CS378: Natural Language Processing

Lecture 18: Contextualized Word Embeddings / Language Model For Everything

TEXAS
The University of Texas at Austin

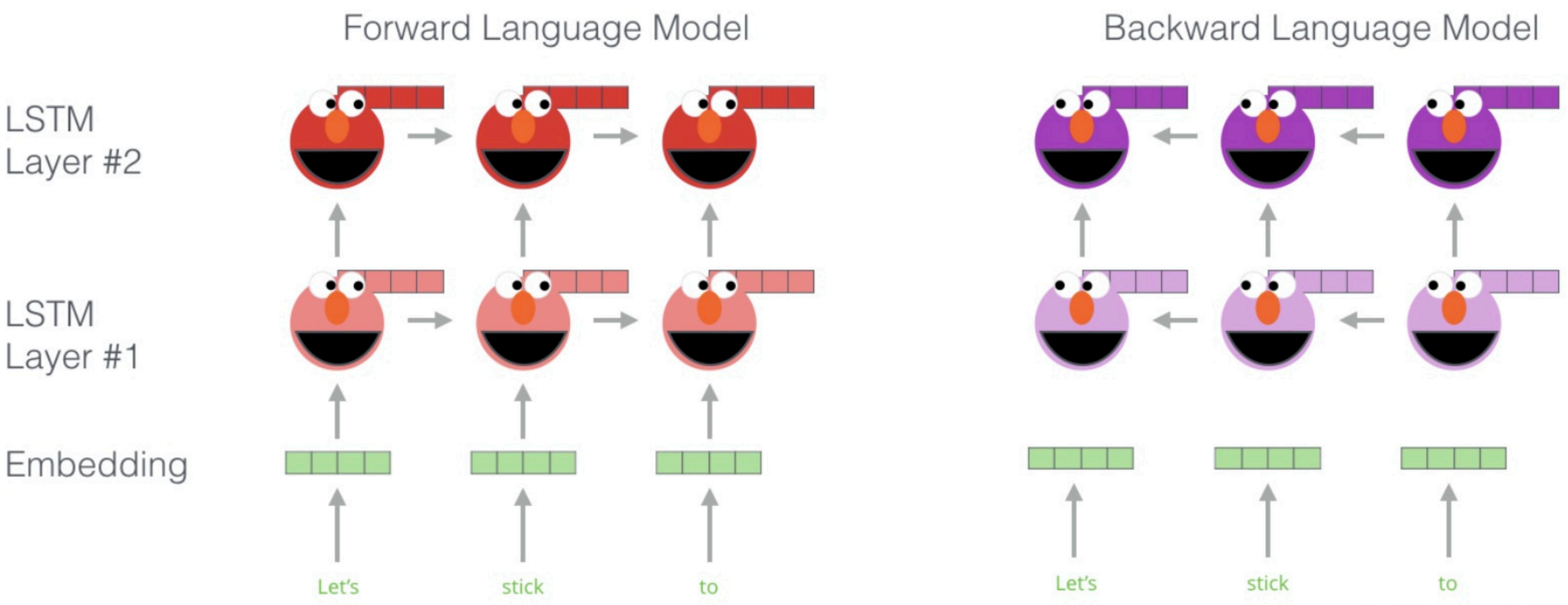
Eunsol Choi

Slides adapted from Greg Durrett, Graham Neubig, Yejin Choi

Today

- Contextualized Word Embeddings
  - Brief Recap on ELMo
  - BERT
  - Encoder-Decoder Model

Recap:Embeddings from Language Models



words in the sentence

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

input

softmax

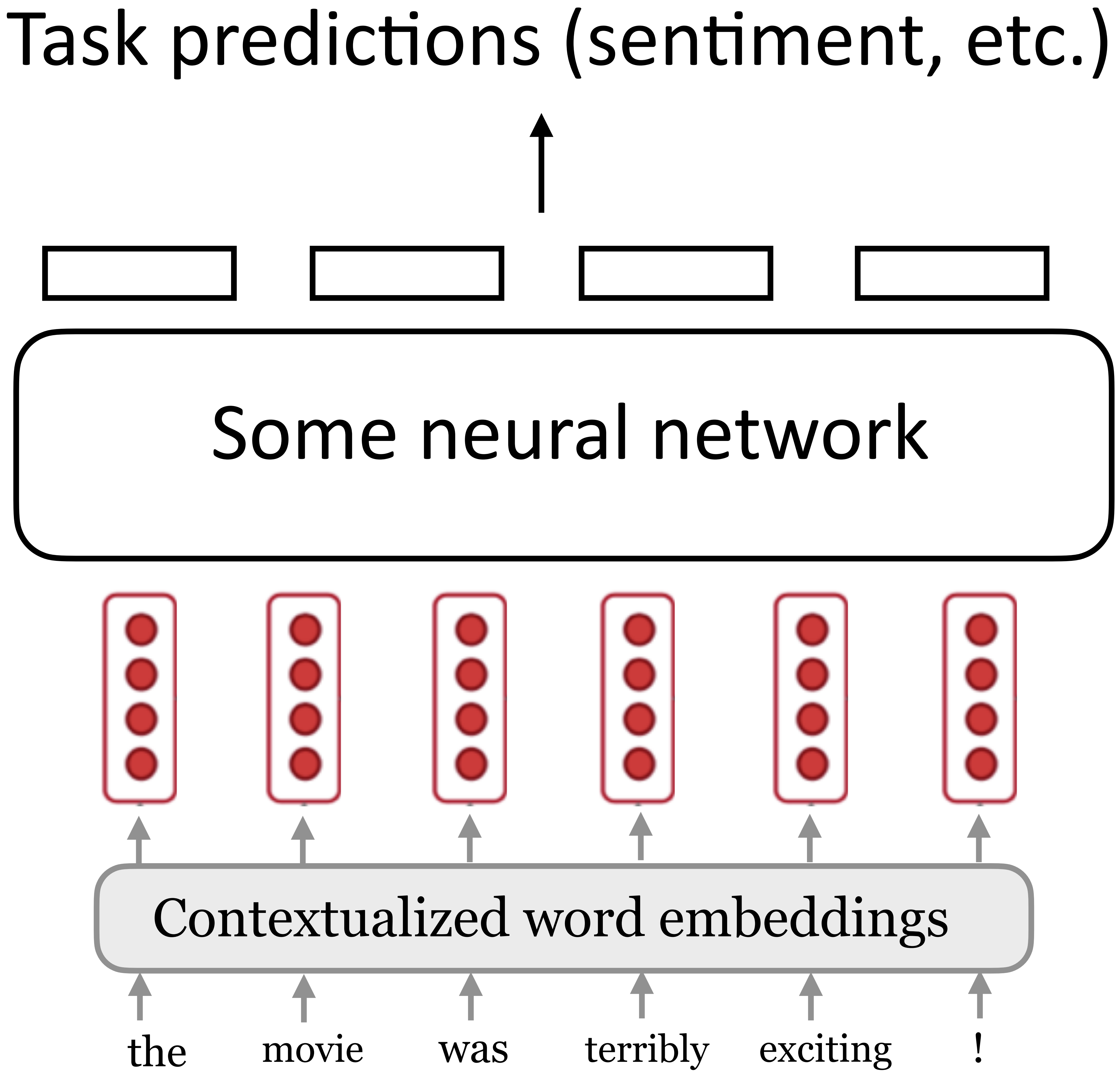
Peters et al. (2018)

Recap: Applying ELMo

- Take those embeddings and feed them into whatever architecture you want to use for your task
- *Frozen* embeddings: update the weights of your network but keep ELMo's parameters frozen
- *Plug ELMo into any (neural) NLP model: freeze all the LMs weights and change the input representation to:*

$$[\mathbf{x}_k; \mathbf{ELMo}_k^{task}]$$

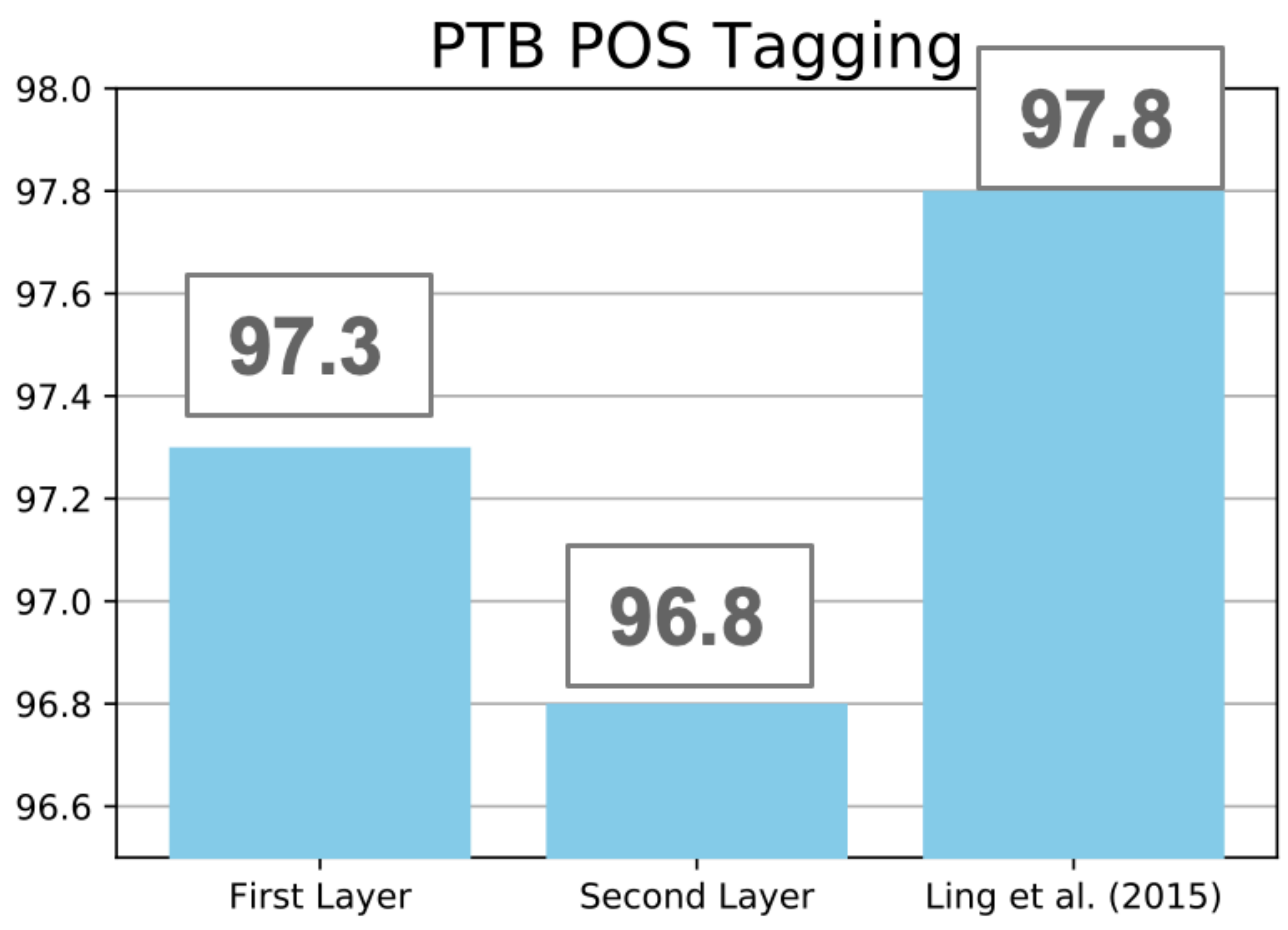
(could also insert into higher layers)



