CS678 Advanced Natural Language Processing

Contextual Representations & Pre-training

Ziyu Yao

https://nlp.cs.gmu.edu/course/cs678-fall22

Acknowledgement: Slides are adapted from Greg Durrett CS388@UT Austin and Antonios Anastasopoulos CS695@GMU

Reminder

- HW 2 is out this week (stay tuned!)
- Group Formation and Project Interest Survey Form due **Sept 16**

Outline

- Contextual Representations
- Pre-trained Language Models (PLMs)
  - ELMo, GPT-2, BERT
- Transfer Learning in NLP

Context-dependent Representations

- Context-*independent* word embeddings:
  - e.g., word2vec
  - Same representation of "*bank*" in "financial *bank*" and "river *bank*"
- Context-*dependent* word embeddings:
  - The representation of each word depends on the specific context it appears
  - Different representations of "*bank*" in "financial *bank*" and "river *bank*"

ELMo

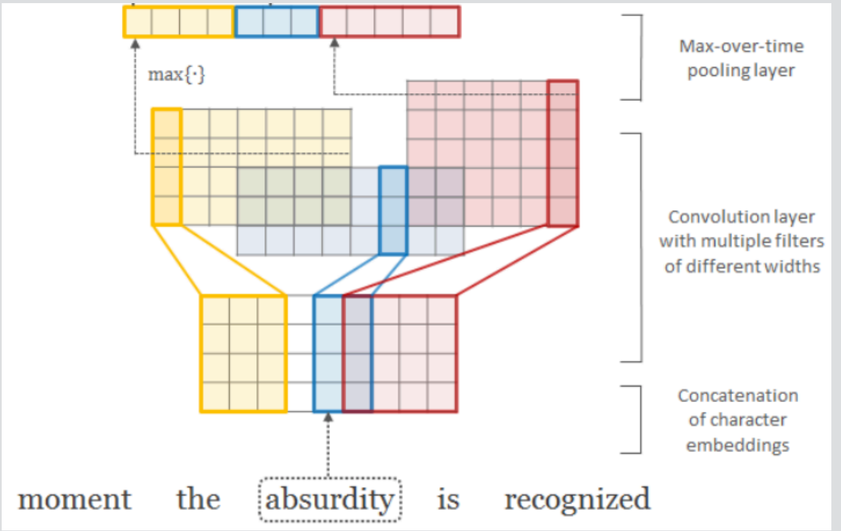
(Peters et al., 2018)

- ELMo: Embeddings from Language Models

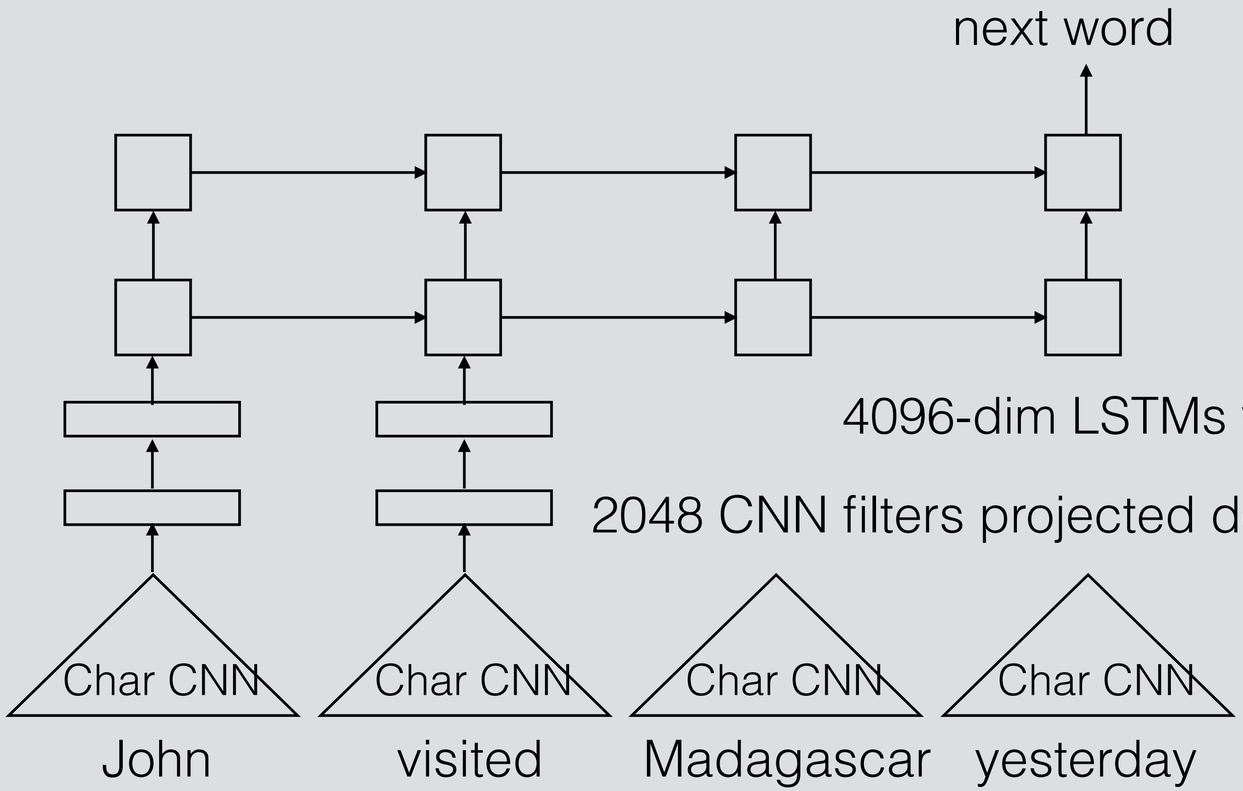
- Key characteristics:
  - *Contextual:* Word representation as a function of the entire input
  - *Deep* representations
    - A mixture of multiple layers of hidden states as the representation
    - Each layer capturing one aspect of word information
  - *Character-based*: modeling the morphological clues to form robust representations for training-time unseen words

ELMo - Pretraining

- CNN over each word => RNN



(Kim et al., 2015)



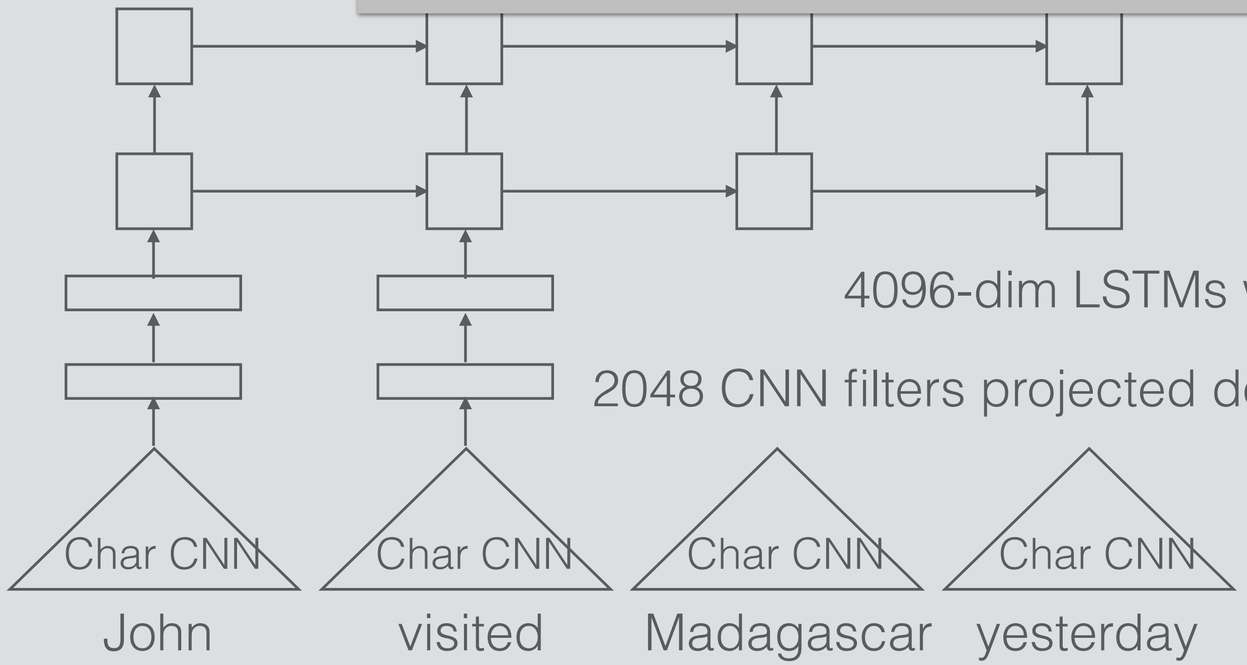
(Peters et al., 2018)

ELM

- CNN over each word =

Objective: jointly train the *forward* & *backward* LMs

$$\sum_{k=1}^{N} \Big(\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \Big)$$



(Peters et al., 2018)

ELMo - Contextual Representation

- *2L* + 1 representations from a *L*-layer ELMo
  - Bottom: the (context-independent) word embedding
  - Upper: hidden states from both directions in each layer
- Word representation = a *task-specific* weighted combination of different representations

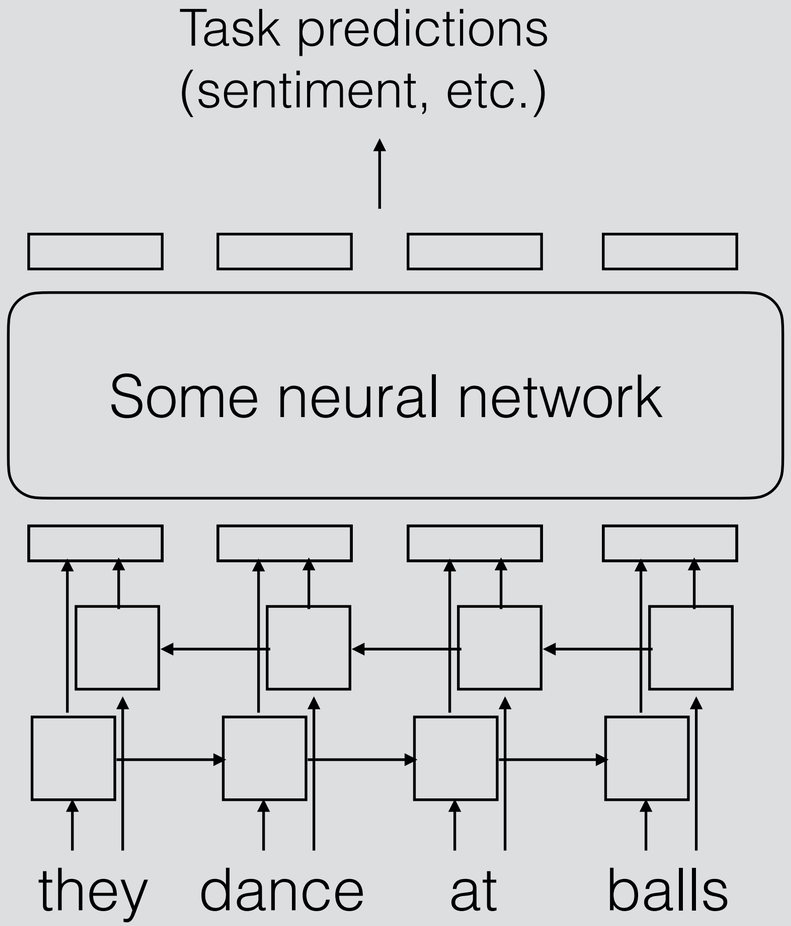
$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}$$

(For token t_k)

(Peters et al., 2018)

How to apply ELMo?

- Take those embeddings and feed them into whatever architecture you want to use for your task
  - e.g., concatenating EMLo^{task} w/ word vec
- Frozen embeddings: update the weights of your network but keep ELMo's parameters frozen
- Fine-tuning: backpropagate all the way into ELMo when training your model



Results

Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

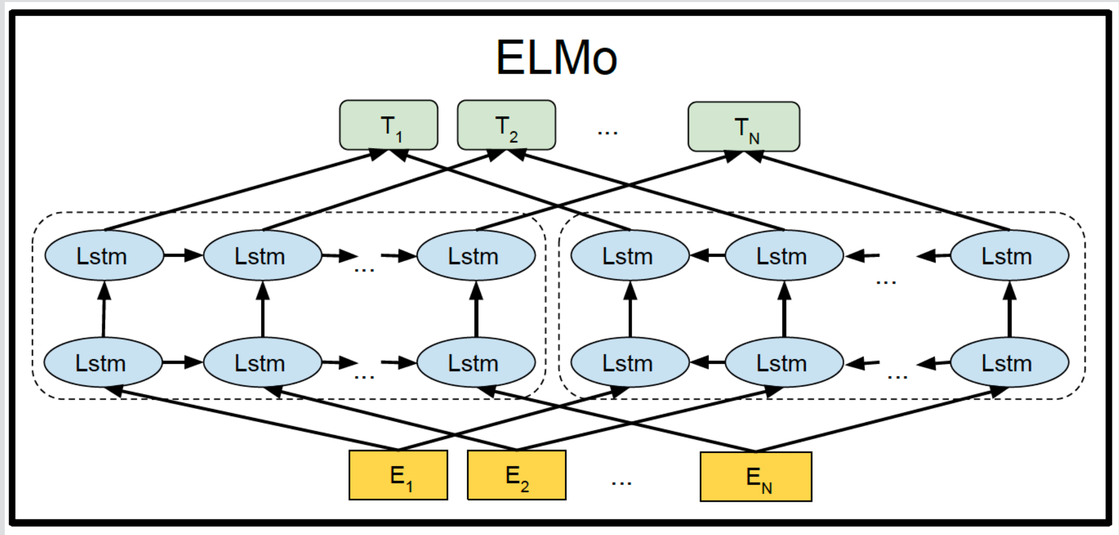
- Massive improvements across 5 benchmark datasets: question answering, natural language inference, semantic role labeling, coreference resolution, named entity recognition, and sentiment analysis

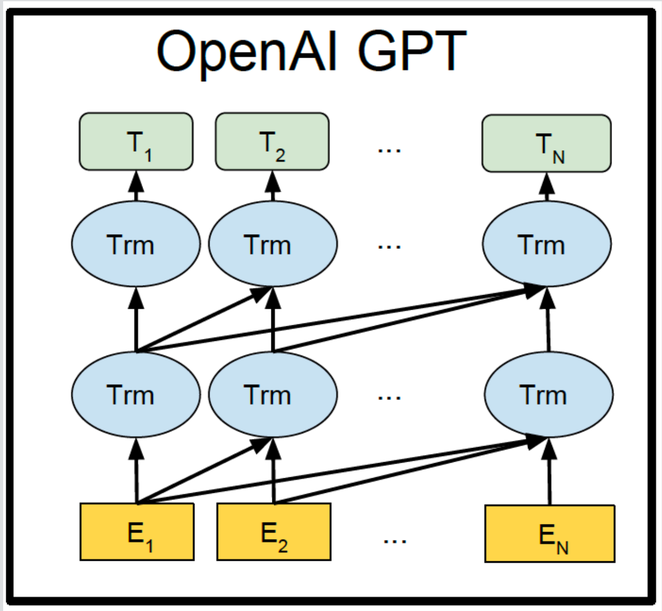
(Peters et al., 2018)

OpenAI GPT Generative Pre-Training

(Radford et al. 2018)

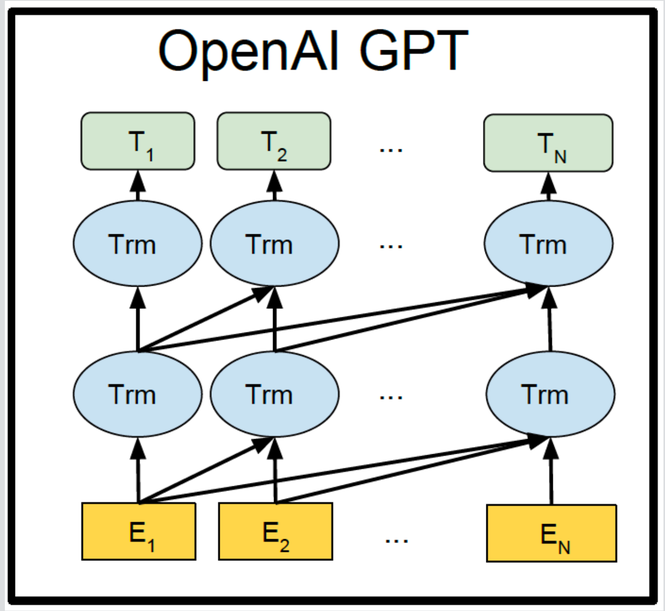
- Similar as ELMo, using (unidirectional) language modeling as pre-training objective
  - But only one direction: left to right
  - Also, using Transformer rather than LSTM





OpenAI GPT

(Radford et al. 2018)

