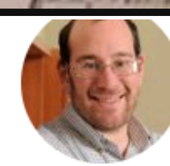
Continuous Prompting

Humans write discrete prompts, which are then turned into text embeddings.

What if we tried to directly **learn** good text embeddings?



Mark Dredze
@mdredze



Mark Dredze
@mdredze