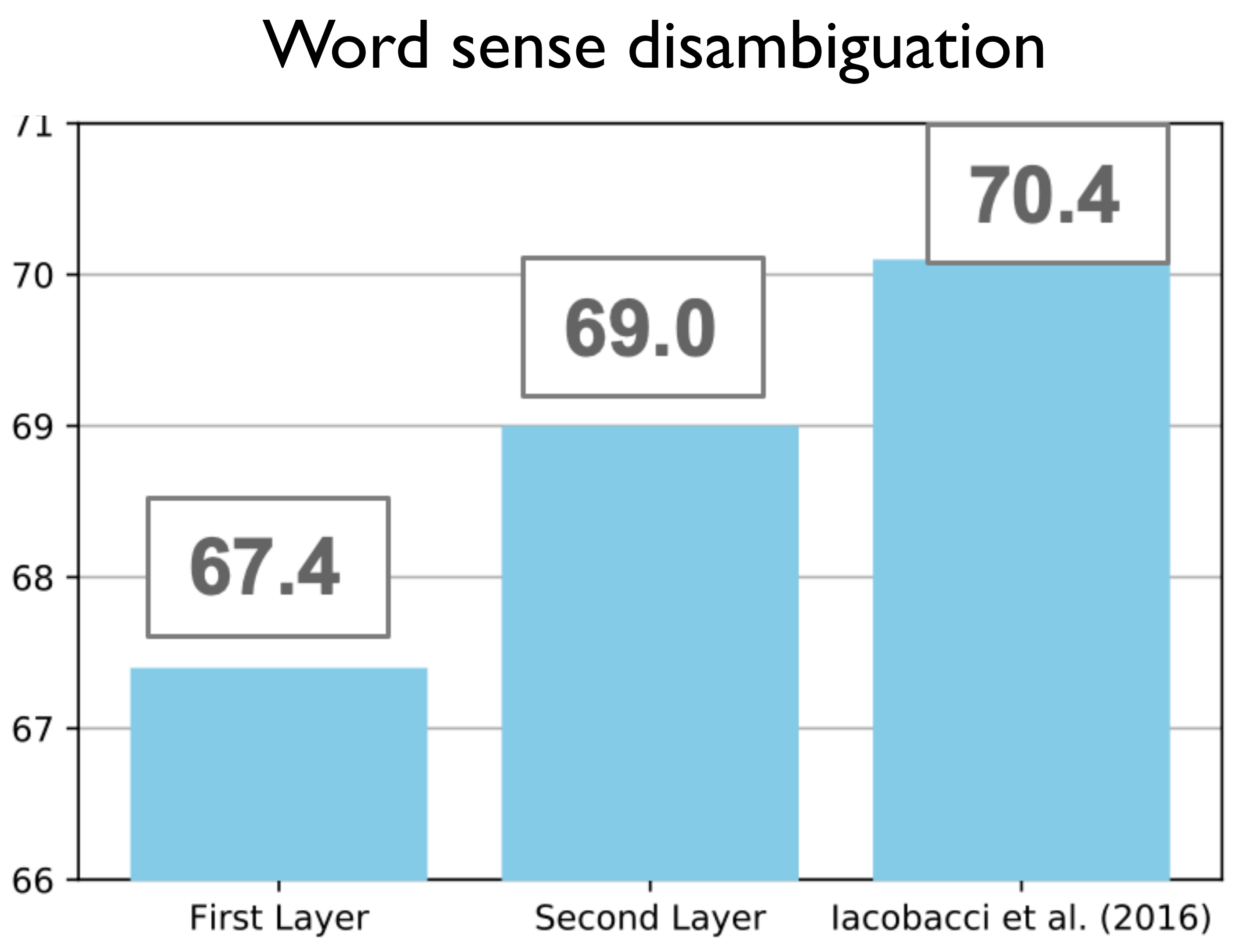
$f: (w_1, w_2, \ldots, w_n) \longrightarrow \mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$

Probing ELMo

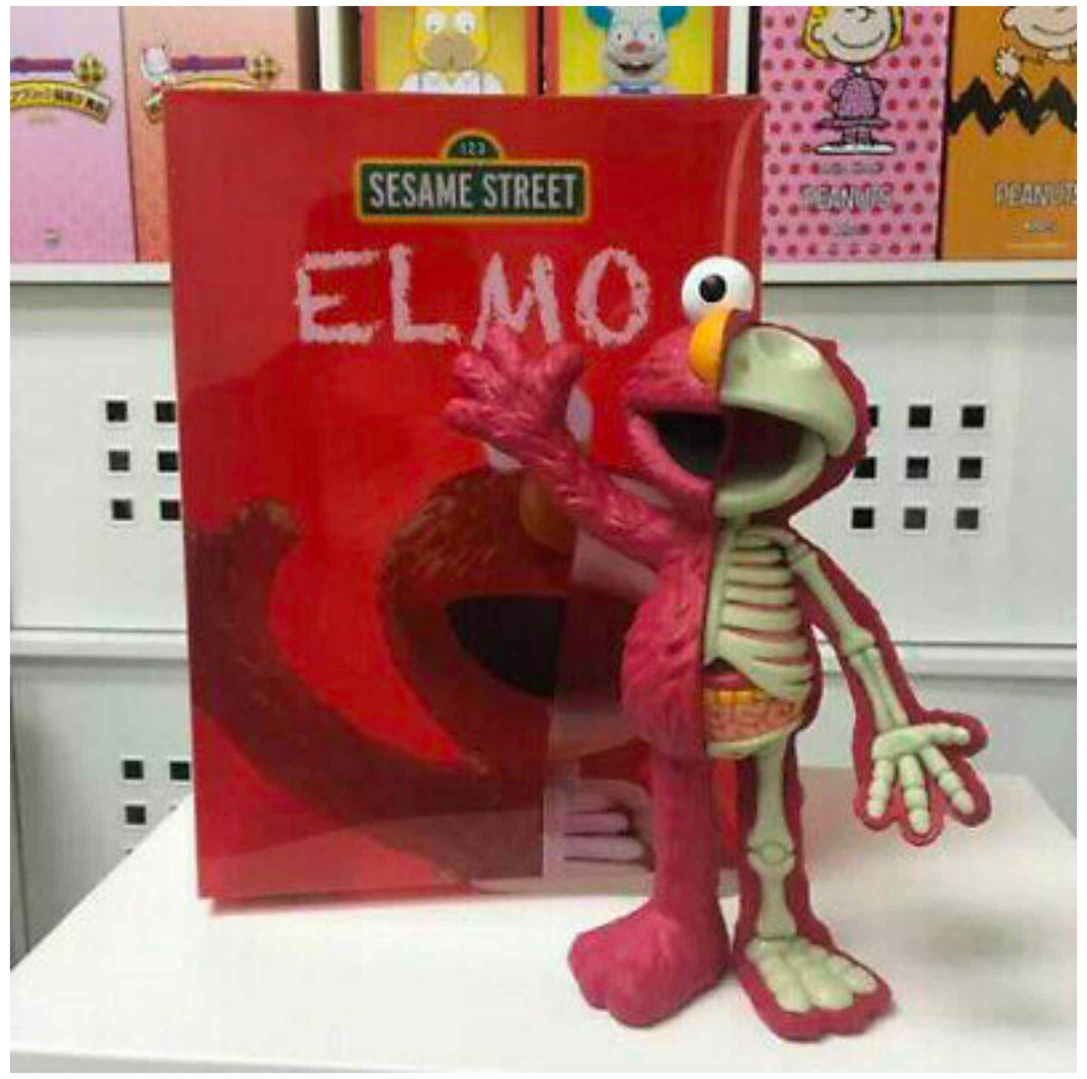
- From each layer of the ELMo model, attempt to predict something: POS tags, word senses, etc.
- Higher accuracy => ELMo is capturing that thing more strongly



First Layer > Second Layer



Second Layer > First Layer



Peters et al. (2018)