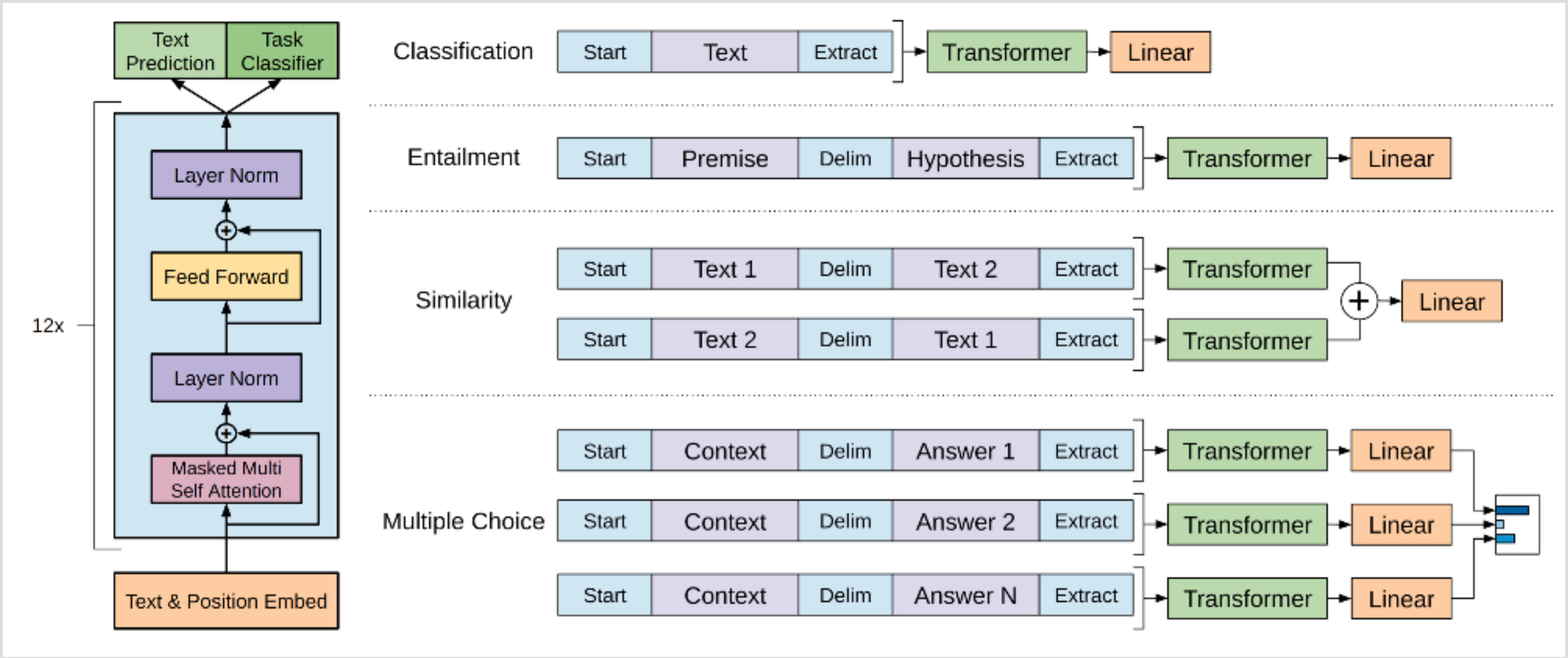


- Similar as ELMo, using (unidirectional) language modeling as pre-training objective
  - But only one direction: left to right
  - Also, using Transformer rather than LSTM
- Pre-training + *fine-tuning*
  - Applying the same model to downstream tasks by slightly *tuning* the pre-trained parameters
    - Output state of the last token as **sentence representation**
  - In contrast, ELMo uses the pre-trained representations as ***features (contextualized word embeddings)*** and trains a new downstream model

OpenAI GPT

(Radford et al. 2018)

- Task-specific fine-tuning (using representation from the final transformer block)



OpenAI GPT-2
(Radford et al., 2019)

- GPT-2 is a direct scale-up of GPT, with more than 10X the parameters and trained on more than 10X the amount of data.
  - 1.5 billion parameters
- State-of-the-art performance in *“zero shot”* setting (when it came out)
  - i.e., using the pre-trained checkpoint, not fine-tuned on task-specific dataset
  - From “single task, single dataset” to “multiple tasks, multiple datasets”
  - Why it matters? Data, compute, physical storage, etc.
  - *(Note that GPT-2 can definitely be fine-tuned, just like GPT)*

https://openai.com/blog/better-language-models/#sample2 & “Policy Implications”

OpenAI GPT2

(Radford et al., 2019)

SYSTEM PROMPT (HUMAN-WRITTEN)	*A train carriage containing controlled nuclear materials was stolen in Cincinnati today. Its whereabouts are unknown.*
MODEL COMPLETION (MACHINE-WRITTEN, FIRST TRY)	The incident occurred on the downtown train line, which runs from Covington and Ashland stations. In an email to Ohio news outlets, the U.S. Department of Energy said it is working with the Federal Railroad Administration to find the thief. “The theft of this nuclear material will have significant negative consequences on public and environmental health, our workforce and the economy of our nation,” said Tom Hicks, the U.S. Energy Secretary, in a statement. “Our top priority is to secure the theft and ensure it doesn’t happen again.” The stolen material was taken from the University of Cincinnati’s Research Triangle Park nuclear research site, according to a news release from Department officials. The Nuclear Regulatory Commission did not immediately release any information. According to the release, the U.S. Department of Energy’s Office of Nuclear Material Safety and Security is leading that team’s investigation. “The safety of people, the environment and the nation’s nuclear stockpile is our highest priority,” Hicks said. “We will get to the bottom of this and make no excuses.

Slide credit: OpenAI

Zero-shot Experiments

Word prediction based on long-term dependencies

Cloze test given a children book story, by ans word categories (Common Nouns, Named Entities)

Commonly used text corpora for LM evaluation

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

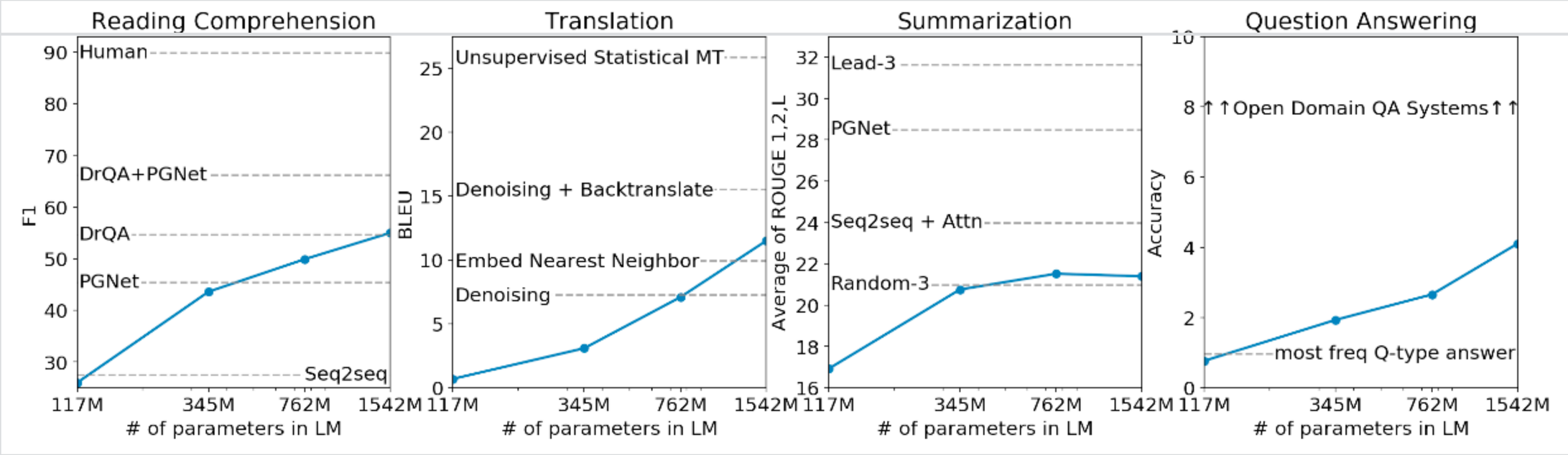
Although still fall short in…

Given a doc and conv history, prompted by “A:” (CoQA)

Prompted by examples of “english sentence = french sentence” followed by the tested eng sent

Given a news article, prompted by “TL;DR:”

Prompted by a few QA pairs followed by the tested Q



Discussion

- Using language modeling as training objective
  - Successfully predicting next words requires modeling different effects in text
  - "*Unsupervised* learning": large training corpus *without* human annotations
  - Not a new idea, but succeeds with: large data, right architecture, hyper-parameter tuning
- The paradigm of pre-training and then fine-tuning
  - Essentially doing ***transfer learning (our topic in the second half)***
  - Transferring knowledge learned in the pre-training phase (typically abundant data) to the downstream task in the fine-tuning phase (typically fewer or *no* training data)

Discussion

- (ELMo) Feature-based
  - Plug and play
  - Work with existing task models: both good (widely applicable) and bad (still have to learn task-specific models on task-specific training data)
- (GPTs) Fine-tuning based
  - Few randomly initialized parameters
    - Good for small data
  - No architecture search (significantly weaken the need for task-specific architecture, with knowledge transferred in the self-attention blocks)
  - Issue: pre-training fine-tuning mismatch

(Devlin et al., 2018)

BERT

(Devlin et al., 2018)

- AI2 released ELMo in spring 2018, GPT was released in summer 2018, BERT came out October 2018

- Three major changes compared to ELMo:
  - Transformers instead of LSTMs (transformers in GPT as well)
  - Bidirectional <=> *Masked LM* objective instead of standard LM
  - Fine-tune instead of freeze at test time

NAACL 2018 Best Paper Presentation: https://vimeo.com/365139010

BERT

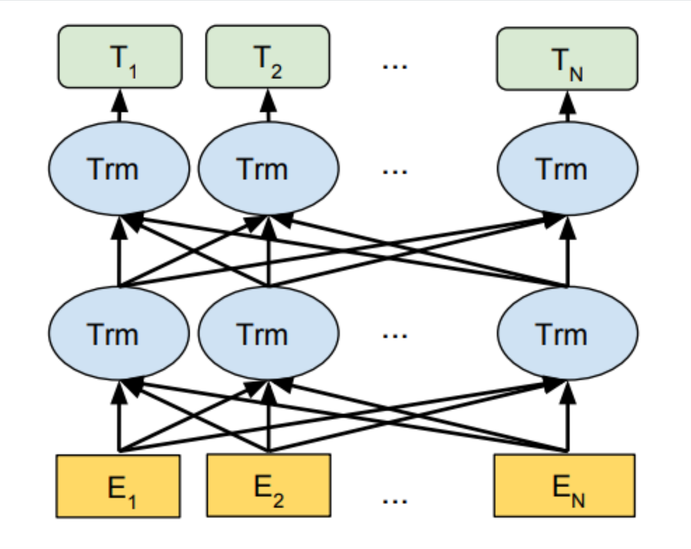
(Devlin et al., 2018)

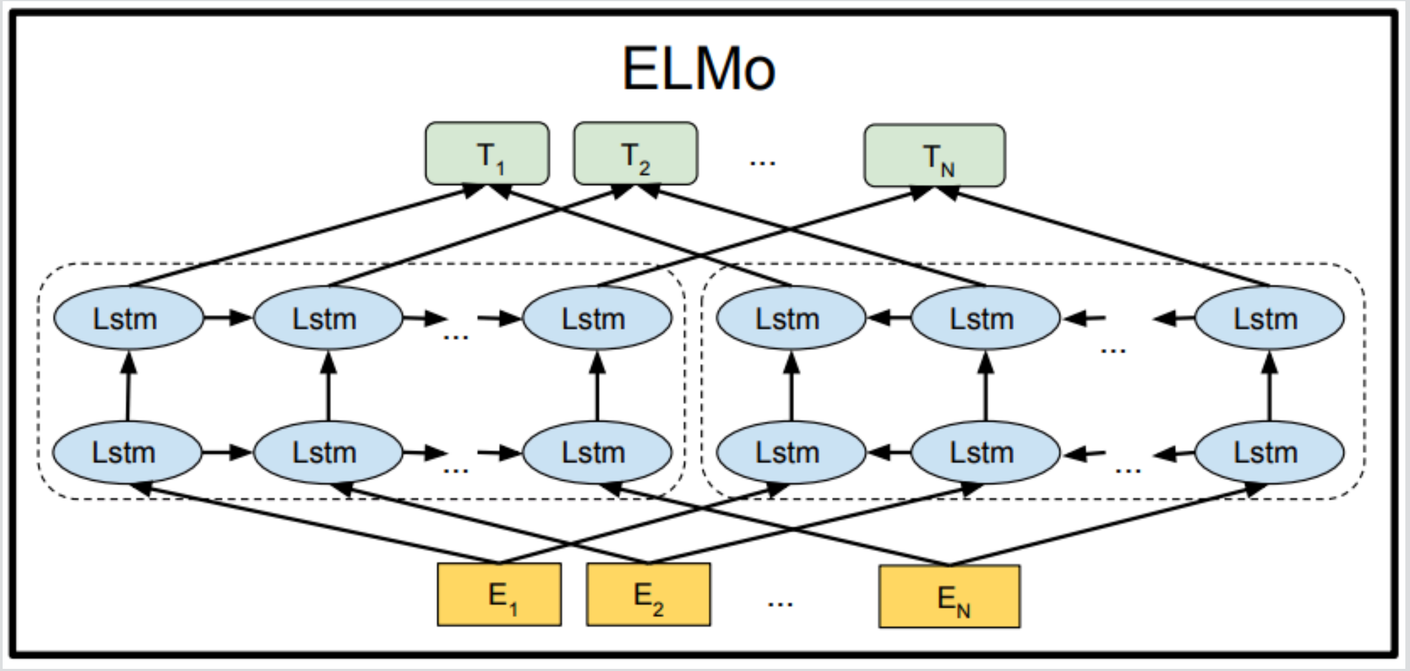
- Design logic: from usage need (fine-tuning) to pre-training strategies, i.e., “goal driven”
- What do we need for understanding NL? Think about possible downstream tasks!
  - Single-sentence classification
  - Sentence pair classification
  - Sequence tagging
  - Need model the context from both directions & attention => unidirectional LM is not sufficient!

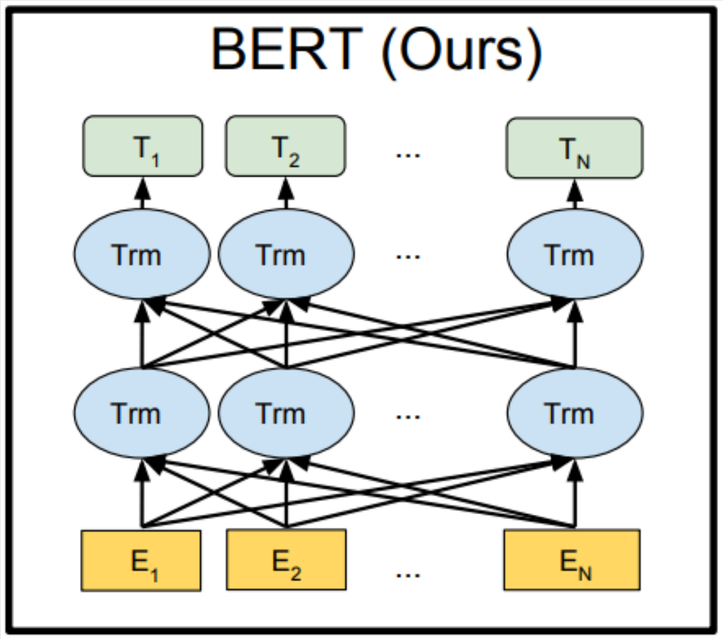
BERT

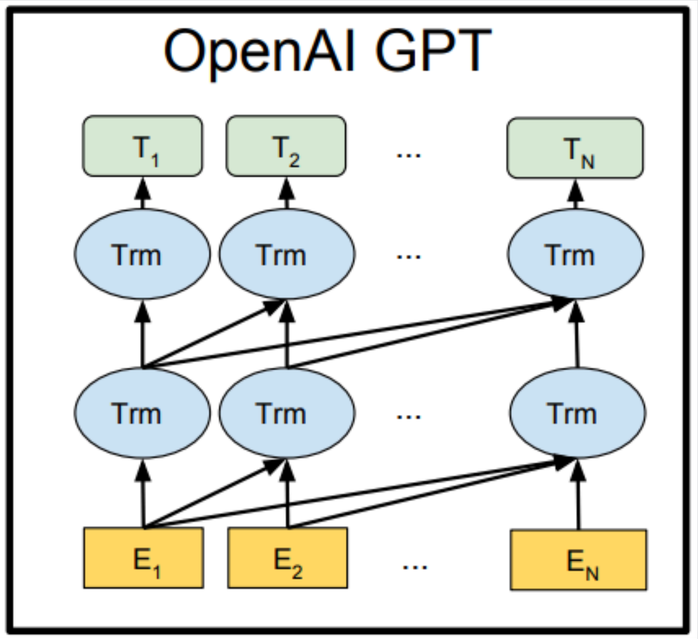
(Devlin et al., 2018)

- BERT = Bidirectional Encoder Representations from Transformers
  - Note that both ELMo and GPT model unidirectional contexts
  - With Transformer units







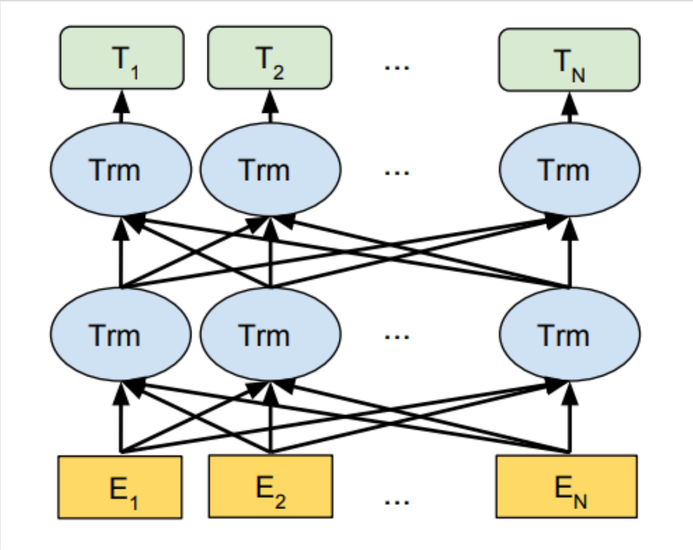


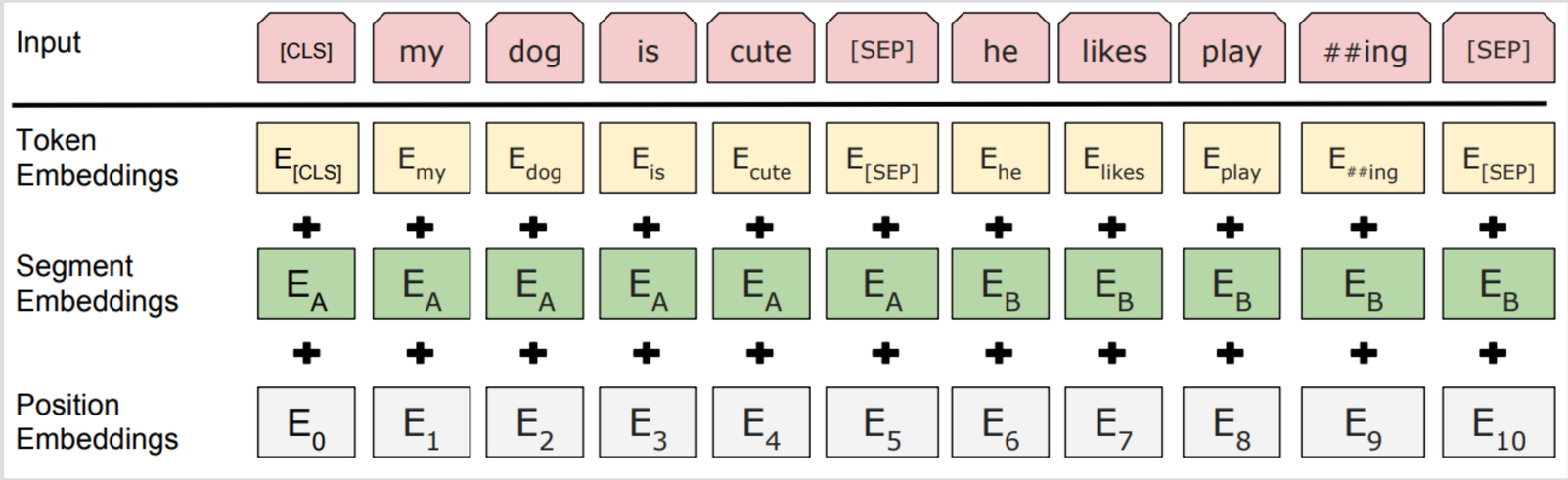
(Devlin et al., 2018)

BERT

(Devlin et al., 2018)

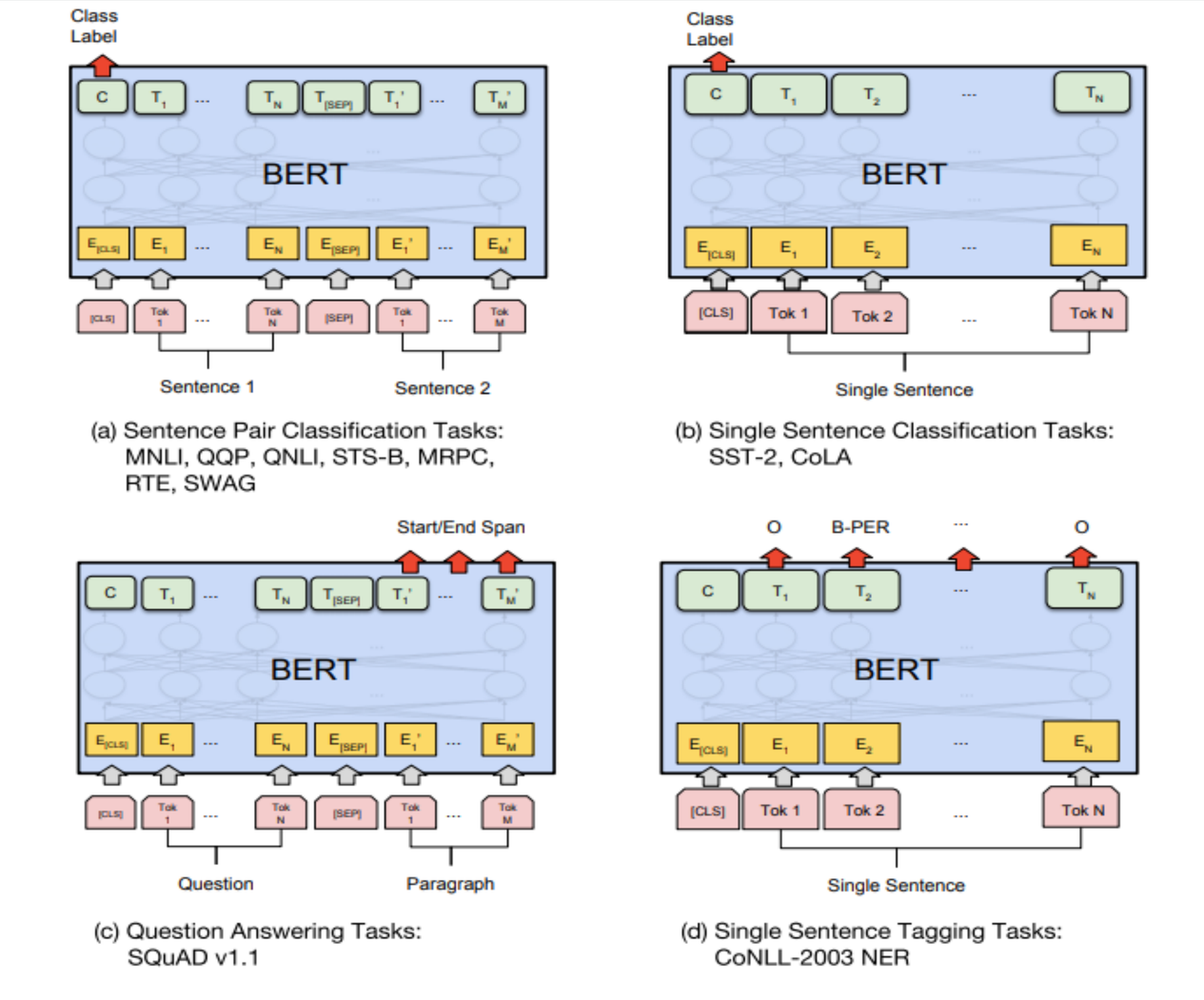
- BERT = Bidirectional Encoder Representations from Transformers
  - Note that both ELMo and GPT model unidirectional contexts
  - With Transformer units; subword representation; [CLS] and [SEP]





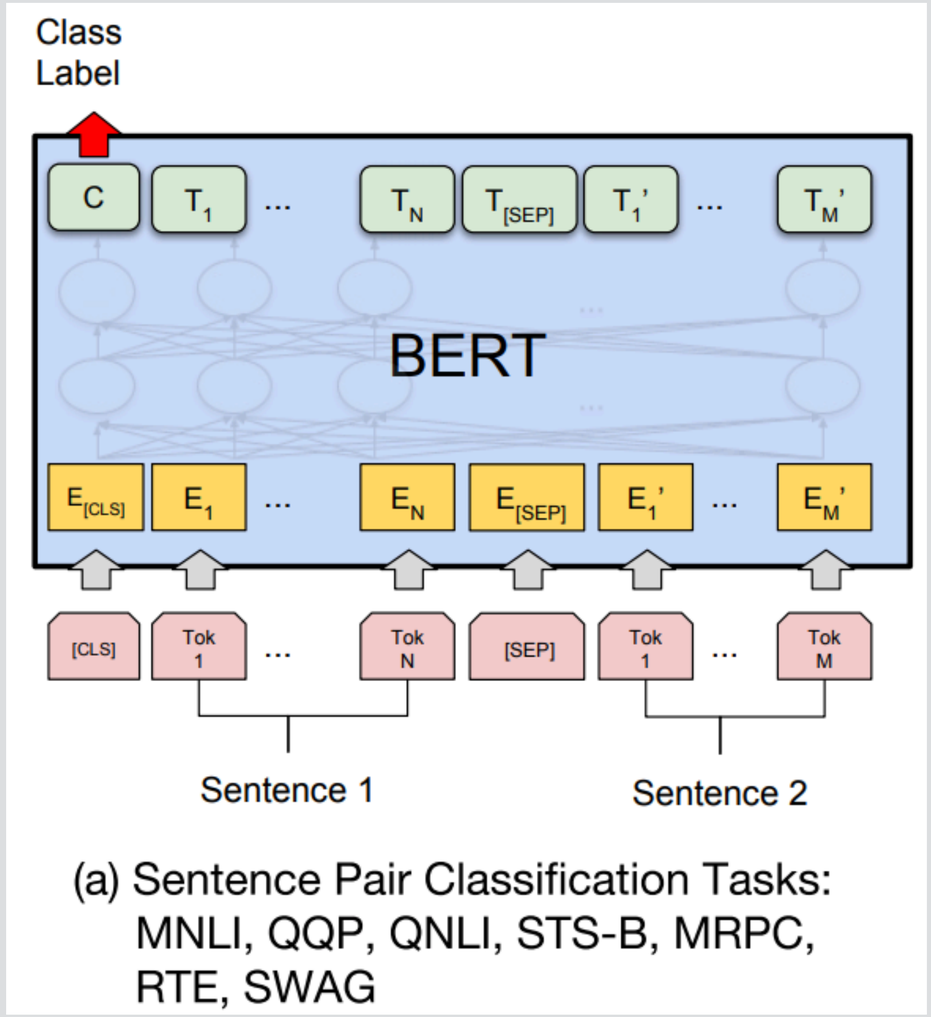
Using BERT

- Use the pre-trained model as the first “layer” of the final model, then train on the desired task



Using BERT

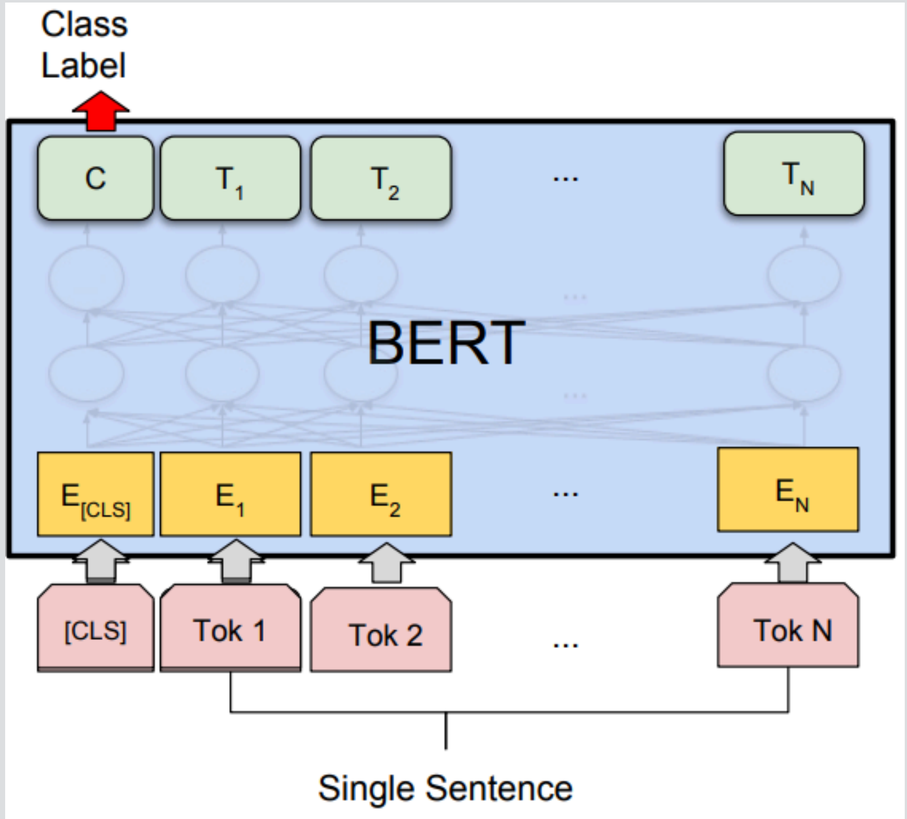
- Use the pre-trained model as the first “layer” of the final model, then train on the desired task



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

Using BERT

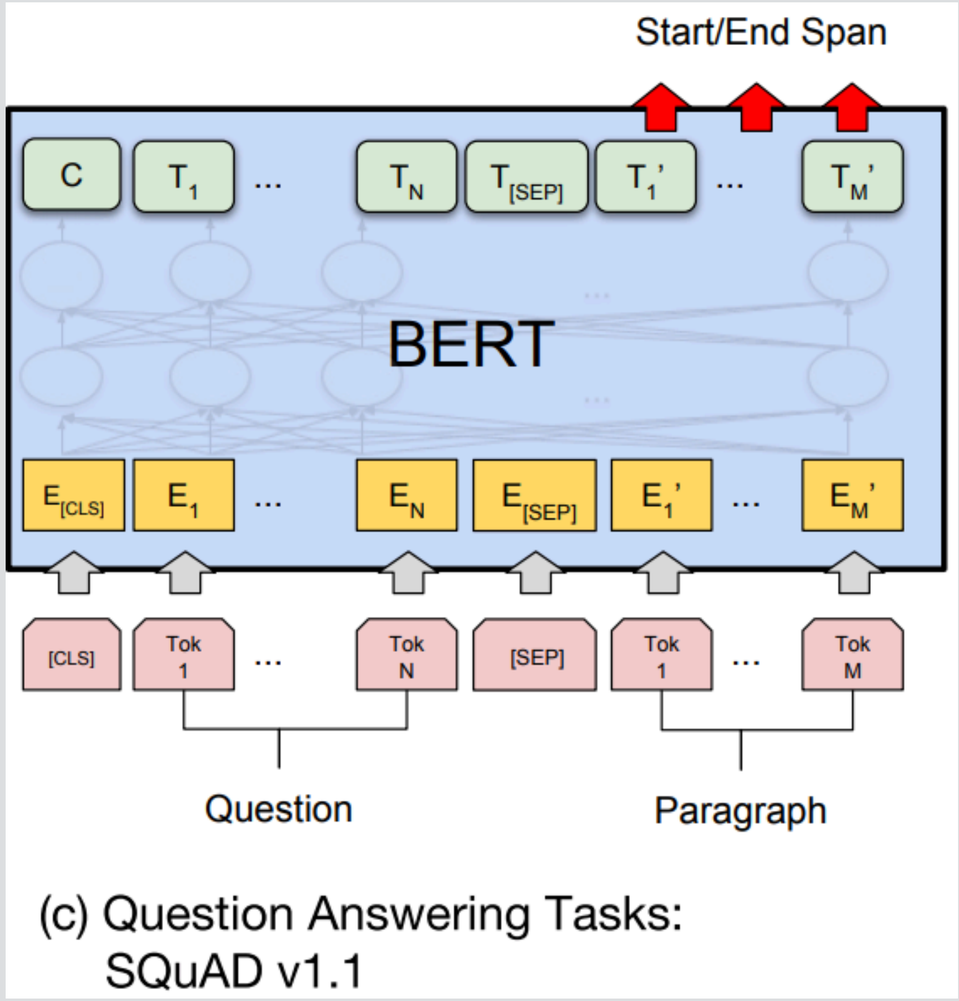
- Use the pre-trained model as the first “layer” of the final model, then train on the desired task



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

Using BERT

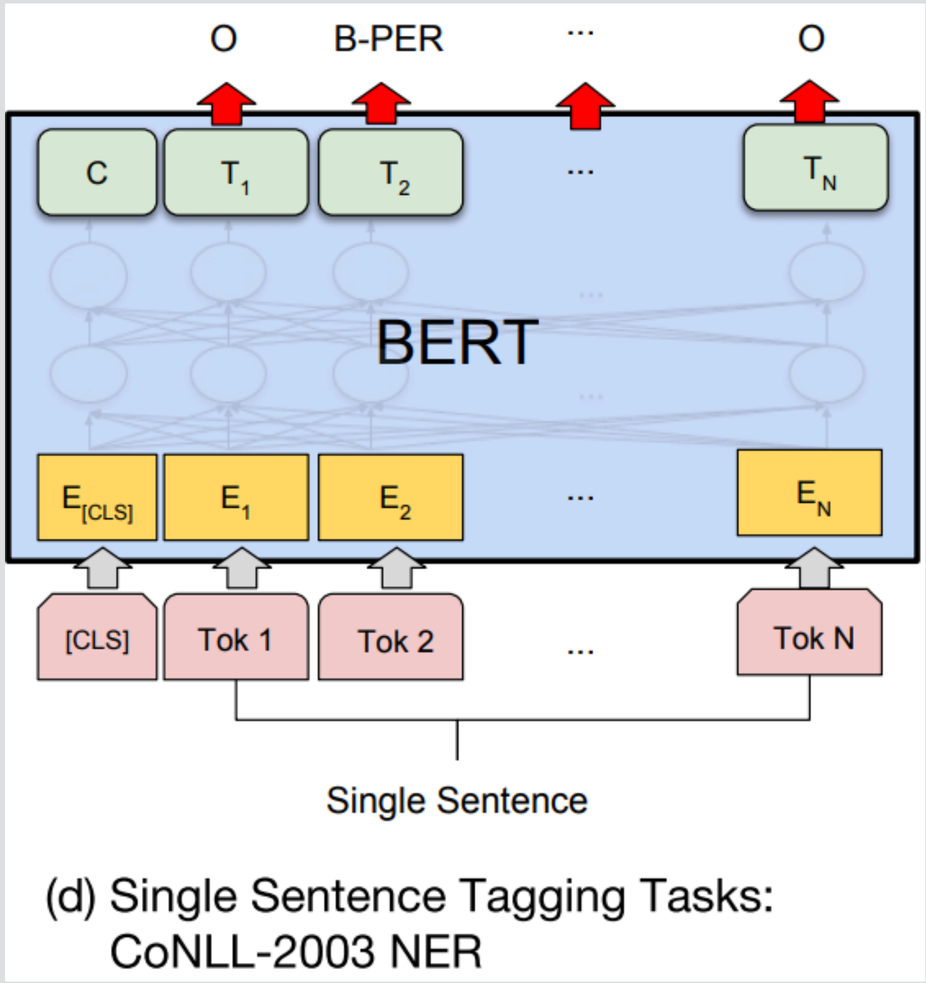
- Use the pre-trained model as the first “layer” of the final model, then train on the desired task