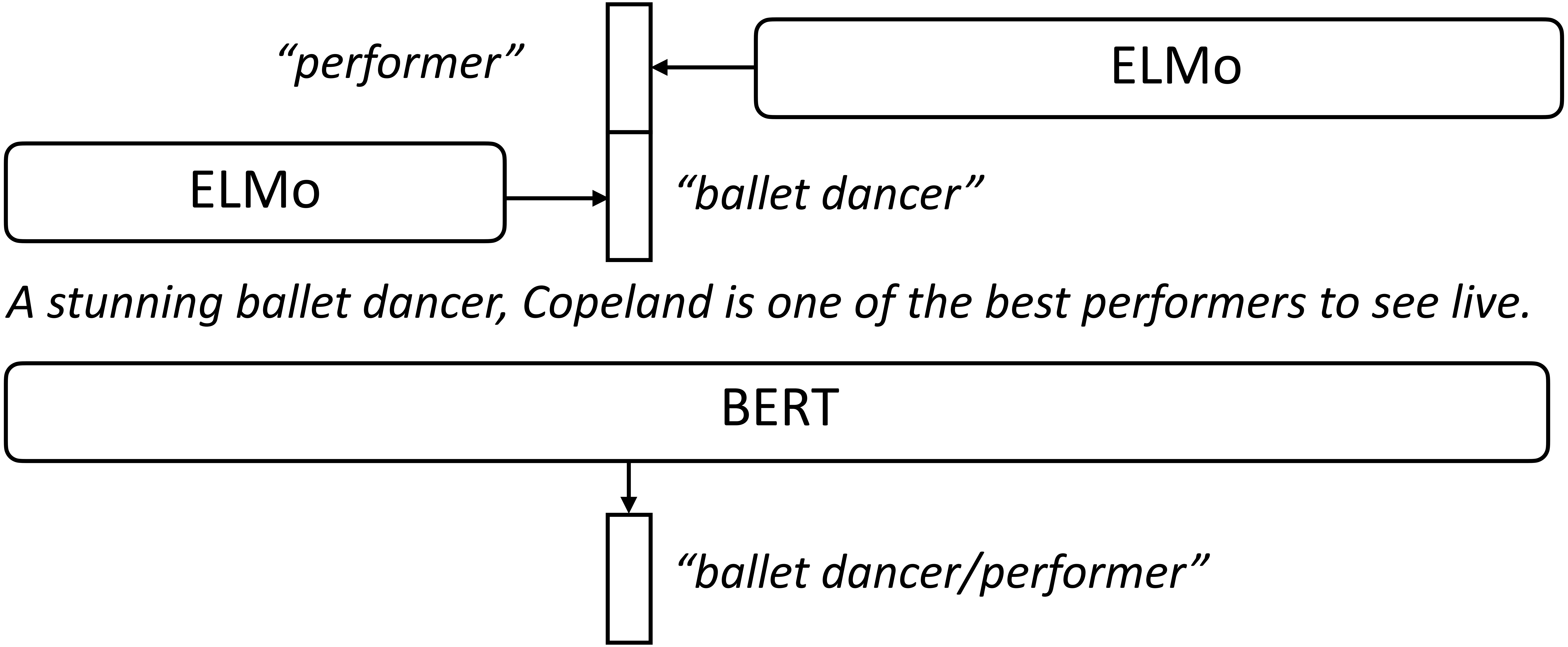
Timeline of Pretrained LM

BERT

- Four major changes compared to ELMo:
  - Transformers instead of LSTMs
  - Bidirectional model with "Masked LM" objective instead of standard LM
  - Fine-tune all parameters instead of freezing LM parameters
  - Operates over word pieces (sub word vocabulary)

BERT

- ELMo is a unidirectional model (as is GPT): we can concatenate two unidirectional models, but is this the right thing to do?

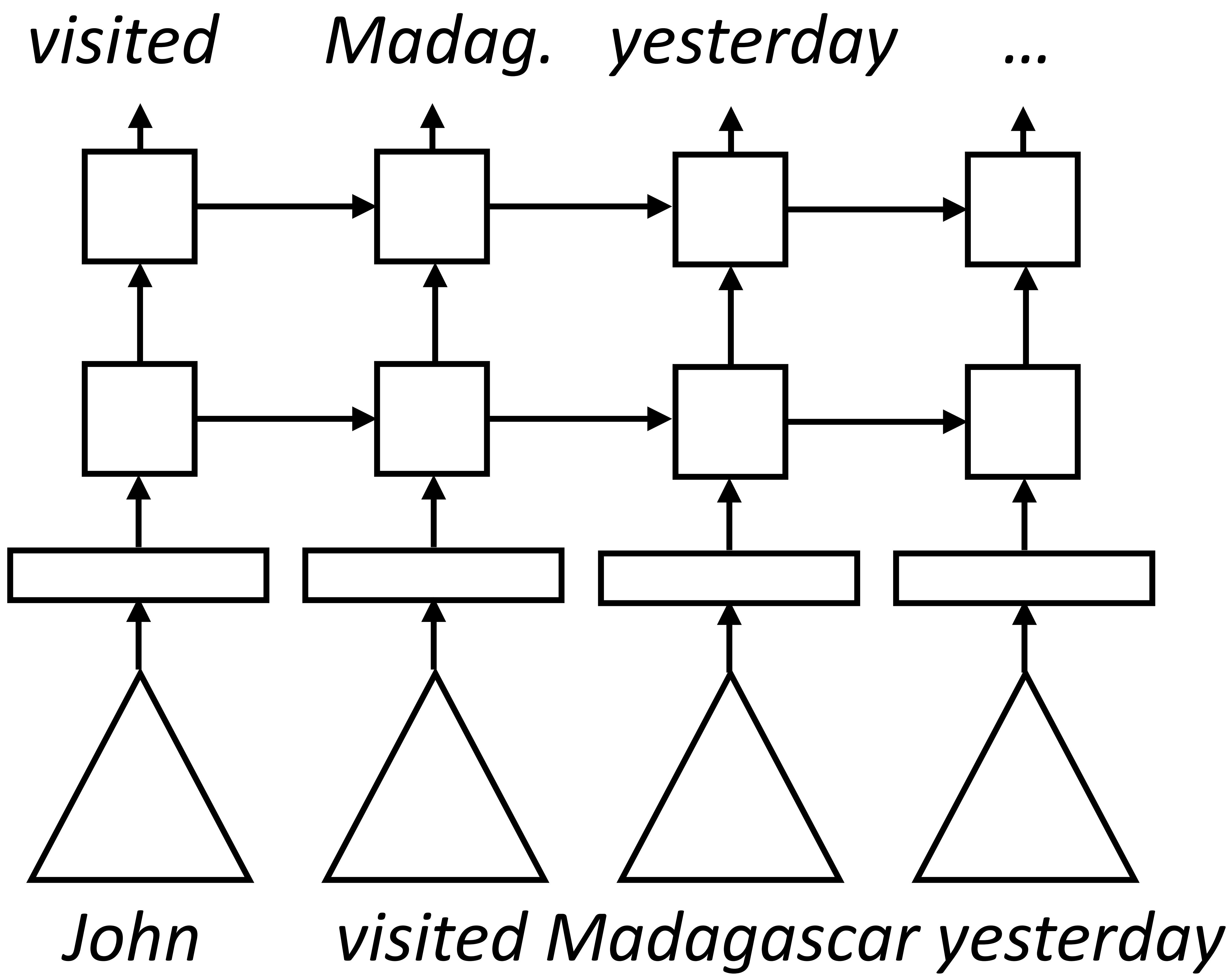


Devlin et al. (2019)

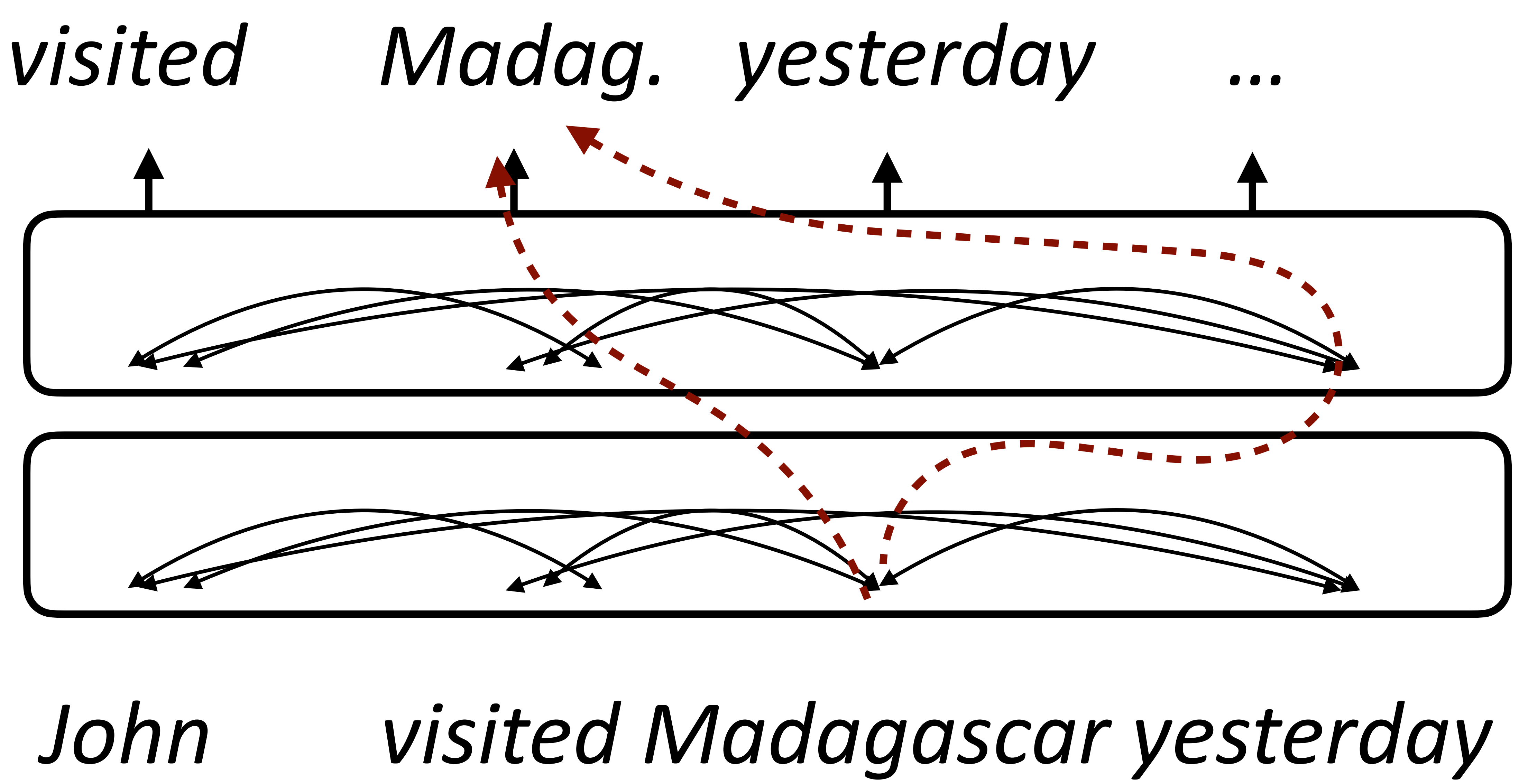
BERT

- How to learn a "deeply bidirectional" model? What happens if we just replace an LSTM with a transformer?

ELMo (Language Modeling)



BERT

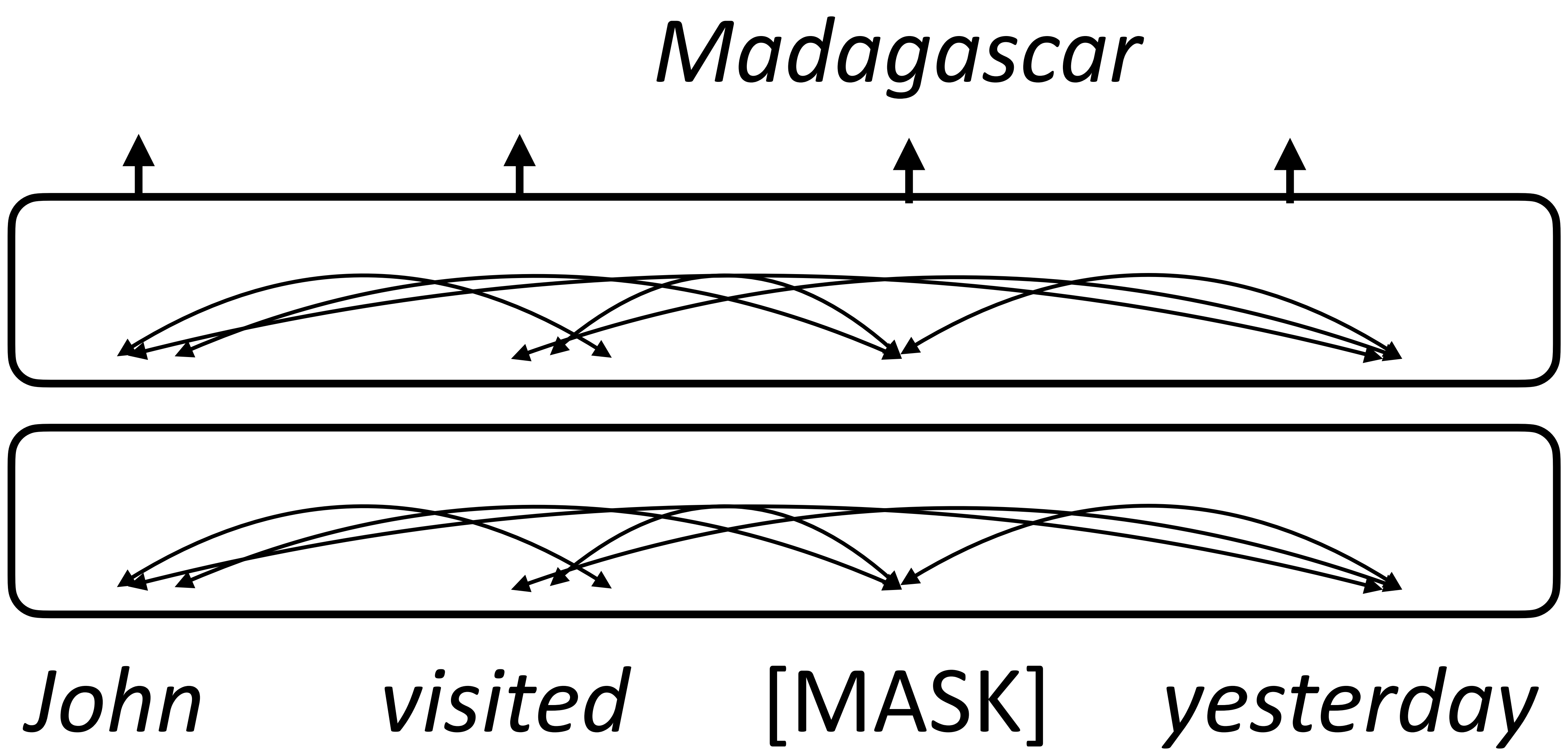


- You could do this with a "one-sided" transformer, but this "two-sided" model can cheat

Masked Language Modeling

- How to prevent cheating? Next word prediction fundamentally doesn't work for bidirectional models, instead do *masked language modeling*

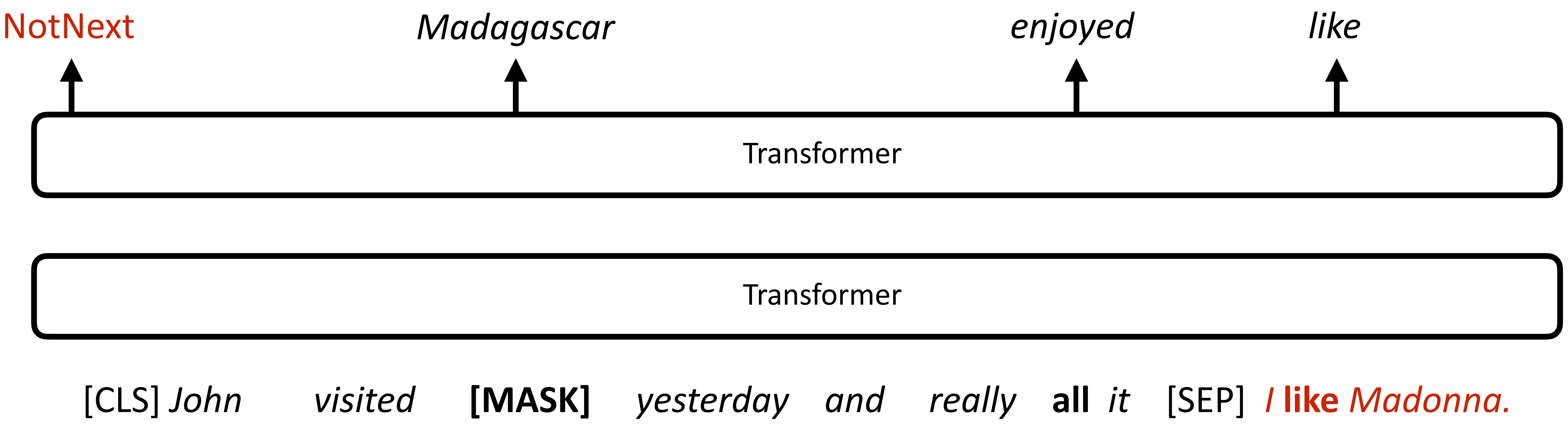
- BERT formula: take a chunk of text, mask out 15% of the tokens, and try to predict them



Devlin et al. (2019)