(c) Question Answering Tasks: SQuAD v1.1

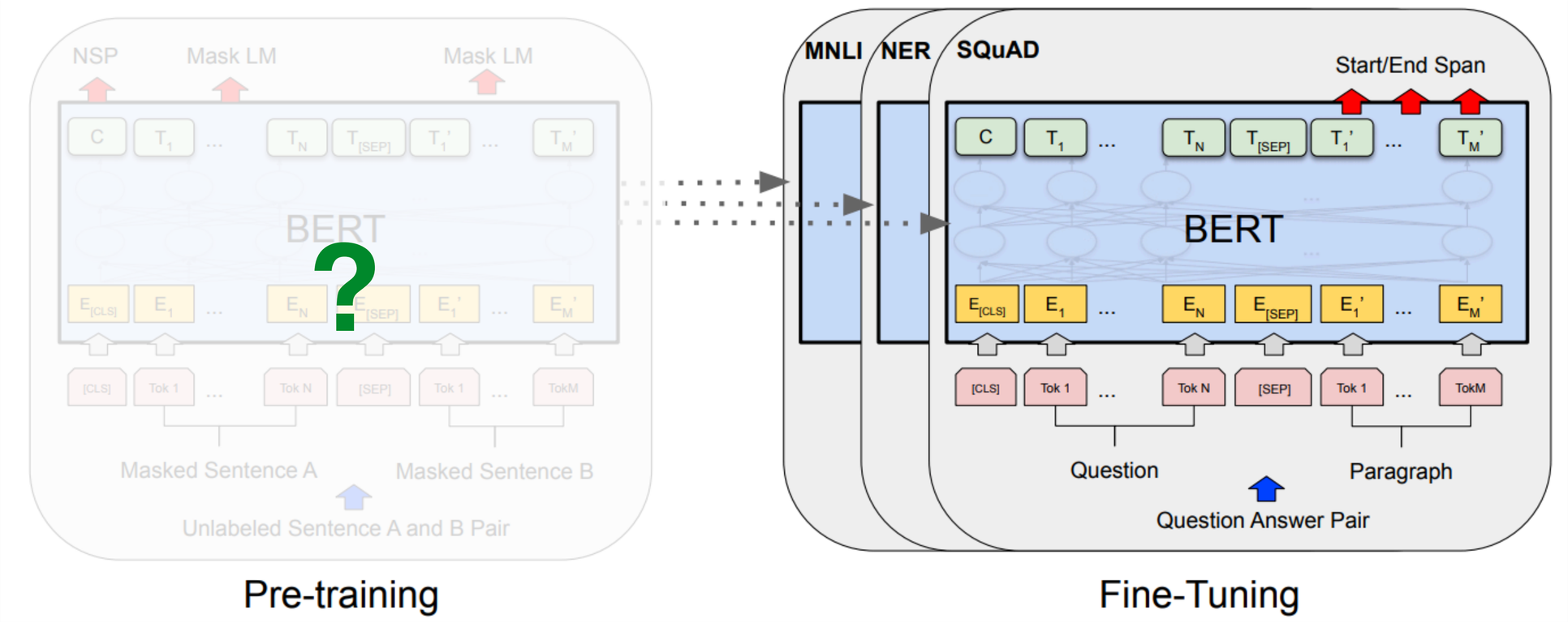
Using BERT

- Use the pre-trained model as the first “layer” of the final model, then train on the desired task



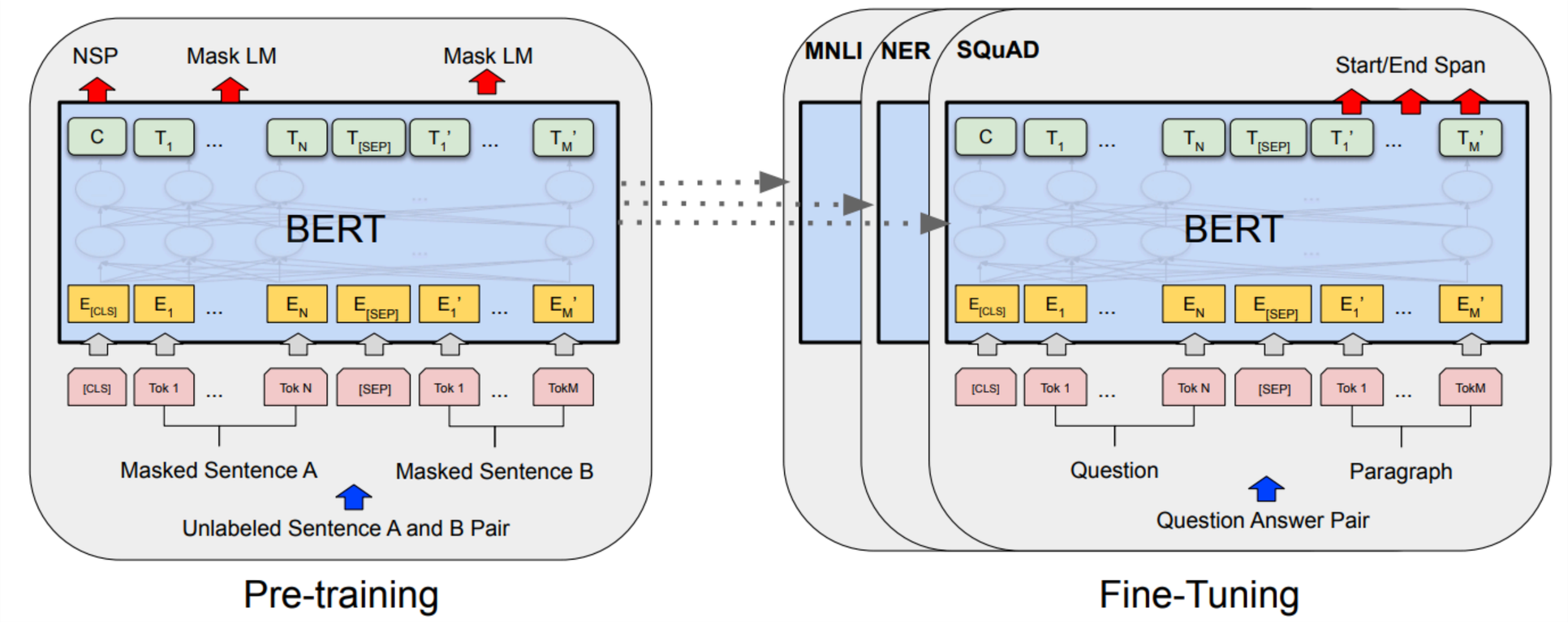
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Pre-train then Fine-tune



Note: all parameters are fine-tuned

Pre-train then Fine-tune



Note: all parameters are fine-tuned

Masked Word Prediction

(Devlin et al. 2018)

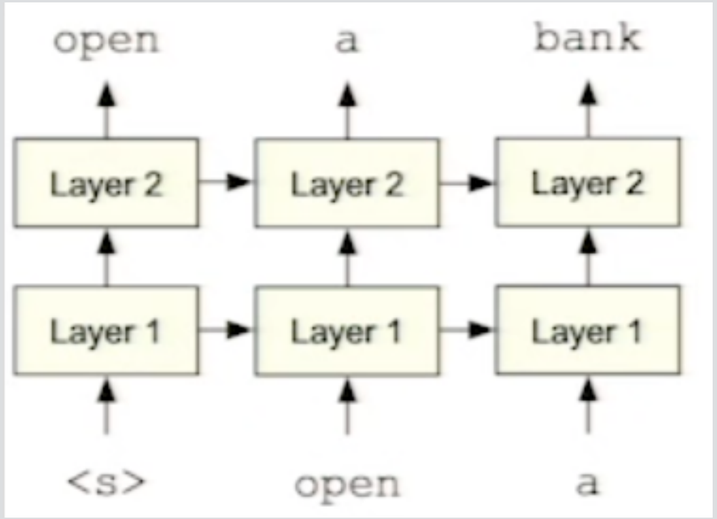
- Predict a masked word
  - 15% of all word piece tokens in each sentence is selected at random
  - And then,
    - 80%: substitute input word with [MASK]
    - 10%: substitute input word with random word
    - 10%: no change
- “Cloze” style

Can we not use [MASK]?

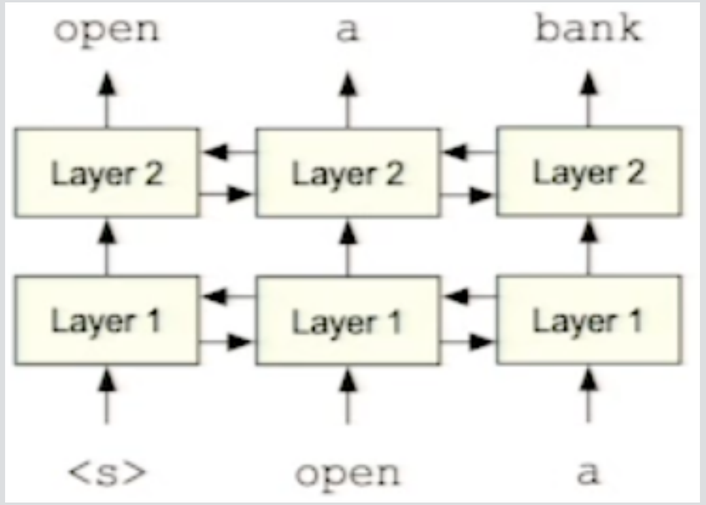
Masked Word Prediction

(Devlin et al. 2018)

Unidirectional context



Bidirectional context



Issue: words can see themselves

Consecutive Sentence Prediction

(Devlin et al. 2018)

- Classify two sentences as consecutive or not
  - 50% of training data is "consecutive"

Input = `[CLS] the man [MASK] to the store [SEP]`
`penguin [MASK] are flight ##less birds [SEP]`
Label = `NotNext`

Input = `[CLS] the man went to [MASK] store [SEP]`
`he bought a gallon [MASK] milk [SEP]`
Label = `IsNext`

BERT Pretraining

- Pretraining data:
  - BooksCorpus (800M words) + English Wikipedia (2,500M words)
  - Only text, no tables, headers, etc.
  - Document-level (rather than sentence-level) corpus
- Model configurations:
  - BERTbase: 110M parameters
  - BERTlarge: 340M parameters

	H=128	H=256	H=512	H=768
L=2	**2/128 (BERT-Tiny)**	2/256	2/512	2/768
L=4	4/128	**4/256 (BERT-Mini)**	**4/512 (BERT-Small)**	4/768
L=6	6/128	6/256	6/512	6/768
L=8	8/128	8/256	**8/512 (BERT-Medium)**	8/768
L=10	10/128	10/256	10/512	10/768
L=12	12/128	12/256	12/512	**12/768 (BERT-Base)**

GLUE Benchmark

Corpus	|Train|	|Test|	Task	Metrics	Domain
			Single-Sentence Tasks		
CoLA	8.5k	**1k**	acceptability	Matthews corr.	misc.
SST-2	67k	1.8k	sentiment	acc.	movie reviews
			Similarity and Paraphrase Tasks		
MRPC	3.7k	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	**391k**	paraphrase	acc./F1	social QA questions
			Inference Tasks		
MNLI	393k	**20k**	NLI	matched acc./mismatched acc.	misc.
QNLI	105k	5.4k	QA/NLI	acc.	Wikipedia
RTE	2.5k	3k	NLI	acc.	news, Wikipedia
WNLI	634	**146**	coreference/NLI	acc.	fiction books

https://gluebenchmark.com/

(Wang et al., 2019)

GLUE Benchmark

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
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**

Comparable model size (~110M)

(Devlin et al. 2018)

Results

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
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**

(SQuAD 1.1)

(SWAG)

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

A girl is going across a set of monkey bars. She

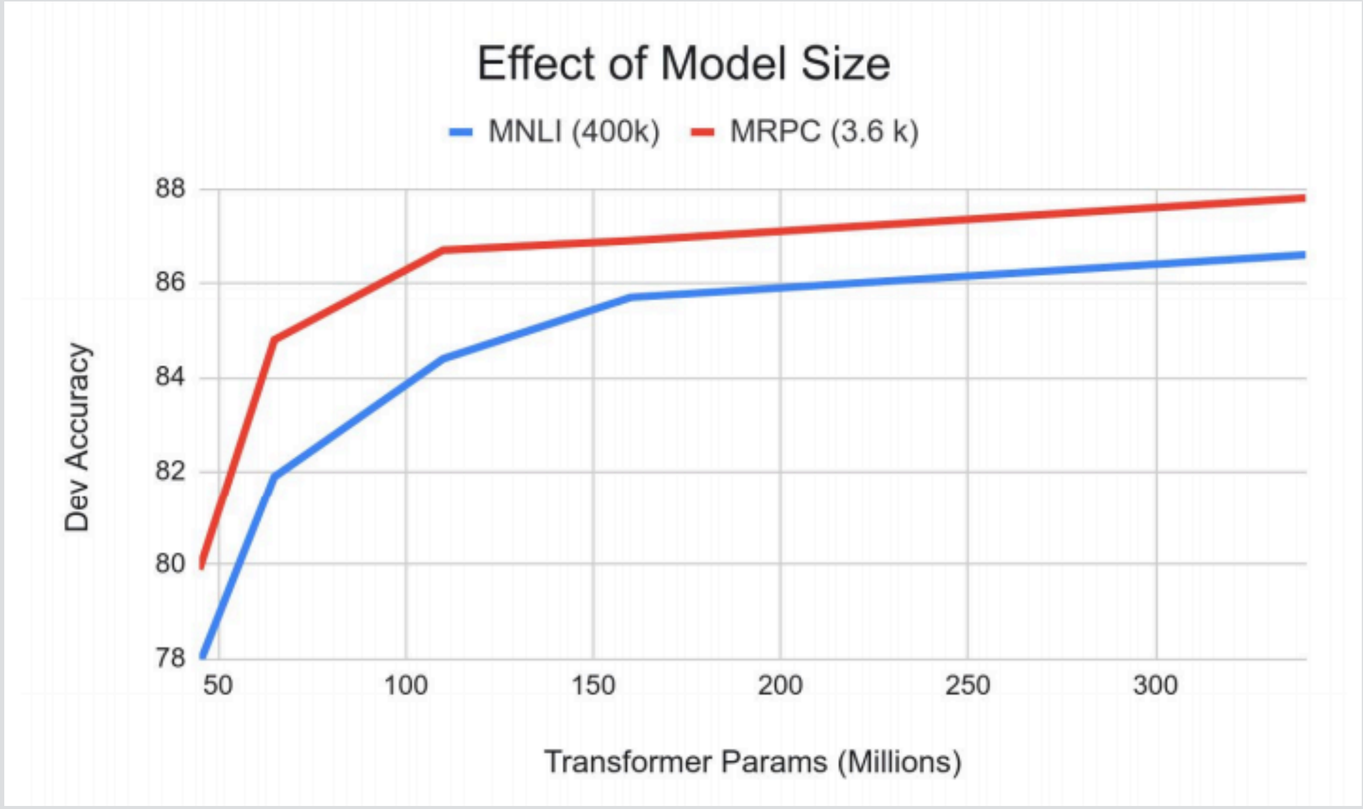
(i) jumps up across the monkey bars.

(ii) struggles onto the bars to grab her head.

(iii) gets to the end and stands on a wooden plank.

(iv) jumps up and does a back flip.

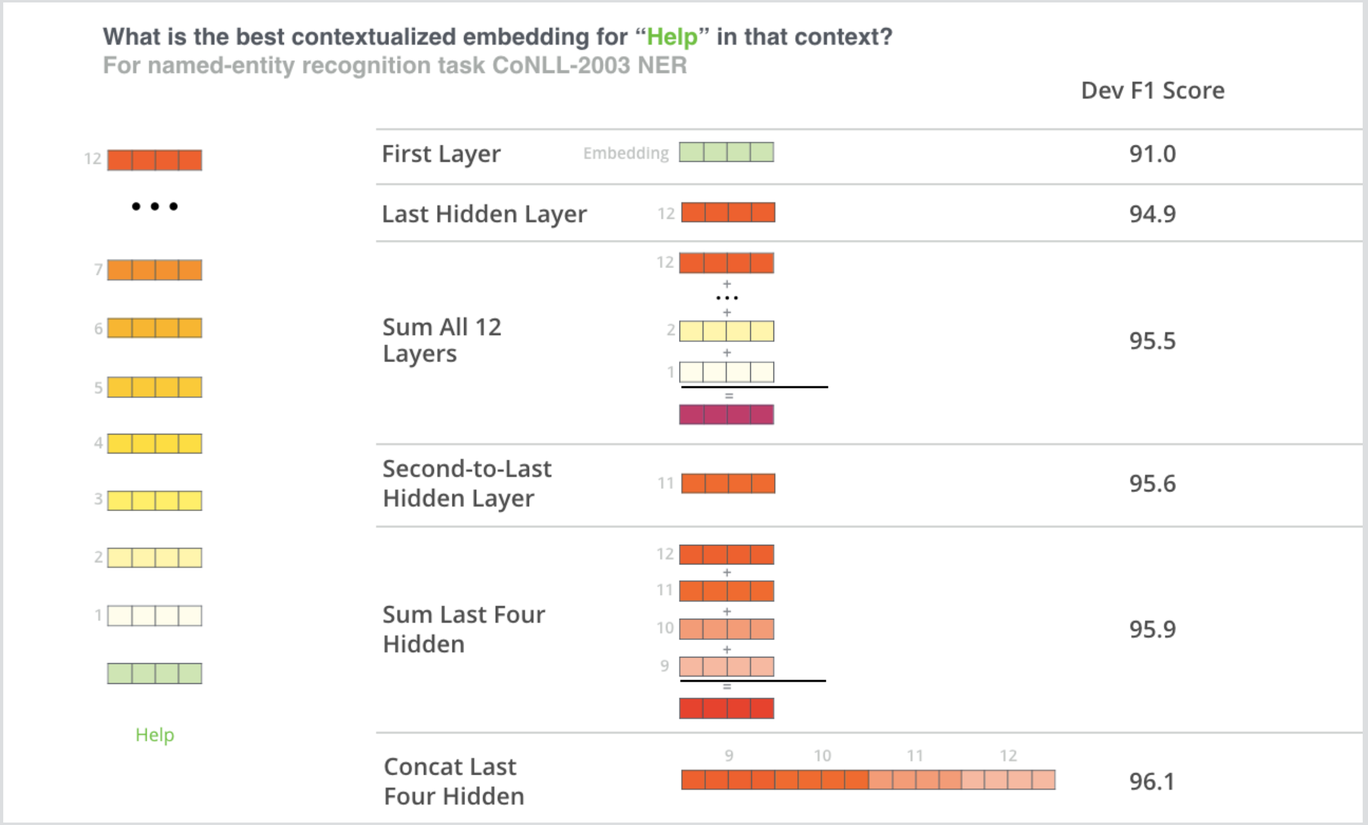
Effect of Model Size



Big models help (even with small labeled examples)

Using BERT for Representations

- Use the pre-trained model to obtain contextualized word representations for the input



[visualization from The Illustrated BERT: https://jalammar.github.io/illustrated-bert/]

Using BERT as “Features”

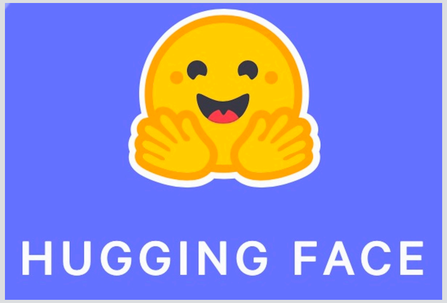
- In a similar way of using ELMo
  - Extracting the middle layer output as word representation, then feeding it into a two-layer BiLSTM (as task-specific model architecture)

System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	**93.1**
Fine-tuning approach		
$\text{BERT}_{\text{LARGE}}$	96.6	92.8
$\text{BERT}_{\text{BASE}}$	96.4	92.4
Feature-based approach ($\text{BERT}_{\text{BASE}}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

Table 7: CoNLL-2003 Named Entity Recognition results. Hyperparameters were selected using the Dev set. The reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters.