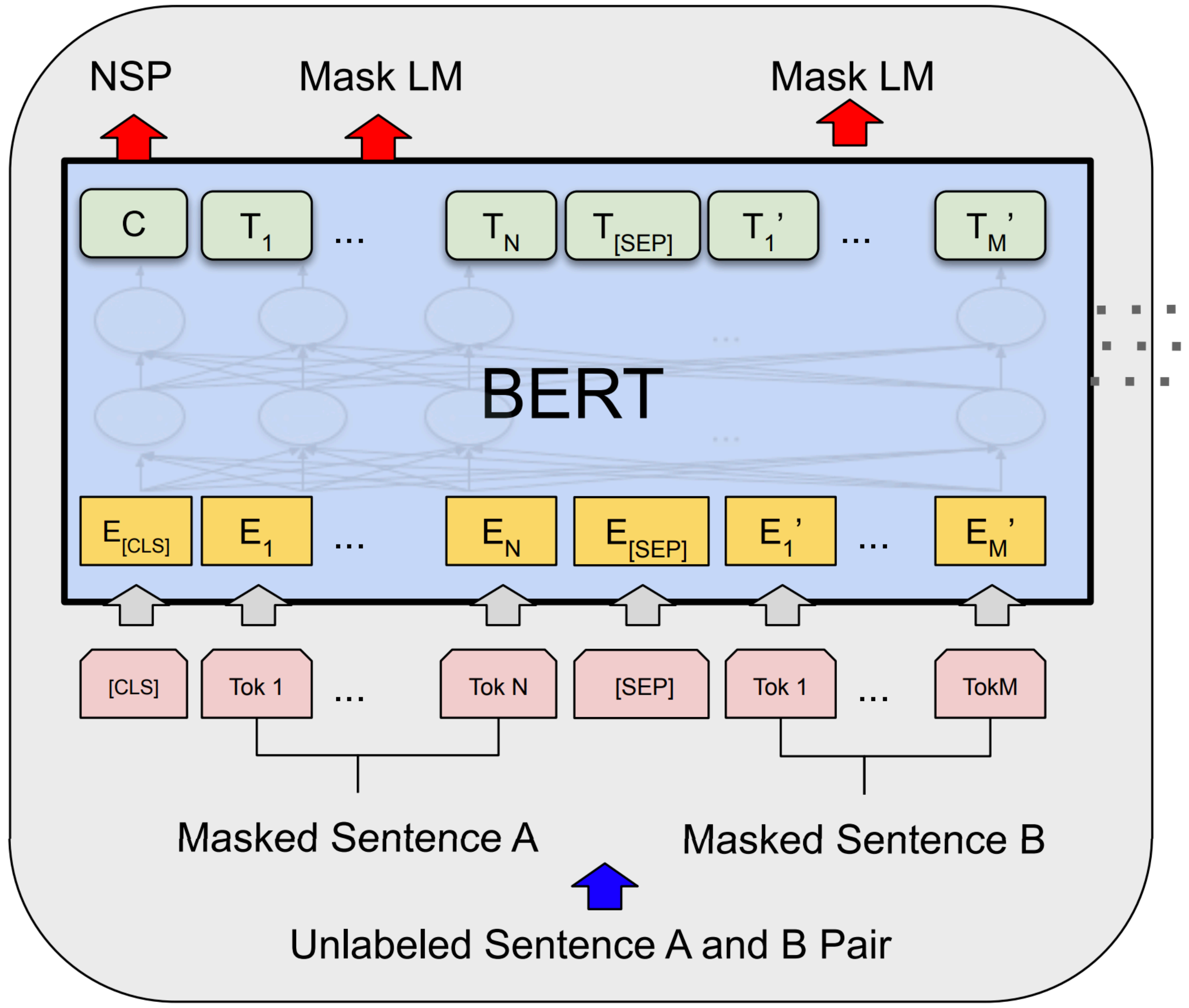
BERT objective

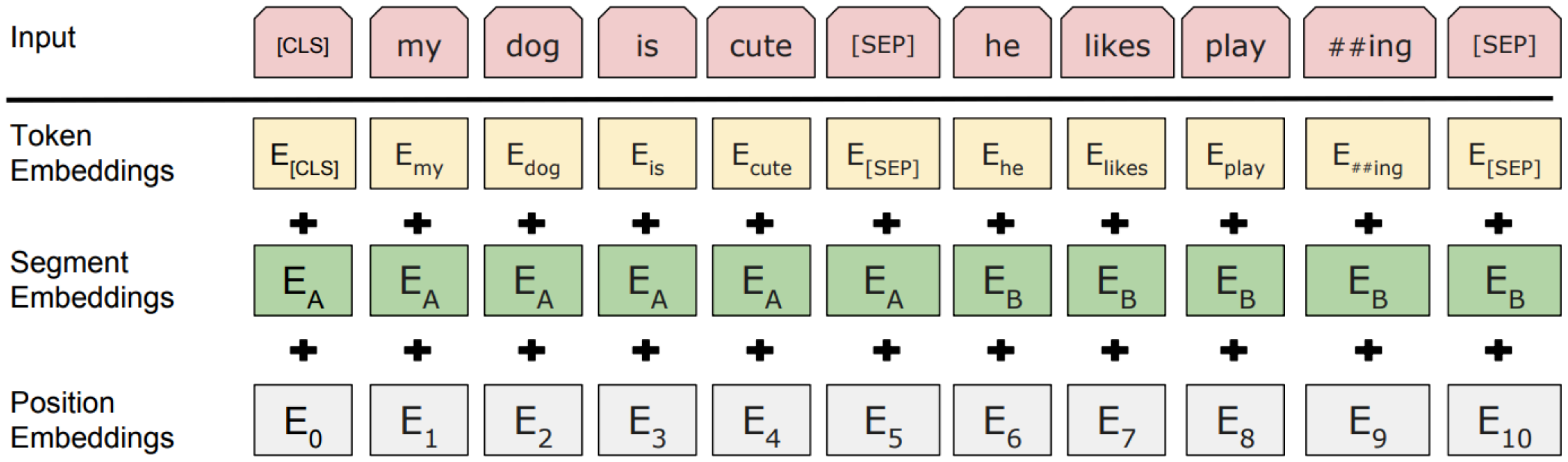
- Language Modeling Objective (Predicting the masked word)
- Next Sentence Prediction Objective
  - Input: [CLS] Text chunk 1 [SEP] Text chunk 2
  - 50% of the time, take the true next chunk of text, 50% of the time take a random other chunk. Predict whether the next chunk is the "true" next



BERT architecture details

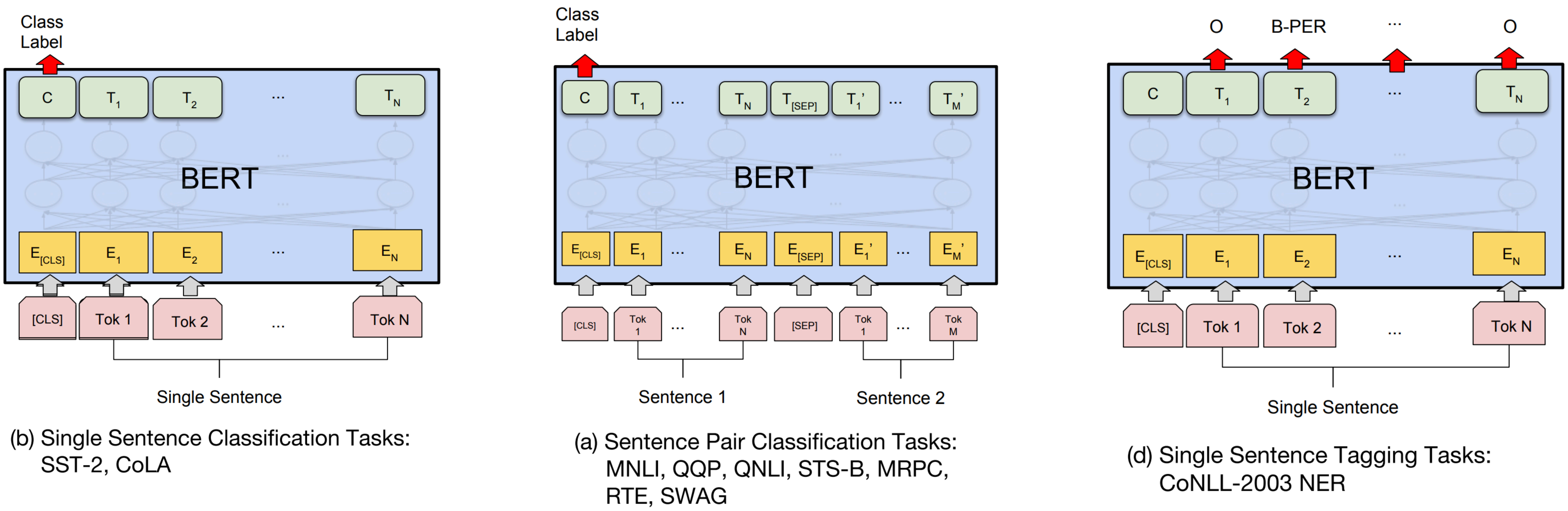
- BERT Base: 12 layers, 768-dim per wordpiece token, 12 heads. Total params = 110M
- BERT Large: 24 layers, 1024-dim per wordpiece token, 16 heads. Total params = 340M
- word pieces instead of words: playing => play ##ing
- Positional embeddings and segment embeddings
- **pre-trained** on a large corpus (40 epochs on Wikipedia (2.5B tokens) + BookCorpus (0.8B tokens)





Devlin et al. (2019)

What can BERT do?

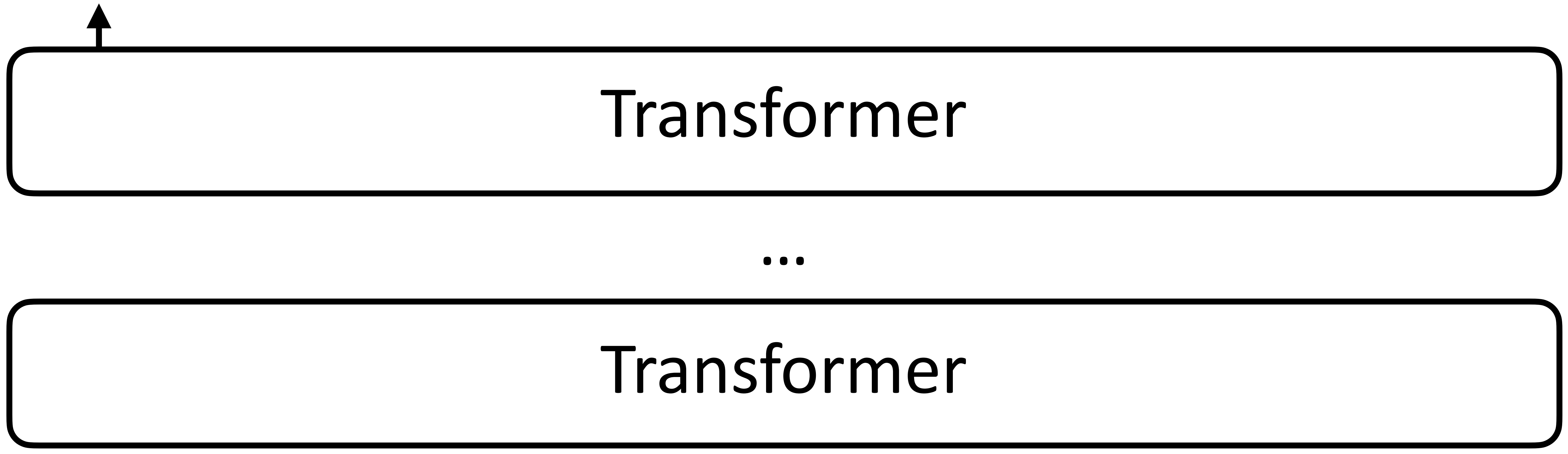


- Artificial [CLS] token is used as the vector to do classification from
- Sentence pair tasks (entailment): feed both sentences into BERT
- BERT can also do tagging by predicting tags at each word piece

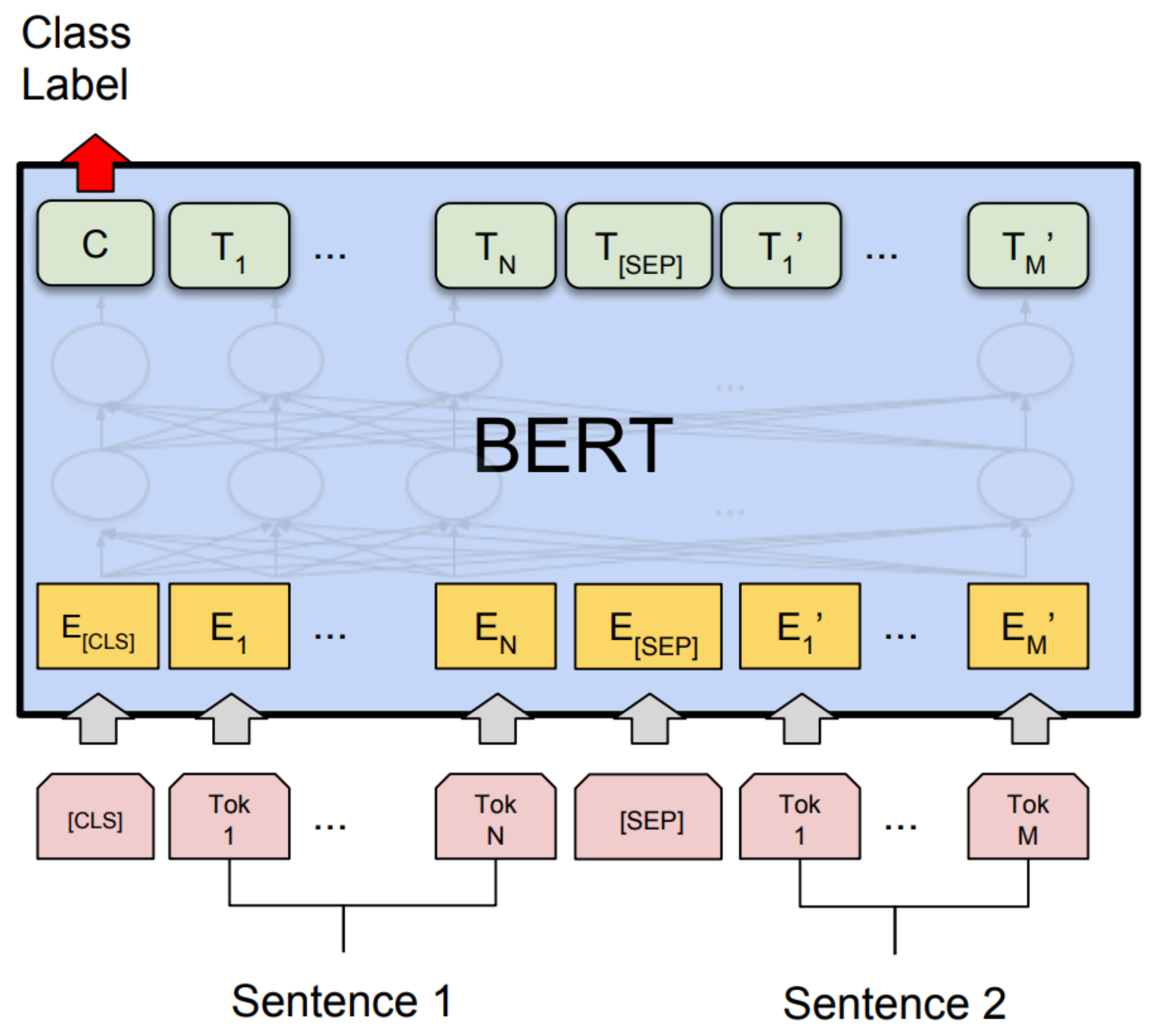
Devlin et al. (2019)

What can BERT do?

Entails (first sentence implies second is true)



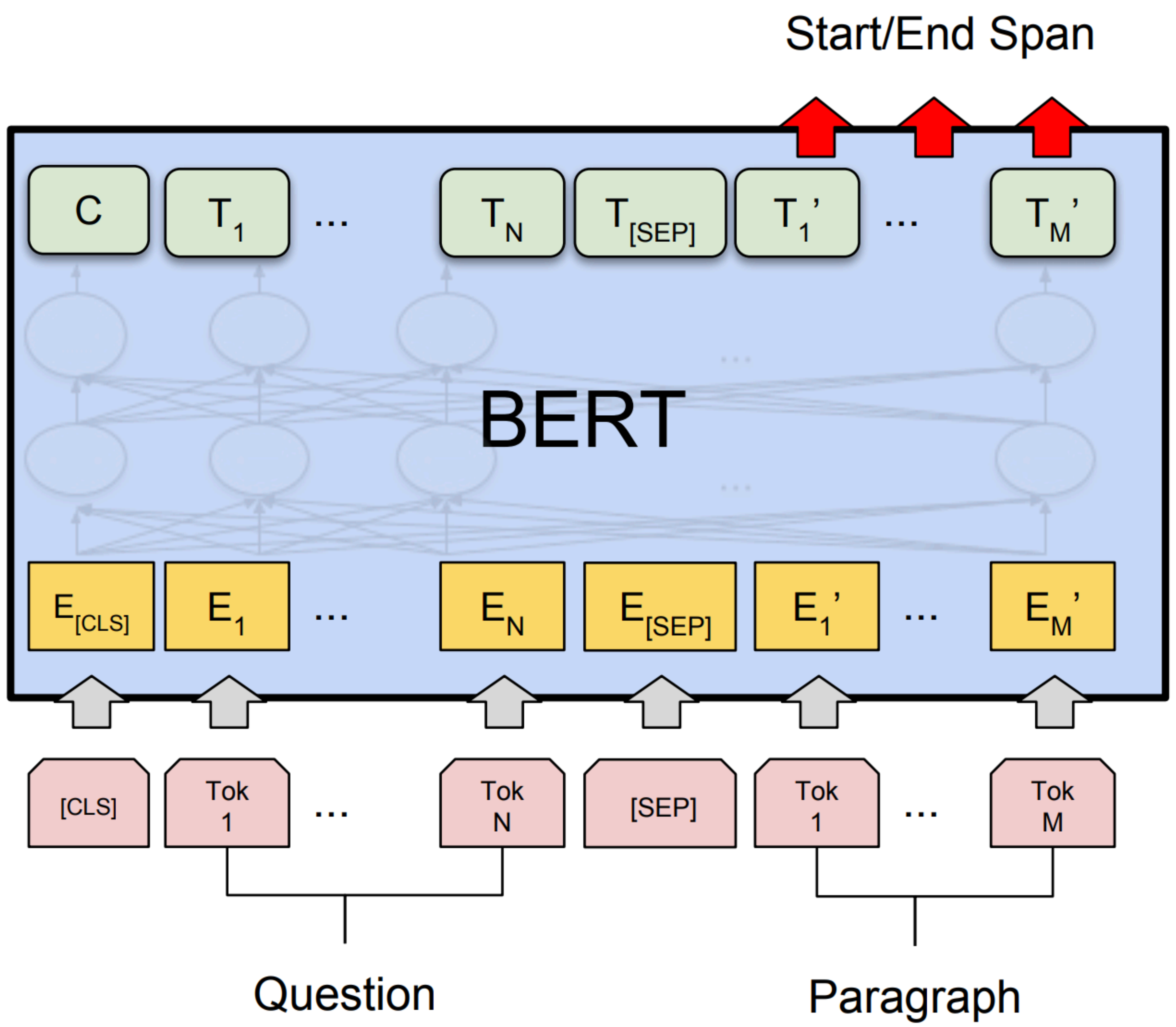
[CLS] A boy plays in the snow [SEP] A boy is outside



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

- How does BERT model a pair of sentences?
- Transformers can model interactions between the two sentences

QA with BERT

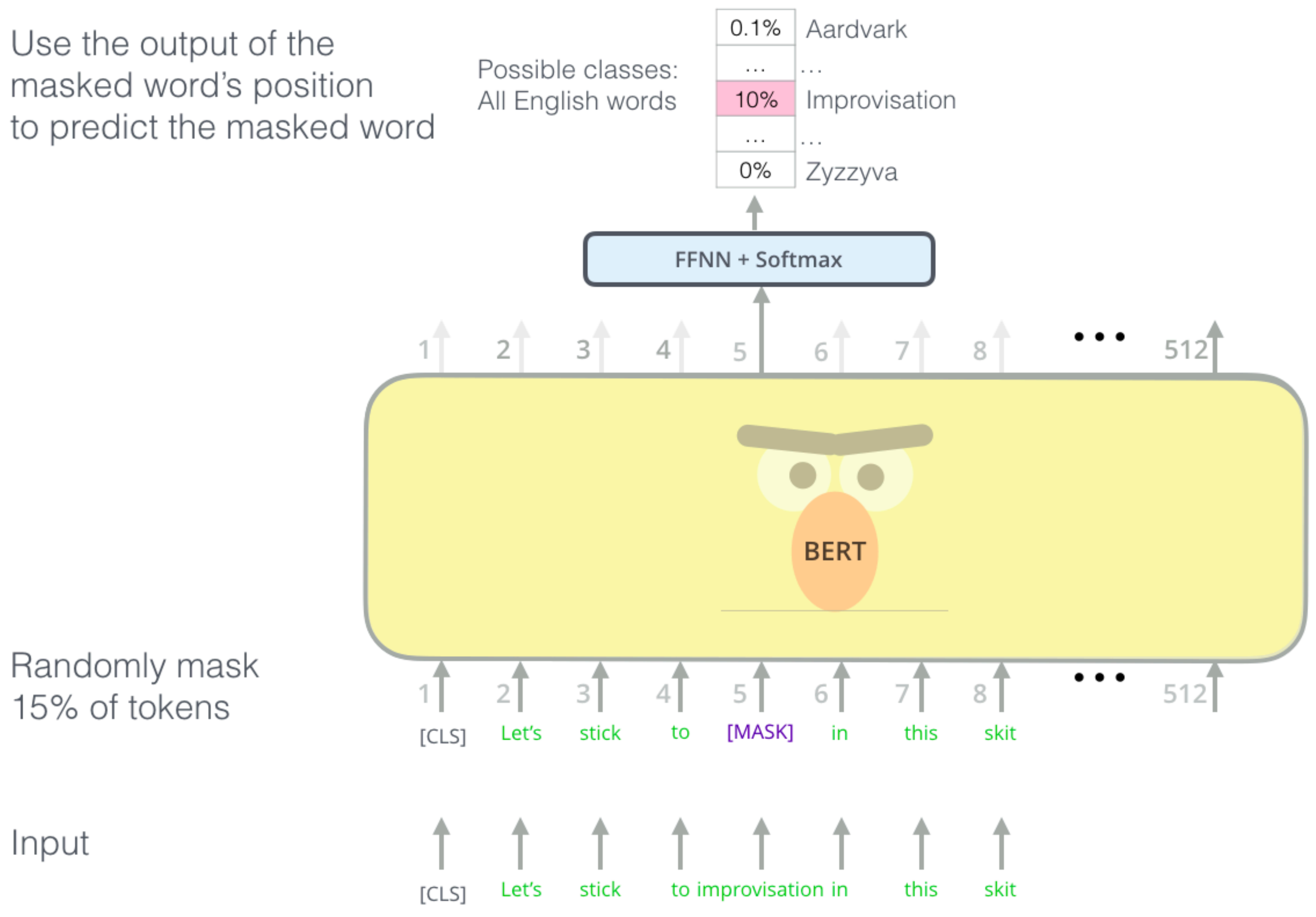


What was Marie Curie the first female recipient of ? [SEP] One of the most famous people born in Warsaw was Marie …