(Devlin et al. 2018)

Open Source!



- A few lines of Python code with Hugging Face

```
>>> from transformers import AutoTokenizer, AutoModel

>>> tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
>>> model = AutoModel.from_pretrained("bert-base-uncased")

>>> inputs = tokenizer("Hello world!", return_tensors="pt")
>>> outputs = model(**inputs)
```

- Many more pre-trained models (GPT2, Google T5, etc., also outside English) to check here: https://huggingface.co/models?library=pytorch

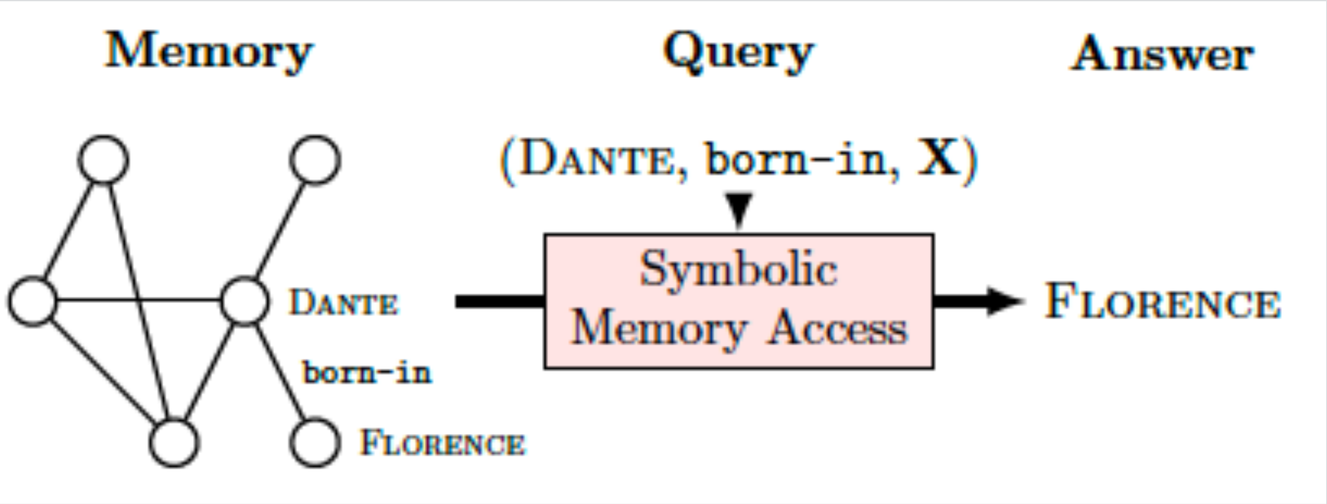
What's learned by BERT?
Example: commonsense knowledge, linguistic structure

Break

Language Models as Knowledge Bases

(Petroni et al., 2019)

- Knowledge bases

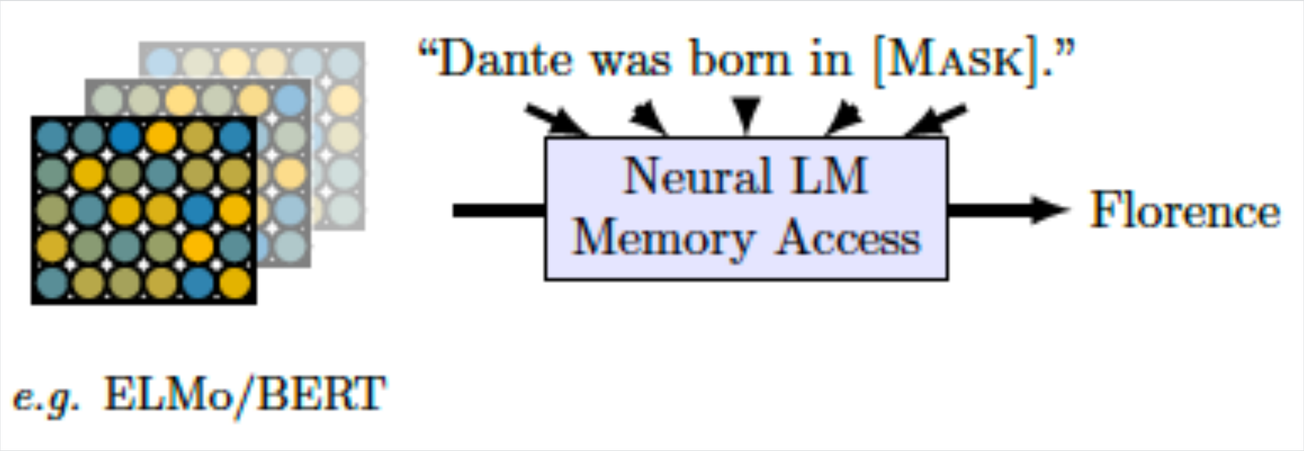


Schema engineering, human annotations for training, incompleteness, outdated information,…

Language Models as Knowledge Bases

(Petroni et al., 2019)

- *PLMs* as knowledge bases



Language Models as Knowledge Bases

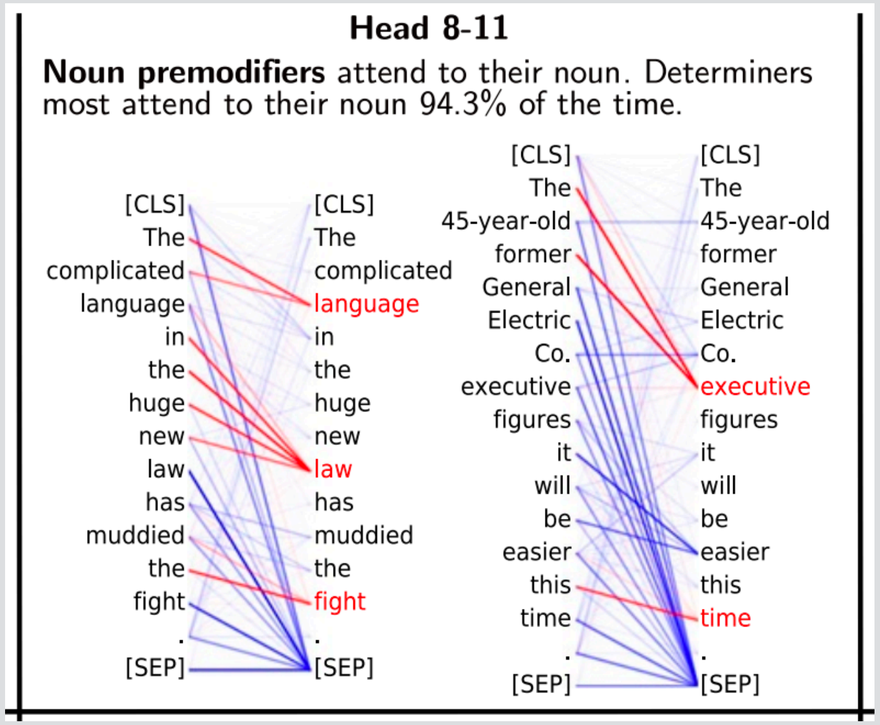
(Petroni et al., 2019)

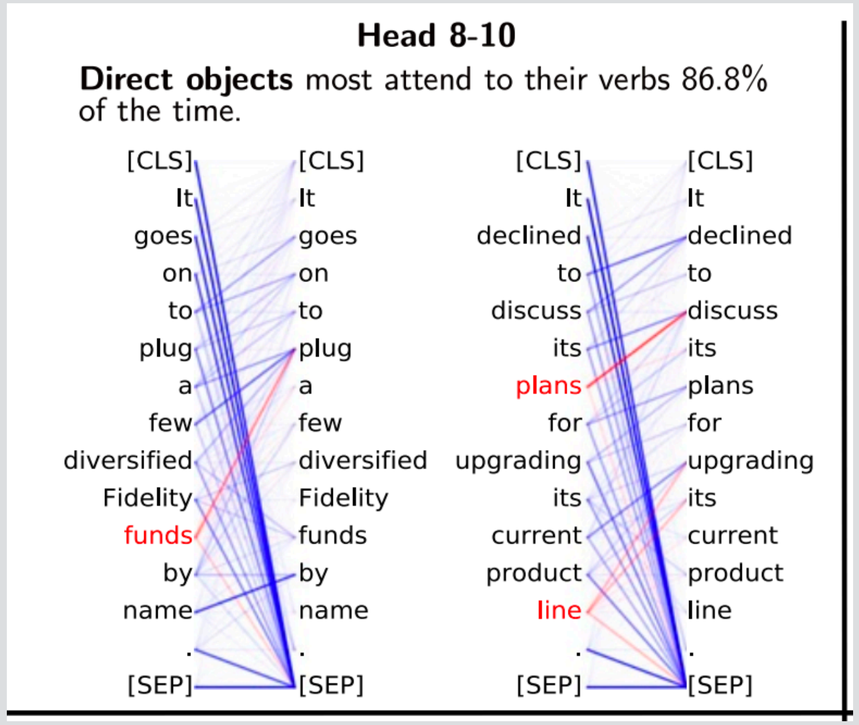
- *PLMs* as knowledge bases

	Relation	Query	Answer	Generation
T-Rex	P19	Francesco Bartolomeo Conti was born in ____.	Florence	Rome [-1.8], **Florence [-1.8]**, Naples [-1.9], Milan [-2.4], Bologna [-2.5]
	P20	Adolphe Adam died in ____.	Paris	**Paris [-0.5]**, London [-3.5], Vienna [-3.6], Berlin [-3.8], Brussels [-4.0]
	P279	English bulldog is a subclass of ____.	dog	dogs [-0.3], breeds [-2.2], **dog [-2.4]**, cattle [-4.3], sheep [-4.5]
	P37	The official language of Mauritius is ____.	English	**English [-0.6]**, French [-0.9], Arabic [-6.2], Tamil [-6.7], Malayalam [-7.0]
	P413	Patrick Oboya plays in ____ position.	midfielder	centre [-2.0], center [-2.2], **midfielder [-2.4]**, forward [-2.4], midfield [-2.7]
	P138	Hamburg Airport is named after ____.	Hamburg	Hess [-7.0], Hermann [-7.1], Schmidt [-7.1], **Hamburg [-7.5]**, Ludwig [-7.5]
	P364	The original language of Mon oncle Benjamin is ____.	French	**French [-0.2]**, Breton [-3.3], English [-3.8], Dutch [-4.2], German [-4.9]
	P54	Dani Alves plays with ____ .	Barcelona	Santos [-2.4], Porto [-2.5], Sporting [-3.1], Brazil [-3.3], Portugal [-3.7]
	P106	Paul Toungui is a ____ by profession .	politician	lawyer [-1.1], journalist [-2.4], teacher [-2.7], doctor [-3.0], physician [-3.7]
	P527	Sodium sulfide consists of ____.	sodium	water [-1.2], sulfur [-1.7], **sodium [-2.5]**, zinc [-2.8], salt [-2.9]
	P102	Gordon Scholes is a member of the ____ political party.	Labor	Labour [-1.3], Conservative [-1.6], Green [-2.4], Liberal [-2.9], **Labor [-2.9]**
	P530	Kenya maintains diplomatic relations with ____.	Uganda	India [-3.0], **Uganda [-3.2]**, Tanzania [-3.5], China [-3.6], Pakistan [-3.6]
	P176	iPod Touch is produced by ____.	Apple	**Apple [-1.6]**, Nokia [-1.7], Sony [-2.0], Samsung [-2.6], Intel [-3.1]
	P30	Bailey Peninsula is located in ____.	Antarctica	**Antarctica [-1.4]**, Bermuda [-2.2], Newfoundland [-2.5], Alaska [-2.7], Canada [-3.1]
	P178	JDK is developed by ____.	Oracle	IBM [-2.0], Intel [-2.3], Microsoft [-2.5], HP [-3.4], Nokia [-3.5]
	P1412	Carl III used to communicate in ____.	Swedish	German [-1.6], Latin [-1.9], French [-2.4], English [-3.0], Spanish [-3.0]
	P17	Sunshine Coast, British Columbia is located in ____.	Canada	**Canada [-1.2]**, Alberta [-2.8], Yukon [-2.9], Labrador [-3.4], Victoria [-3.4]
	P39	Pope Clement VII has the position of ____ .	pope	cardinal [-2.4], Pope [-2.5], **pope [-2.6]**, President [-3.1], Chancellor [-3.2]
	P264	Joe Cocker is represented by music label ____.	Capitol	EMI [-2.6], BMG [-2.6], Universal [-2.8], **Capitol [-3.2]**, Columbia [-3.3]
	P276	London Jazz Festival is located in ____.	London	**London [-0.3]**, Greenwich [-3.2], Chelsea [-4.0], Camden [-4.6], Stratford [-4.8]
	P127	Border TV is owned by ____.	ITV	Sky [-3.1], **ITV [-3.3]**, Global [-3.4], Frontier [-4.1], Disney [-4.3]
	P103	The native language of Mammootty is ____.	Malayalam	**Malayalam [-0.2]**, Tamil [-2.1], Telugu [-4.8], English [-5.2], Hindi [-5.6]
	P495	The Sharon Cuneta Show was created in ____.	Philippines	Manila [-3.2], **Philippines [-3.6]**, February [-3.7], December [-3.8], Argentina [-4.0]

Linguistic Features in BERT

- BERT learns linguistic structure, although it is not explicitly trained to
- Syntactic word dependencies through layers of *multi-head self-attention*

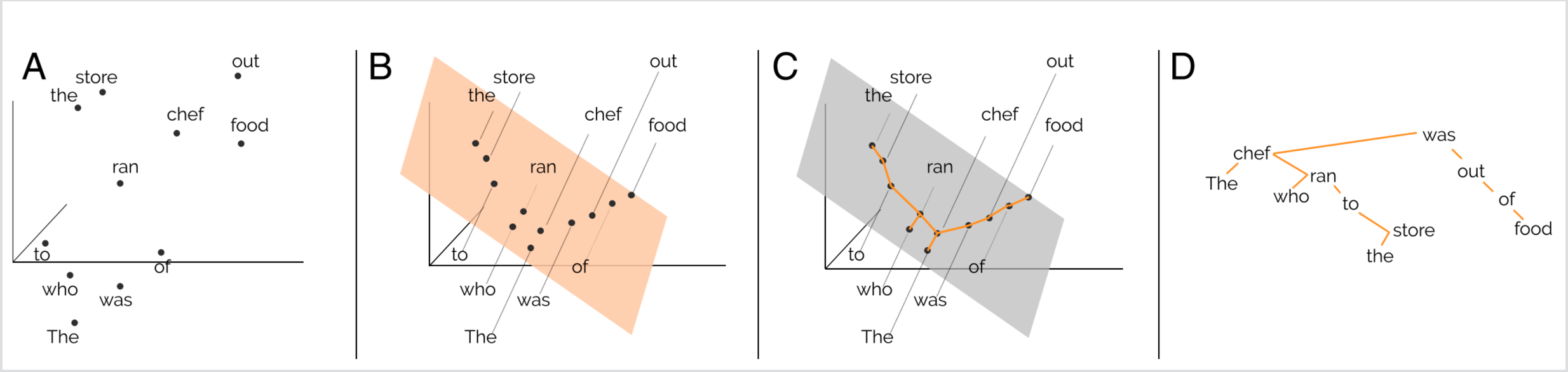




Emergent linguistic structure in artificial neural networks trained by self-supervision, Manning et al., PNAS 2019

Linguistic Features in BERT

- BERT learns linguistic structure, although it is not explicitly trained for it
- Can even recover the syntactic tree structure from BERT (through linear projections)



Emergent linguistic structure in artificial neural networks trained by self-supervision, Manning et al., PNAS 2019

What can BERT NOT do?