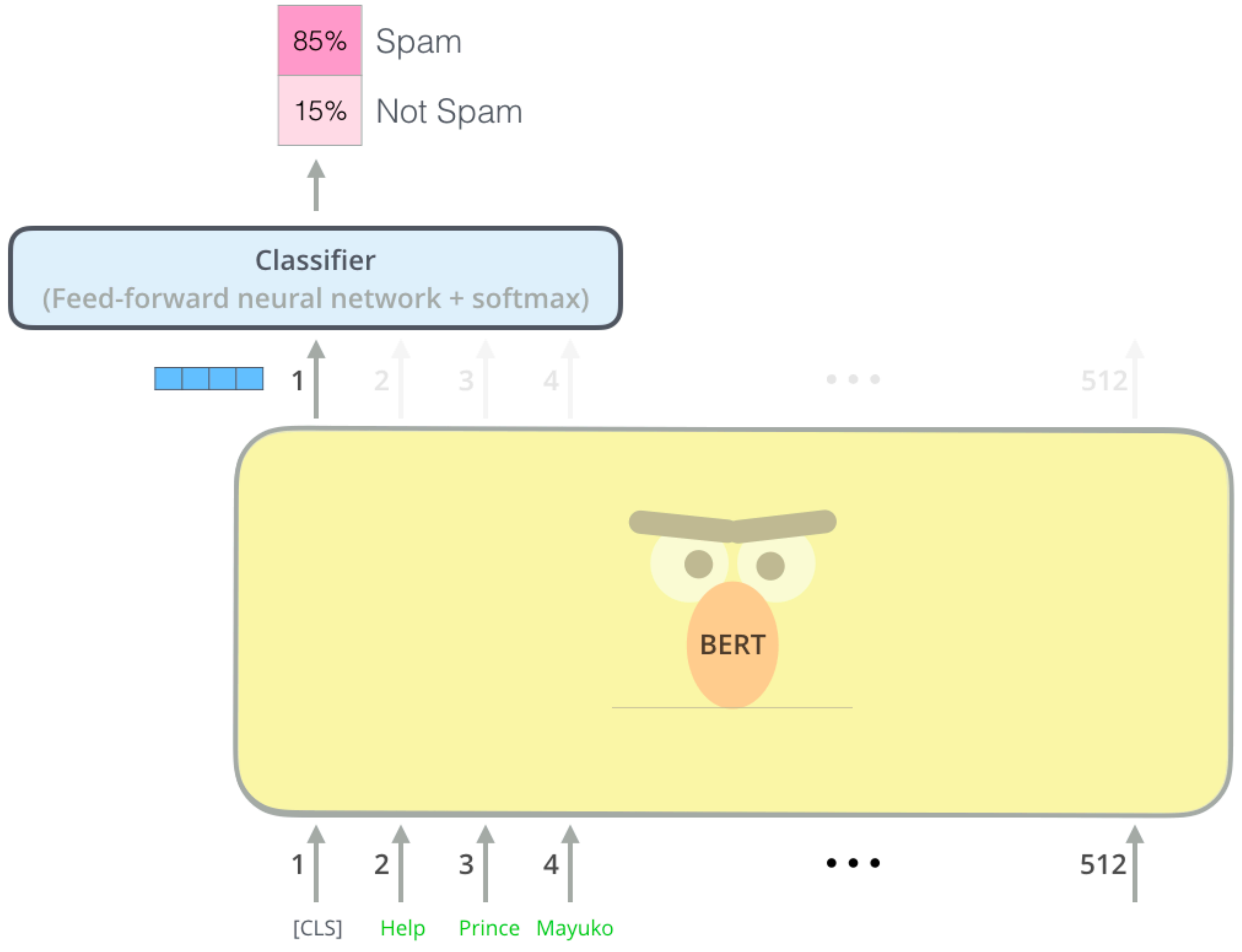
- Predict start and end positions in passage

Devlin et al. (2019)

Pretraining and fine-tuning



Pre-training



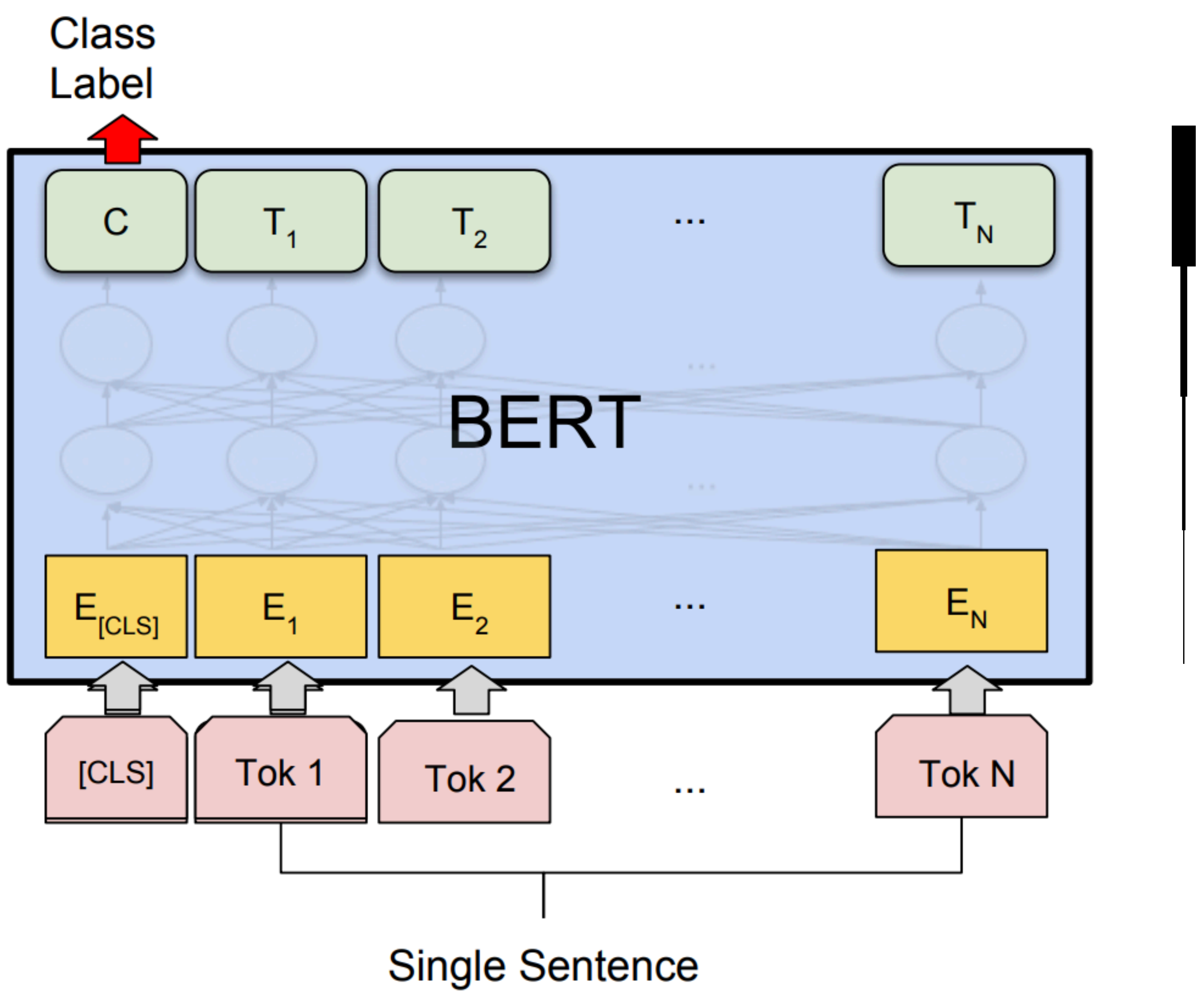
Fine-tuning

language modeling objective

task-specific objective

Fine-tuning BERT

- Fine-tune for 1-3 epochs, batch size 2-32, learning rate 2e-5 - 5e-5



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

- Large changes to weights up here (particularly in last layer to route the right information to [CLS])
- Smaller changes to weights lower down in the transformer
- Small LR and short fine-tuning schedule mean weights don't change much

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	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

- Huge improvements over prior work (even compared to ELMo)
- Effective at "sentence pair" tasks: textual entailment (does sentence A imply sentence B), paraphrase detection

Devlin et al. (2018)

RoBERTa

- "Robustly optimized BERT"
- 160GB of data instead of 16 GB
- Dynamic masking: standard BERT uses the same MASK scheme for every epoch, RoBERTa recomputes them

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

- New training hyperparameters + more data = better performance

Liu et al. (2019)

Design Choices for Language Model

- Tokenization:
  - how do you segment text? how do you construct the vocabulary?
- Model Architecture:
  - LSTM / CNN / Transformer (or combinations of them)
  - Hyper-parameters (hidden dimensions, etc)
- Learning Objective
  - During Pre-training
  - During Task-specific Fine-tuning

Question:

What are design choices that impact the performances of pre-trained language models?

How would they interact with each other?

(e.g., if you decide to train with a large amount of data instead of moderately size of the data, how would it interact with the size of the architecture?)

What can BERT NOT do?