- BERT cannot **generate** text (at least not in an obvious way)
  - Not an autoregressive model, can do weird things like stick a [MASK] at the end of a string, fill in the mask, and repeat
- Masked language models are intended to be used primarily for “analysis” tasks

Subsequent Improvements to BERT

- Dynamic masking: standard BERT uses the same MASK scheme for every epoch, RoBERTa recomputes them

- Whole word masking: don’t mask out parts of words

… _John _visited _Mada gas car yesterday …

(Liu et al., 2019)

RoBERTa

- “Robustly optimized BERT” incorporating some of these tricks
- 160GB of data instead of 16 GB
- New training + more data = better performance

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

Facebook
(Liu et al., 2019)

SpanBERT

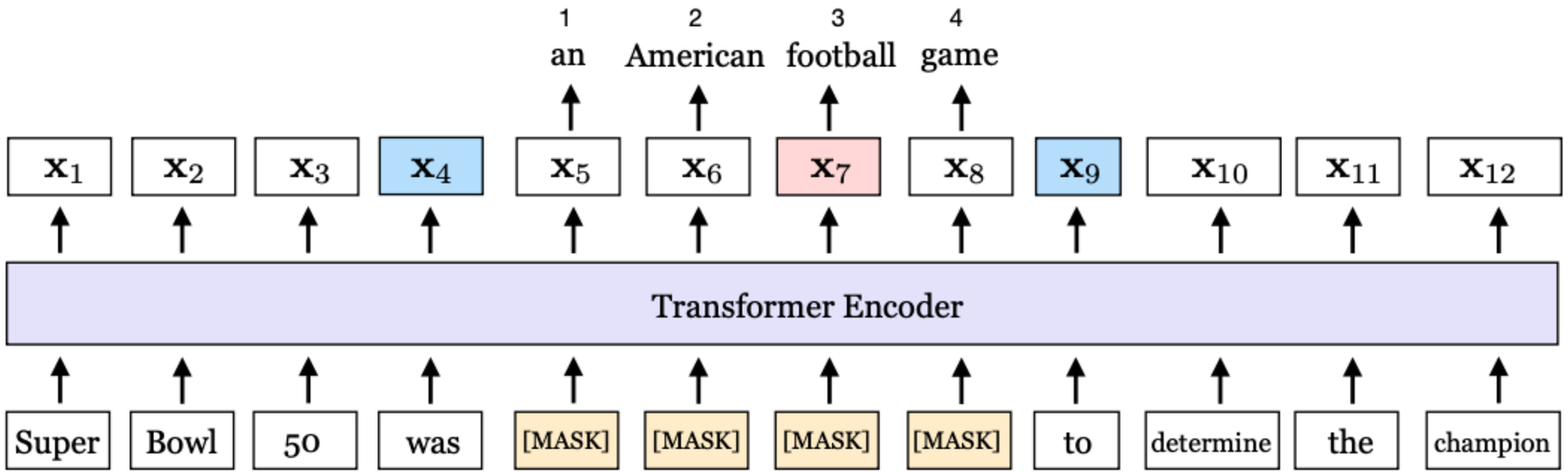
(Joshi et al., 2019)

	SQuAD 1.1 EM	SQuAD 1.1 F1	SQuAD 2.0 EM	SQuAD 2.0 F1
Human Perf.	82.3	91.2	86.8	89.4
Google BERT	84.3	91.3	80.0	83.3
Our BERT	86.5	92.6	82.8	85.9
Our BERT-1seq	87.5	93.3	83.8	86.6
SpanBERT	**88.8**	**94.6**	**85.7**	**88.7**

- Masking spans rather than tokens

$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$

$$= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)$$



BERT/MLMs

- A lot of ways to train!
- Key factors:
  - Big enough model
  - Big enough data
  - Well-designed “self-supervised” objective (something like language modeling). Needs to be a hard enough problem!

Outline

- Contextual Representations
- Pre-trained Language Models (PLMs)
  - ELMo, GPT-2, BERT
- Transfer Learning in NLP

Types of Learning

- **Multi-task learning** is a general term for training on multiple tasks
- **Transfer learning** is a type of multi-task learning where we only really care about one of the tasks
- **Domain adaptation** is a type of transfer learning, where the output is the same, but we want to handle different topics or genres, etc.

Plethora of Tasks in NLP

- In NLP, there are a plethora of tasks, each requiring different varieties of data
  - **Only text:** e.g. language modeling
  - **Naturally occurring data:** e.g. machine translation
  - **Hand-labeled data:** e.g. most analysis tasks
- And each in many languages, many domains!

Rule of Thumb 1: Multitask to Increase Data

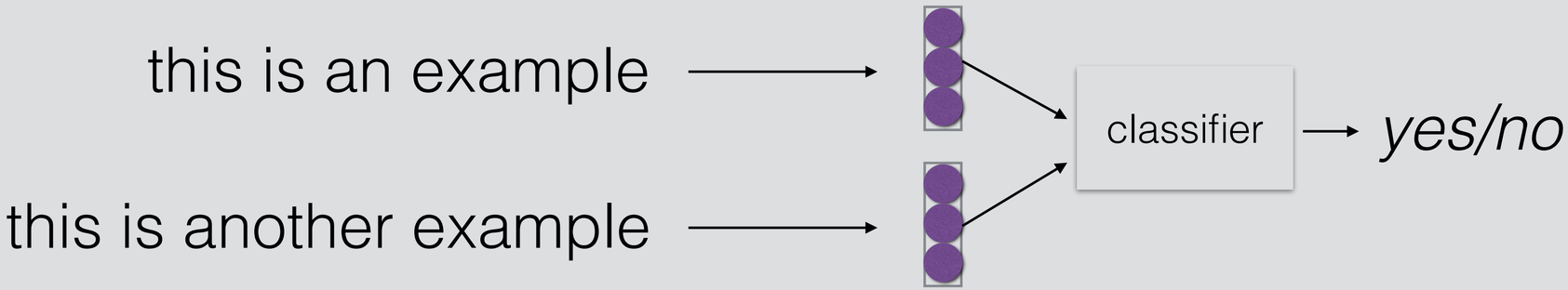
- Perform multi-tasking when one of your two tasks has many fewer data
- **General domain → specific domain**
  (e.g. web text → medical text)
- **High-resourced language → low-resourced language**
  (e.g. English → Telugu)
- **Plain text → labeled text**
  (e.g. LM -> parser)

Rule of Thumb 2:

- Perform multi-tasking when your **tasks are related**
- e.g. predicting eye gaze and summarization (Klerke et al. 2016)

Consider Learning Sentence Representations

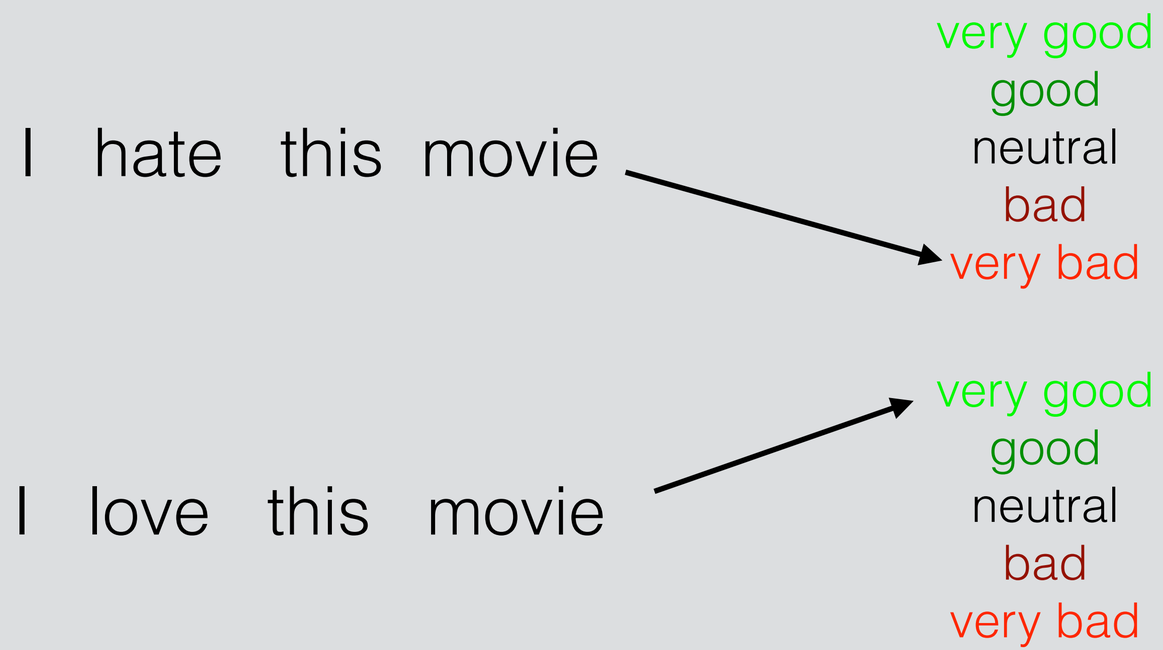
- Calculate vector representation
- Feed vector representation into classifier



How do we get such a representation?

Sentence Classification

- Classify sentences according to various traits
- Topic, sentiment, subjectivity/objectivity, etc.



Paraphrase Identification
(Dolan and Brockett 2005)

- Identify whether A and B mean the same thing

Charles O. Prince, 53, was named as Mr. Weill's successor.

Mr. Weill's longtime confidant, Charles O. Prince, 53,
was named as his successor.

Note: *exactly* the same thing is too restrictive, so use a loose sense of similarity

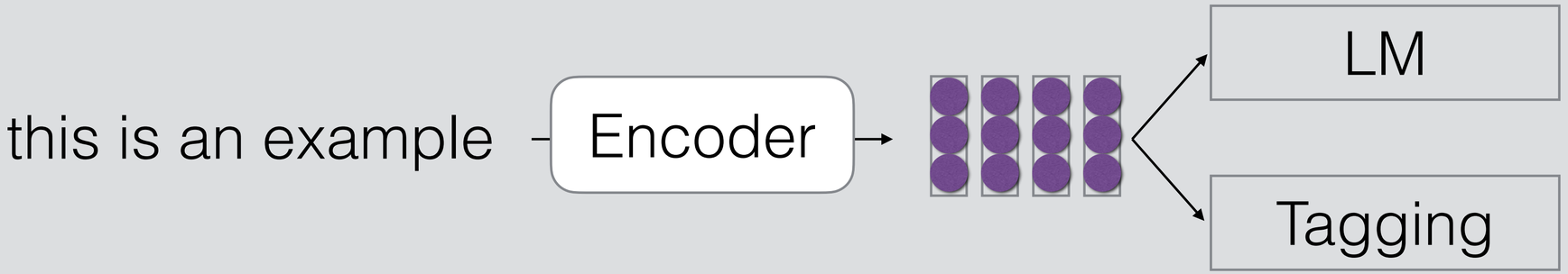
Textual Entailment

(Dagan et al. 2006, Marelli et al. 2014)

- **Entailment:** if A is true, then B is true (c.f. paraphrase, where opposite is also true)
  - The woman bought a sandwich for lunch
    → The woman bought lunch
- **Contradiction:** if A is true, then B is not true
  - The woman bought a sandwich for lunch
    → The woman did not buy a sandwich
- **Neutral:** cannot say either of the above
  - The woman bought a sandwich for lunch
    → The woman bought a sandwich for dinner

Standard Multi-task Learning

- Train representations to do well on multiple tasks at once

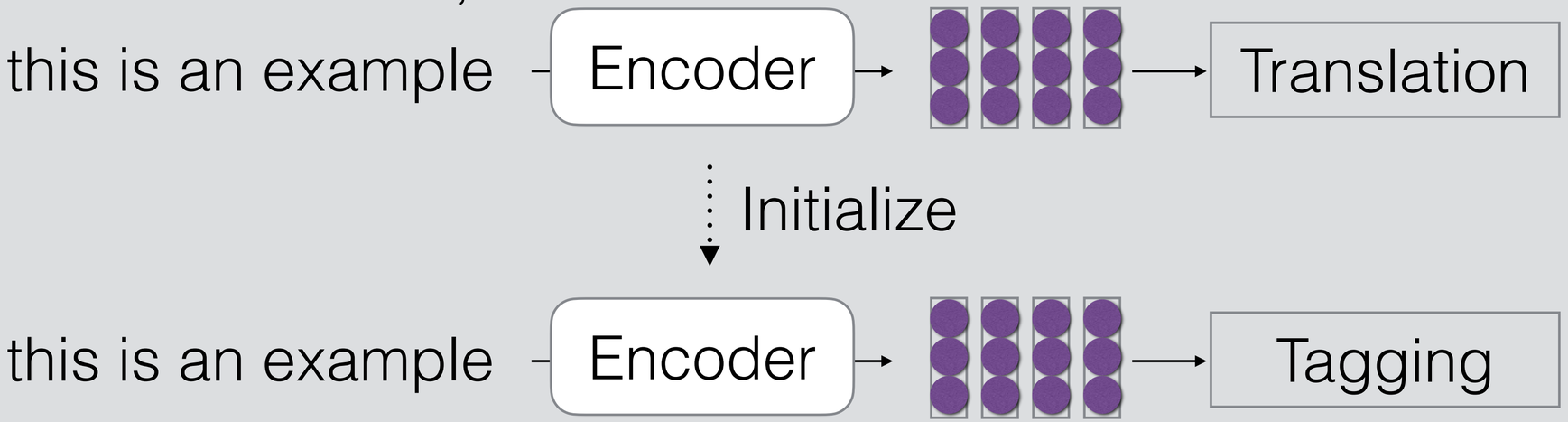


- In general, as simple as randomly choosing minibatch from one of multiple tasks

- Many many examples, starting with Collobert and Weston (2011)

Pre-training

- First train on one task, then train on another



- Widely used in word embeddings (Turian et al. 2010)

- Also pre-training sentence encoders or contextualized word representations (Dai et al. 2015, Melamud et al. 2016)

Thinking about Multi-tasking, and Pre-trained Representations

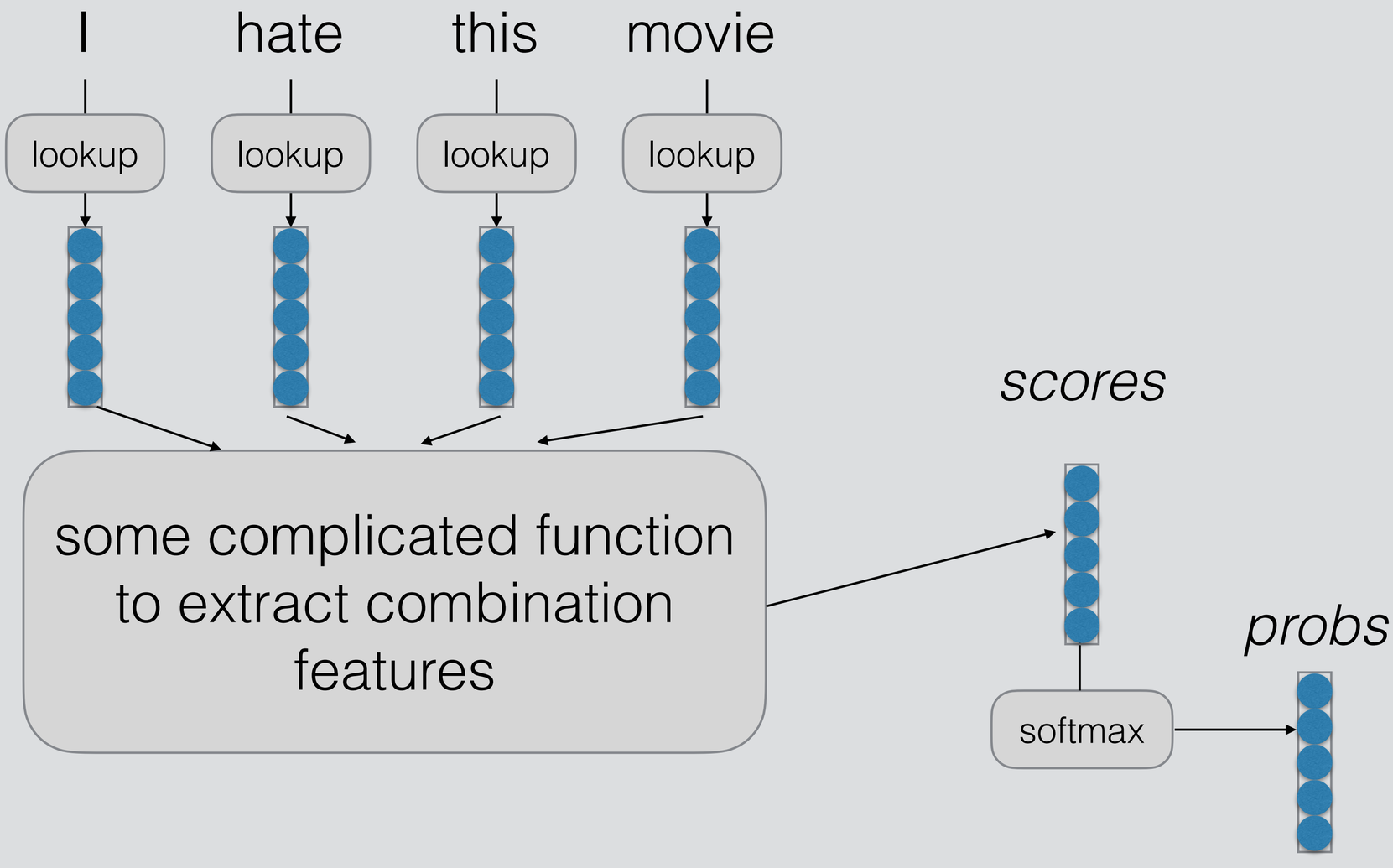
- Many methods have names like SkipThought, ParaNMT, CoVe, ELMo, BERT along with pre-trained models
- These often refer to a combination of
  - **Model:** The underlying neural network architecture
  - **Training Objective:** What objective is used to pre-train
  - **Data:** What data the authors chose to use to train the model
- Remember that these are often conflated (and don't need to be)!

End-to-end vs. Pre-training

- For any model, we can always use an end-to-end training objective
  - **Problem:** paucity of training data
  - **Problem:** weak feedback from end of sentence only for text classification, etc.
- Often better to pre-train sentence embeddings on other task, then use or fine-tune on target task

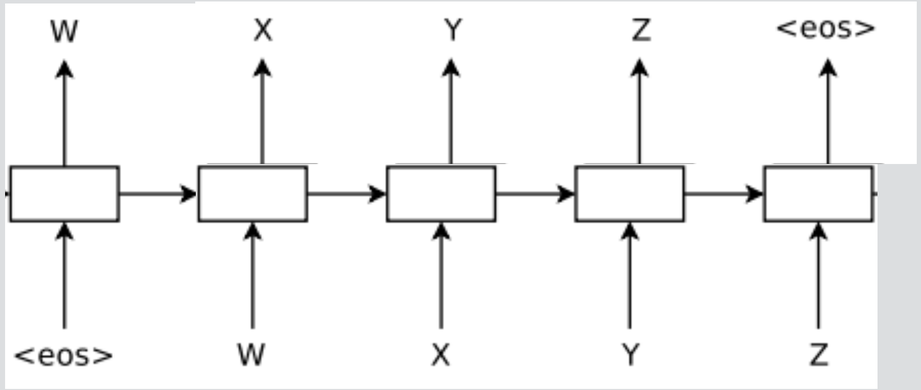
Training Sentence Representations

General Model Overview



Language Model Transfer
(Dai and Le 2015)

- **Model:** LSTM
- **Objective:** Language modeling objective
- **Data:** Classification data itself, or Amazon reviews

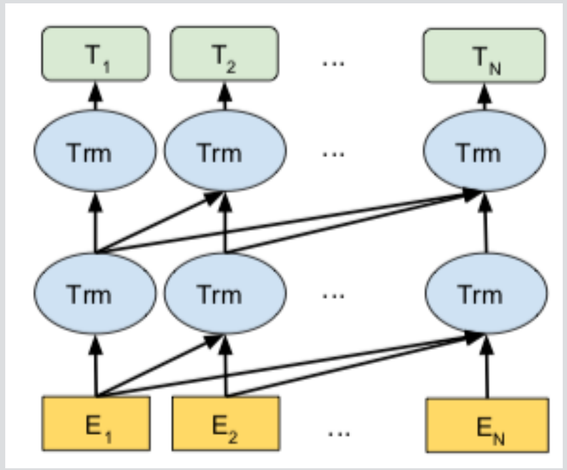


- **Downstream:** On text classification, initialize weights and continue training

Unidirectional Training + Transformer (OpenAI GPT)

(Radford et al. 2018)

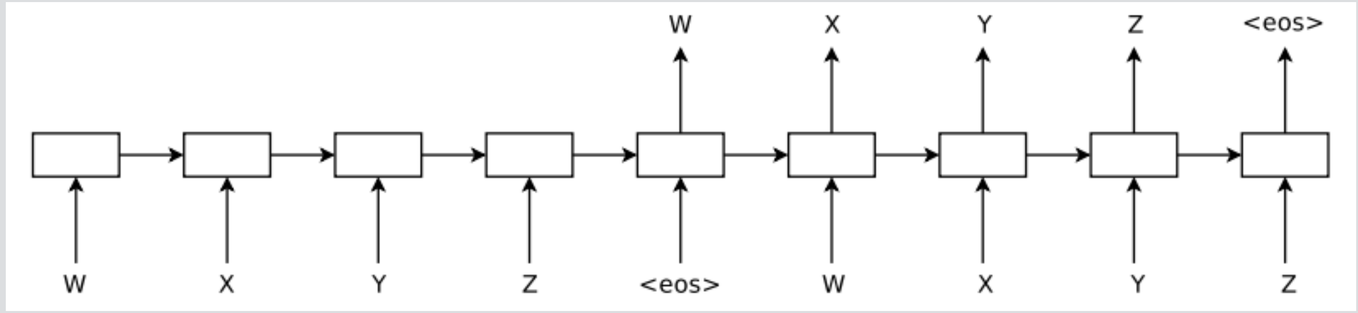
- **Model:** Masked self-attention
- **Objective:** Predict the next word left->right
- **Data:** BooksCorpus



- **Downstream:** Some task fine-tuning, other tasks additional multi-sentence training: Some task fine-tuning, other tasks additional multi-sentence training

Auto-encoder Transfer
(Dai and Le 2015)

- **Model:** LSTM
- **Objective:** From single sentence vector, re-construct the sentence
- **Data:** Classification data itself, or Amazon reviews

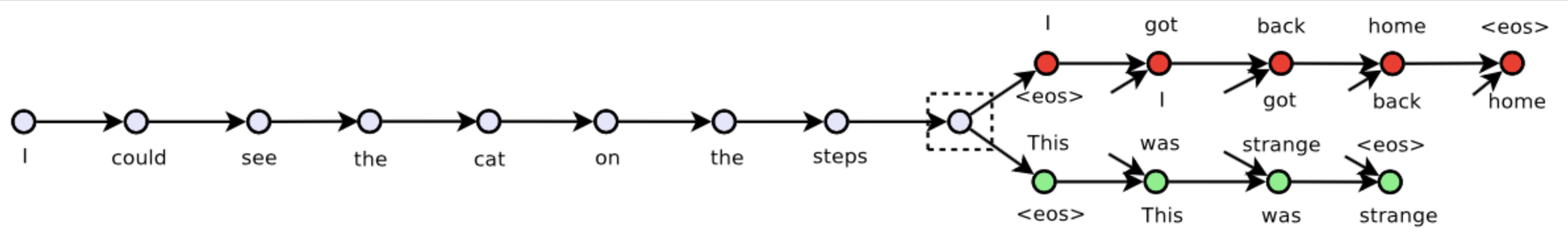


- **Downstream:** On text classification, initialize weights and continue training

Context Prediction Transfer (Skip-thought Vectors)

(Kiros et al. 2015)

- **Model:** LSTM
- **Objective:** Predict the surrounding sentences
- **Data:** Books, important because of context



- **Downstream Usage:** Train logistic regression on [|u-v|; u*v] (component-wise)

Paraphrase ID Transfer (Wieting et al. 2015)

- **Model:** Try many different ones
- **Objective:** Predict whether two phrases are paraphrases or not from
- **Data:** Paraphrase database (http://paraphrase.org), created from bilingual data
- **Downstream Usage:** Sentence similarity, classification, etc.
- **Result:** Interestingly, LSTMs work well on in-domain data, but word averaging generalizes better

Large Scale Paraphrase Data (ParaNMT-50MT)

(Wieting and Gimpel 2018)

- **Automatic construction of large paraphrase DB**
  - Get large parallel corpus (English-Czech)
  - Translate the Czech side using a SOTA NMT system
  - Get automated score and annotate a sample
- Corpus is **huge but includes noise**, 50M sentences (about 30M are high quality)
- Trained representations work quite well and generalize

Entailment Transfer (InferSent)

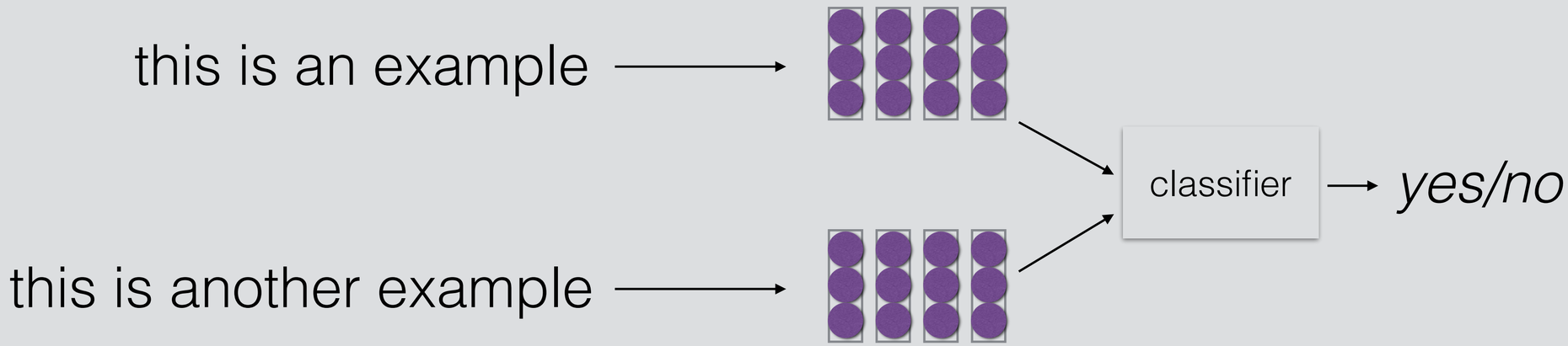
(Conneau et al. 2017)

- Previous objectives use no human labels, but what if:
- **Objective:** supervised training for a task such as entailment learn generalizable embeddings?
  - Task is more difficult and requires capturing nuance → yes?, or data is much smaller → no?
- **Model:** Bi-LSTM + max pooling
- **Data:** Stanford NLI, MultiNLI
- **Results:** Tends to be better than unsupervised objectives such as SkipThought

Contextualized Word Representations

Contextualized Word Representations

- Instead of one vector per sentence, one vector per word!

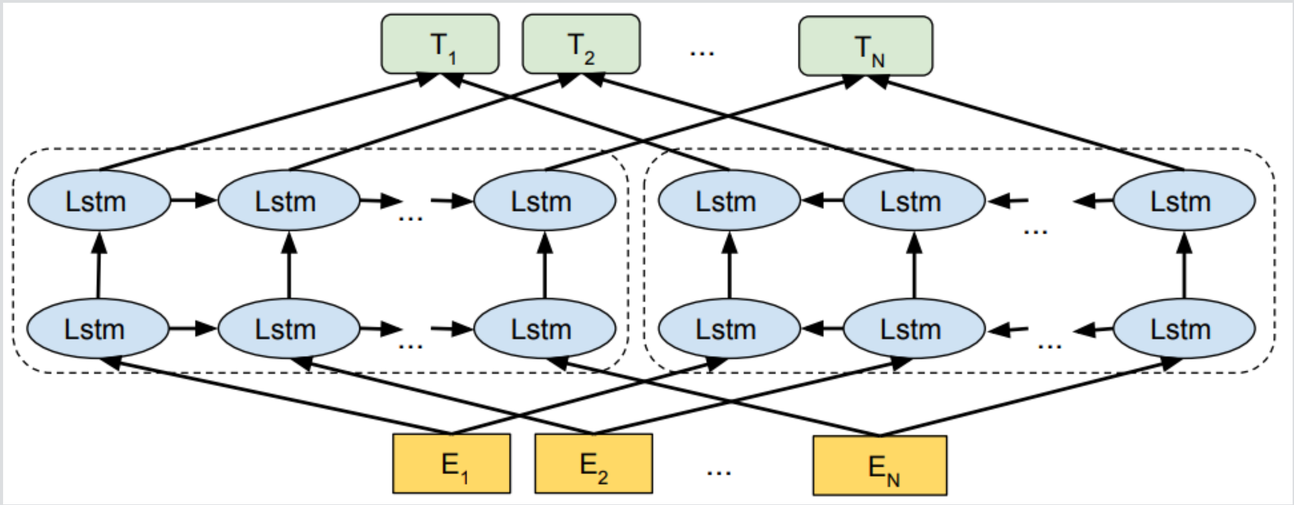


How to train this representation?

Bi-directional Language Modeling Objective (ELMo)

(Peters et al. 2018)

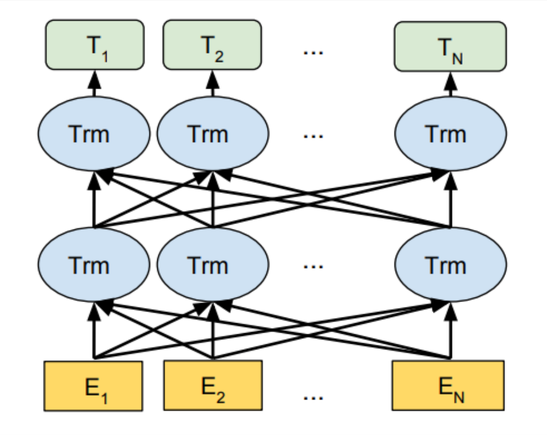
- **Model**: Multi-layer bi-directional LSTM
- **Objective**: Predict the next word left->right, next word right->left independently
- **Data**: 1B word benchmark LM dataset
- **Downstream**: Fine-tune the weights of the linear combination of layers on the downstream task

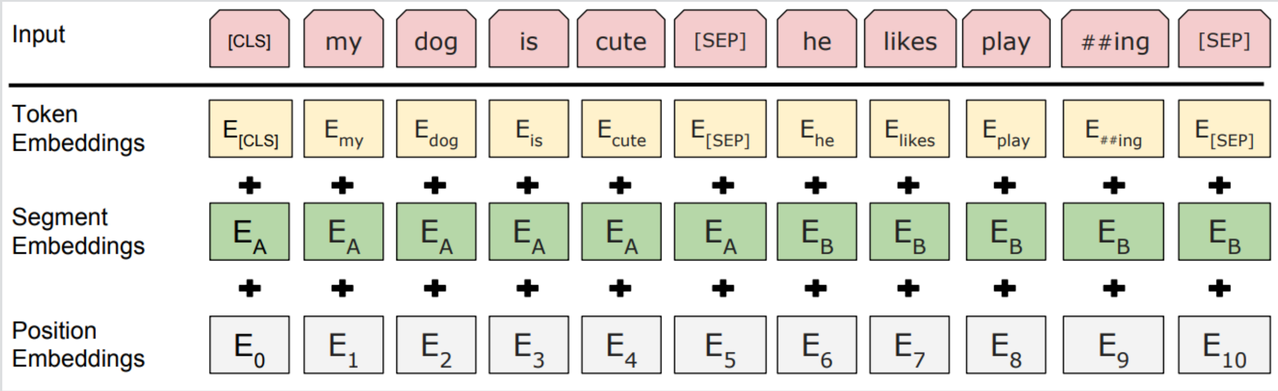


Masked Word Prediction (BERT)

(Devlin et al. 2018)

- **Model**: Multi-layer self-attention. Input sentence or pair, w/ [CLS] token, subword representation





- **Objective**: Masked word prediction + next-sentence prediction

- **Data**: BooksCorpus + English Wikipedia

Which Method is Better?

Which Model?

- Not very extensive comparison...
- Wieting et al. (2015) find that simple word averaging is more robust out-of-domain
- Devlin et al. (2018) compare unidirectional and bi-directional transformer, but no comparison to LSTM like ELMo (for performance reasons?)

Which Training Objective?

- Not very extensive comparison...
- Zhang and Bowman (2018) control for training data, and find that bi-directional LM seems better than MT encoder
- Devlin et al. (2018) find next-sentence prediction objective good compliment to LM objective

Which Data?

- Not very extensive comparison...
- Zhang and Bowman (2018) find that more data is probably better, but results are preliminary.
- Data with context is probably essential.

Some Recent Improvements

Various monolingual BERTs

- French: FlauBERT, CamemBERT
- BERTje, ALBERTO, BETO, KoBERT, FinBERT, Bangla-BERT, German, Chinese, Russian, Japanese, etc
- web-scale scraped corpora: https://oscar-corpus.com/

mBERT

- BERT trained on more than 100 languages
- Really good starting point, but also issues for low-resource languages, e.g. over-segmentation

RoBERTa

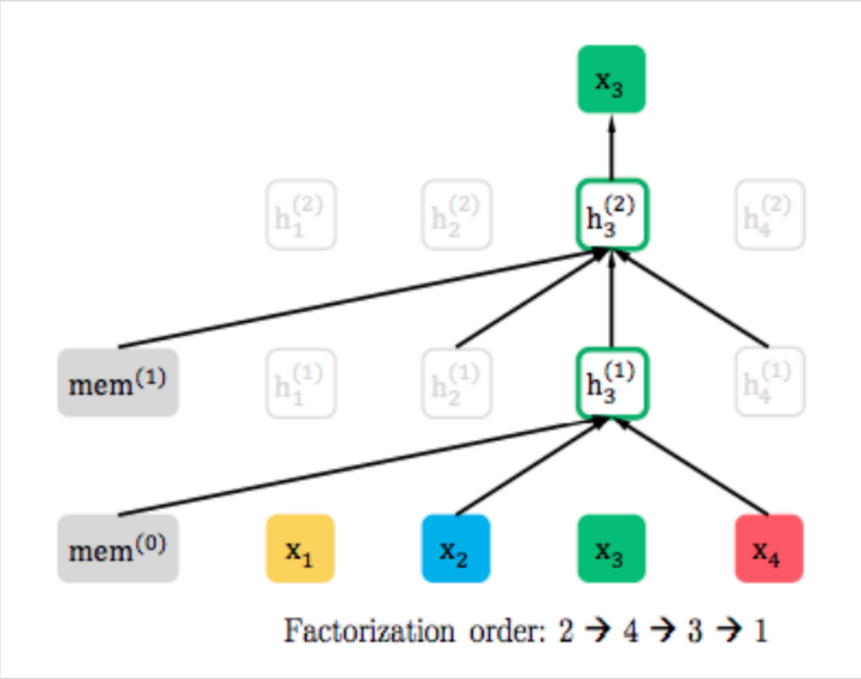
- Original BERT model was under-trained
- Better trained, more data, and more robust model

ALBERT

- BERT with fewer parameters, comparable performance
- Sentence-Order Prediction (instead of Next Sentence Prediction)

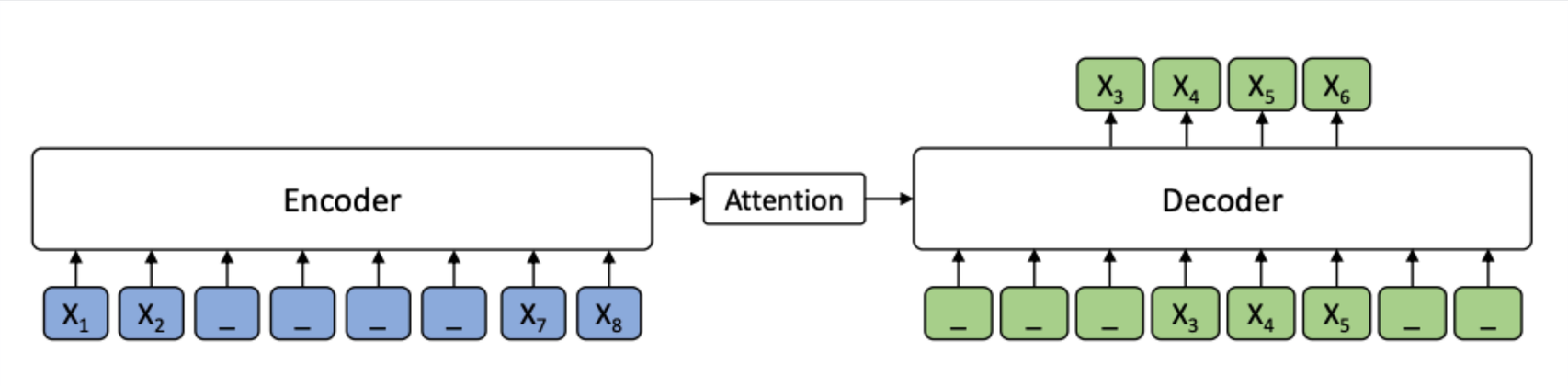
XLNet

- BERT problem: assumes [MASK] tokens are independent of each other
- Auto-Regressive + Permutation Language Modeling



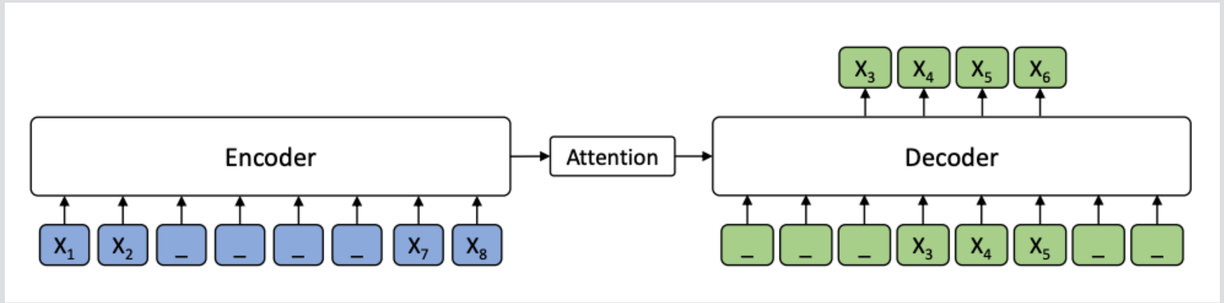
MASS: Masked Sequence to Sequence Pretraining

- BERT problem: Not good for generation
- Return to encoder-decoder model

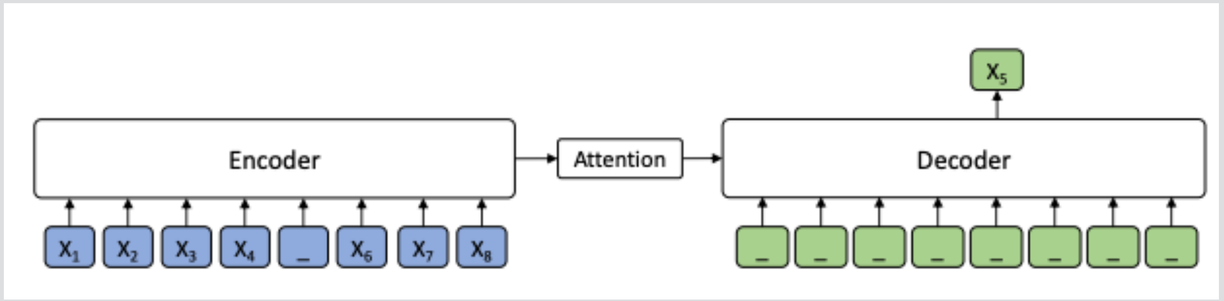


MASS (subcases)

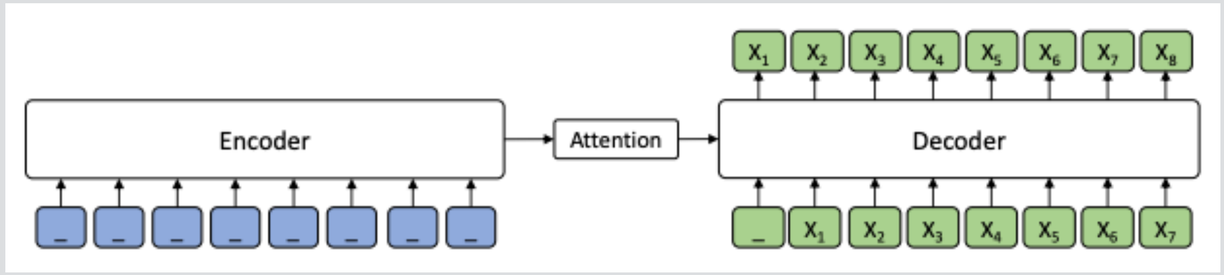
MASS



(BERT)

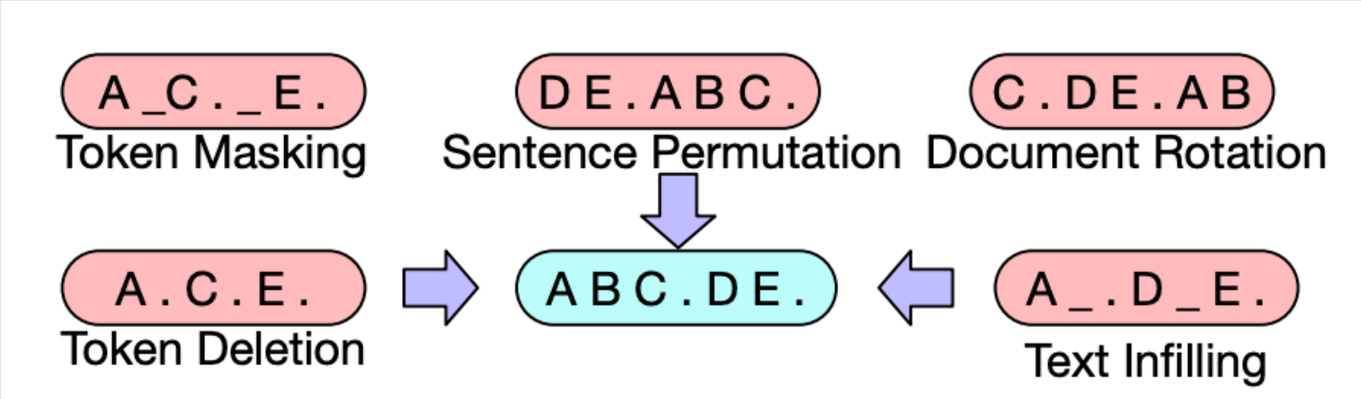


(LM)



BART

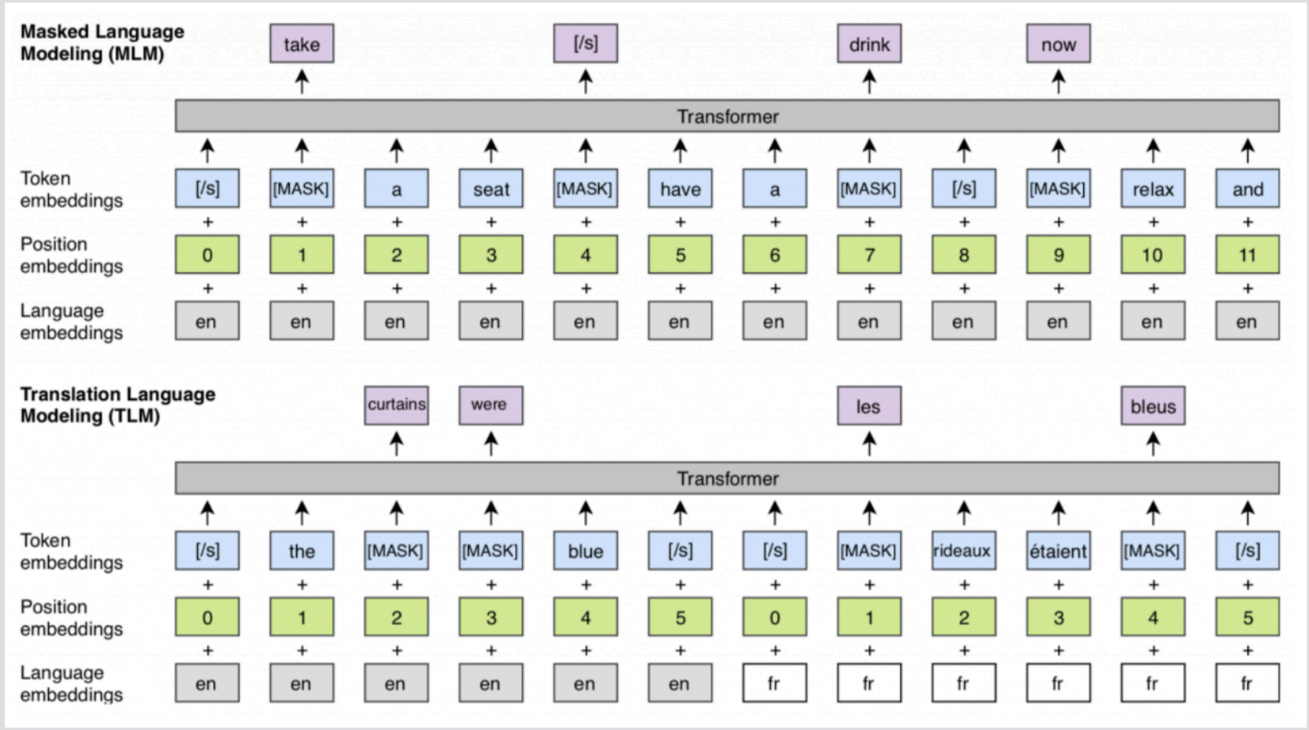
- BERT problem: requires output aligned with input. Return to encoder-decoder model
- Several input-noising functions



- There's also an mBART

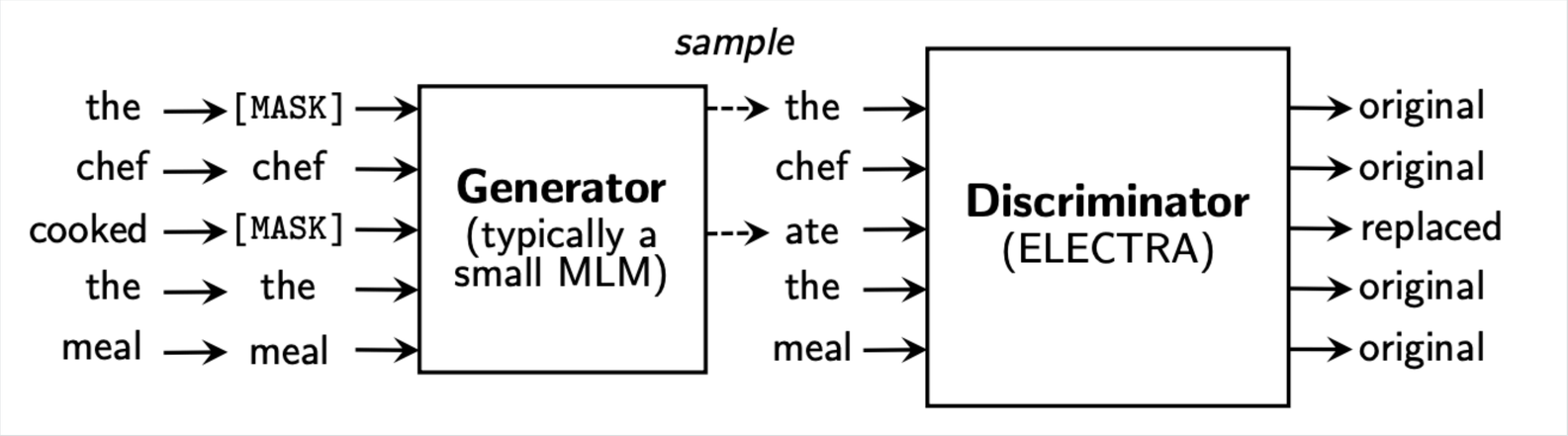
XLM and XLM-R

- BERT problem: each sample in a single language
- Combine MLM with Translation LM



ELECTRA
“Efficiently Learning an Encoder that Classifies Token Replacements Accurately”

- Inspired from GANs
- Objective: discriminate original/replaced tokens



- Trains faster (given BERT), smaller model, better performance

Outline

- Contextual Representations
- Pre-trained Language Models (PLMs)
  - ELMo, GPT-2, BERT
- Transfer Learning in NLP

Reminder

- HW 2 is out this week
- Group Formation and Project Interest Survey Form due **Sept 16**

OpenAI GPT-3

(Brown et al., 2020)

Language Models are Few-Shot Learners

Tom B. Brown* **Benjamin Mann*** **Nick Ryder*** **Melanie Subbiah***

Jared Kaplan† **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam** **Girish Sastry**

Amanda Askell **Sandhini Agarwal** **Ariel Herbert-Voss** **Gretchen Krueger** **Tom Henighan**

Rewon Child **Aditya Ramesh** **Daniel M. Ziegler** **Jeffrey Wu** **Clemens Winter**

Christopher Hesse **Mark Chen** **Eric Sigler** **Mateusz Litwin** **Scott Gray**

Benjamin Chess **Jack Clark** **Christopher Berner**

Sam McCandlish **Alec Radford** **Ilya Sutskever** **Dario Amodei**

OpenAI

OpenAI GPT-3

(Brown et al., 2020)

- GPT-2 but even larger: 1.5B -> 175B parameter models

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

- Trained on 570GB of Common Crawl
- 175B parameter model’s parameters alone take >400GB to store (4 bytes per param). Trained in parallel on a “high bandwidth cluster provided by Microsoft”

Traditional Fine-tuning

- The “normal way”

(Brown et al., 2020)

GPT-3: Few-shot Learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

Few-shot

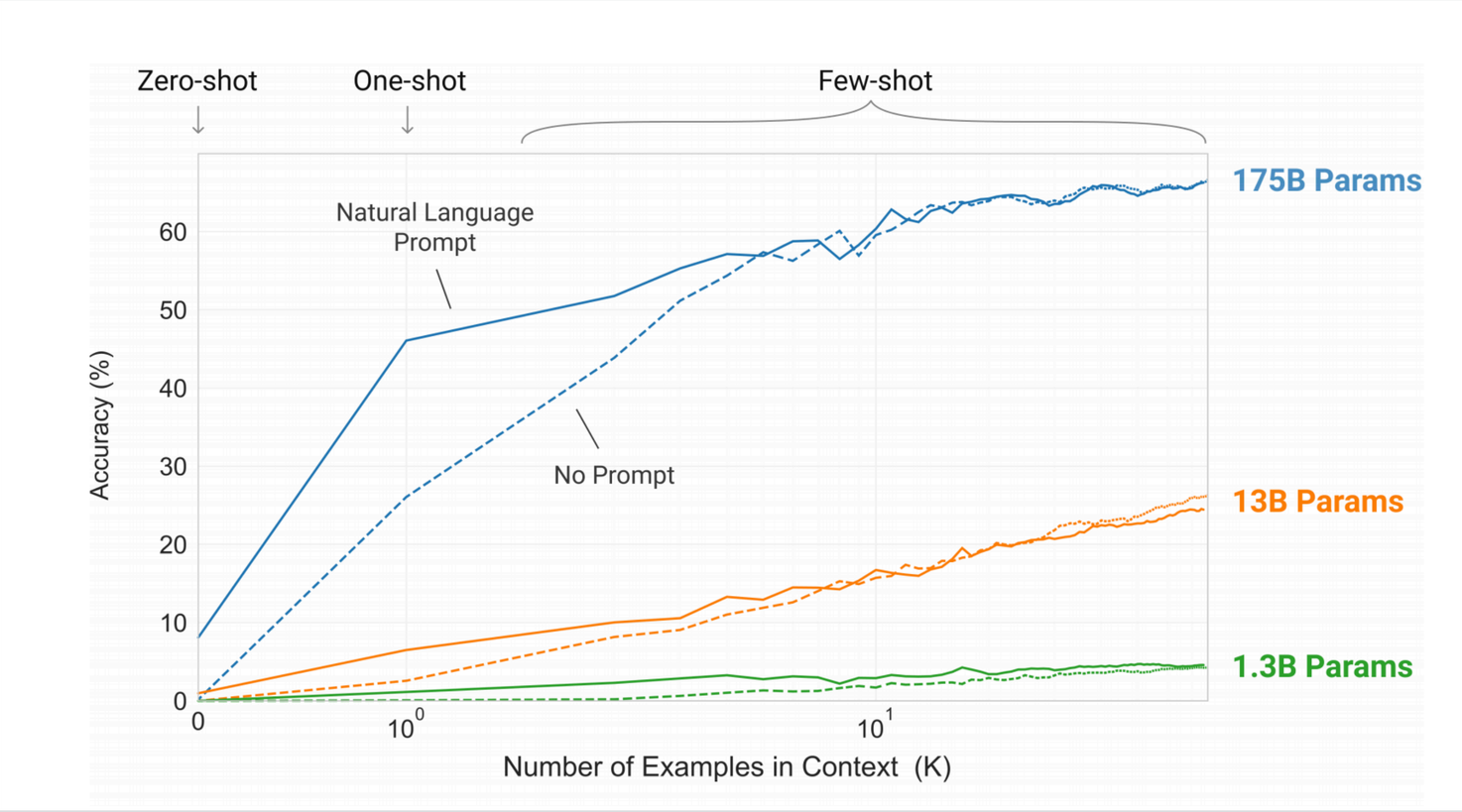
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée
4  plush girafe => girafe peluche
5  cheese =>                           ← prompt
```

(Brown et al., 2020)

GPT-3

- Key observation: few-shot learning only works with the very largest models!



(Brown et al., 2020)

Some Cool Demos

- https://github.com/elyase/awesome-gpt3
- Available at https://beta.openai.com/docs/introduction/overview: some free credits to use

GPT-3 on SuperGLUE

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

- Few shot: 32 examples
- Sometimes very impressive (COPA, ReCoRD, WSC), sometimes very bad (WiC)
- Results on other datasets are equally mixed — but still strong for a few-shot model!

(Brown et al., 2020)

SuperGLUE: https://super.gluebenchmark.com/tasks

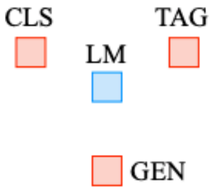
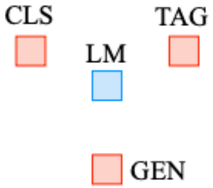
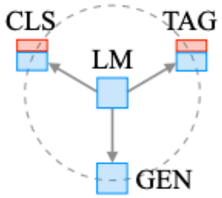
More Experimental Results

- Check more results in paper
  - Other NLP tasks: reading comprehension, machine translation (GPT-3's training corpus includes other languages), natural language inference, etc.
  - Algorithmic test: do well in 2/3 digit addition/subtraction but much worse in more complicated cases (4/5 digit operation, multiplication, sequential operations)
  - Answering SAT-style questions
  - News article generation: not easily distinguished from human writing

(Brown et al., 2020)

The Shift of Paradigm in NLP

(Liu et al., 2021)

Paradigm	Engineering	Task Relation
a. Fully Supervised Learning (Non-Neural Network)	Features (e.g. word identity, part-of-speech, sentence length)	
b. Fully Supervised Learning (Neural Network)	Architecture (e.g. convolutional, recurrent, self-attentional)	
c. Pre-train, Fine-tune	Objective (e.g. masked language modeling, next sentence prediction)	
d. Pre-train, Prompt, Predict	Prompt (e.g. cloze, prefix)