- BERT **cannot** generate text (at least not in an obvious way)
  - Can fill in MASK tokens, but can't generate left-to-right (well, you could put MASK at the end repeatedly, but this is slow)

Timeline of Pretrained LM

First general purpose LM: ELMo (2018)

Precursor to ELMo (2017)
Using LM for sequence tagging

Masked LM:
BERT (late 2018)

Seq2Seq Pretraining:
T5, BART(2019)

Multilingual Language Model: XLM (2019)

Multimodal LMs:
Language / Vision / Audio
(2019-)

Efficient LM:
ELECTRA / ALBERT
(2020)

Larger LM:
GPT3 (2020)

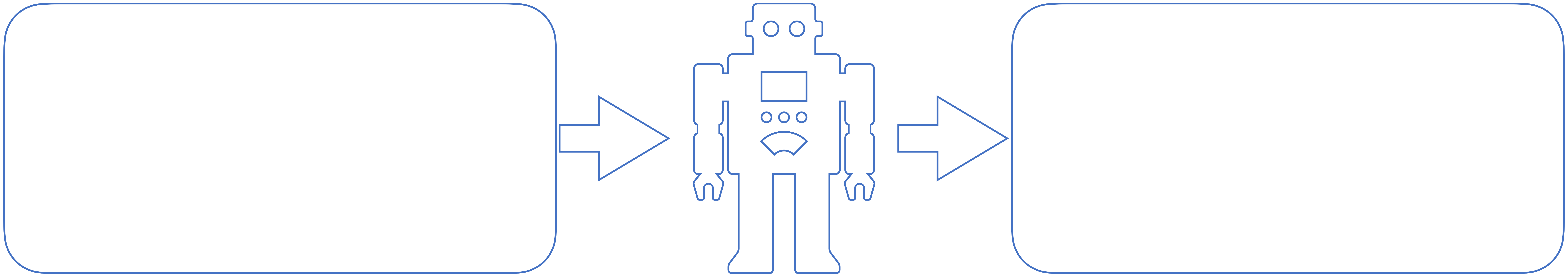
Even larger LM
LM + search

Goal: Use pre-training for conditional text generation

LMs

- Unconditional Text generation,
- You generate text left to right
- $P(w_{i+1} \mid w_1, w_2, \ldots, w_i)$

Conditional Text Generation



Input:

Output:

Conditional Text Generation

Input: Text in {English,…}

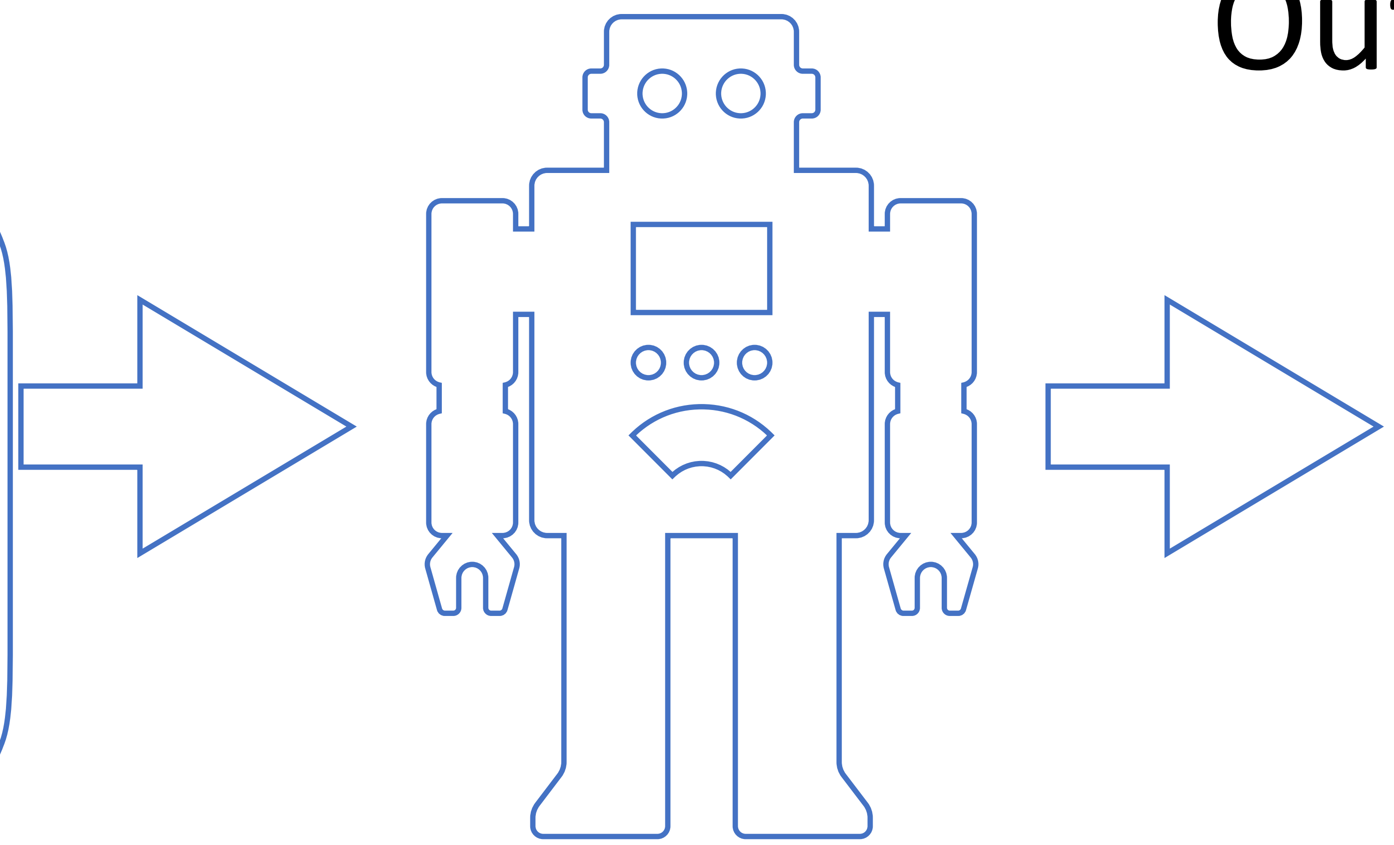
Yesterday was Sunday.

Output: Text in {Portuguese,…}

Ontem foi domingo.

Input: question

Who is the current president of UT Austin?



Output: answer

Jay Hartzell

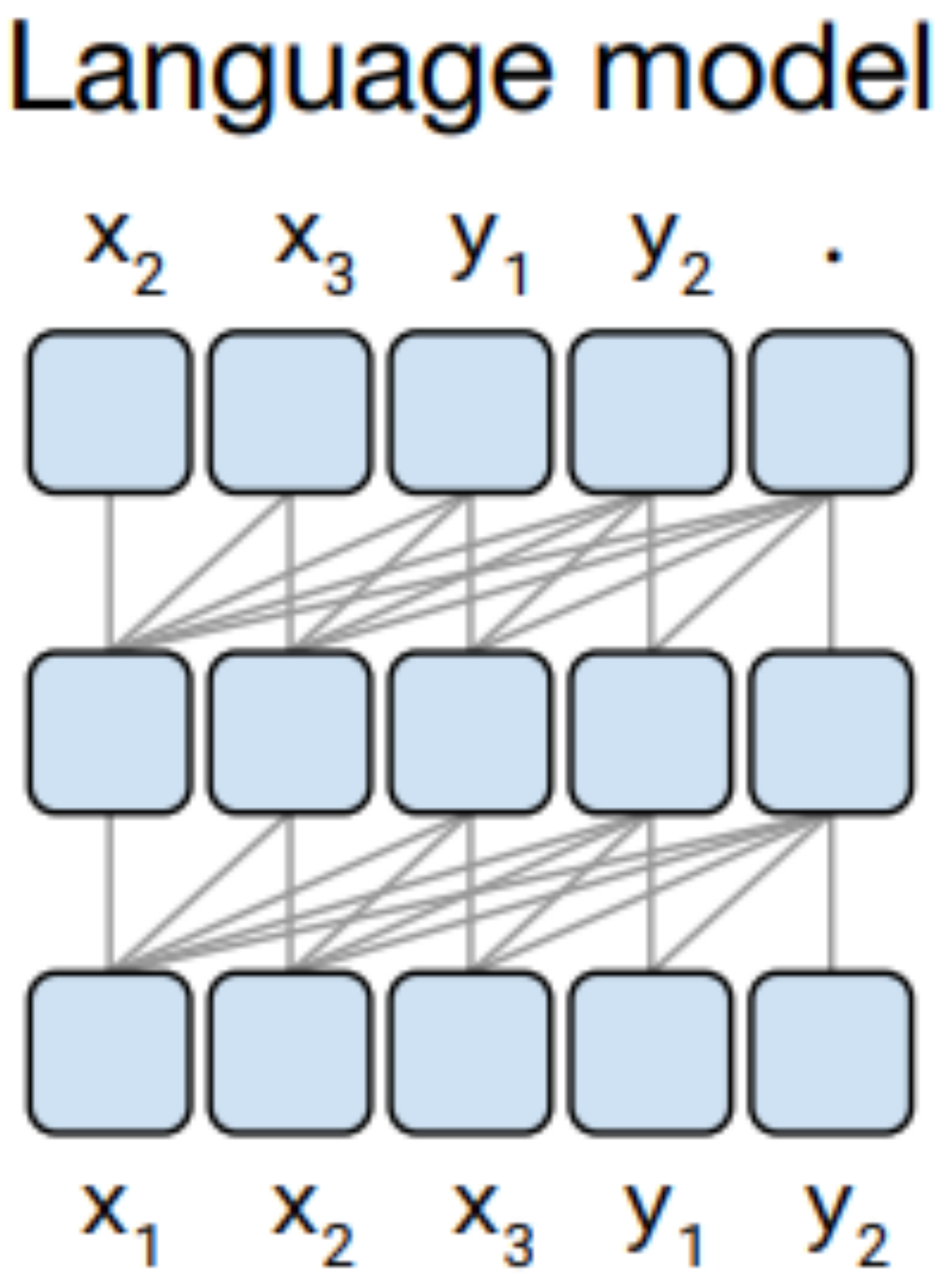
Input: long document

Output: short summary

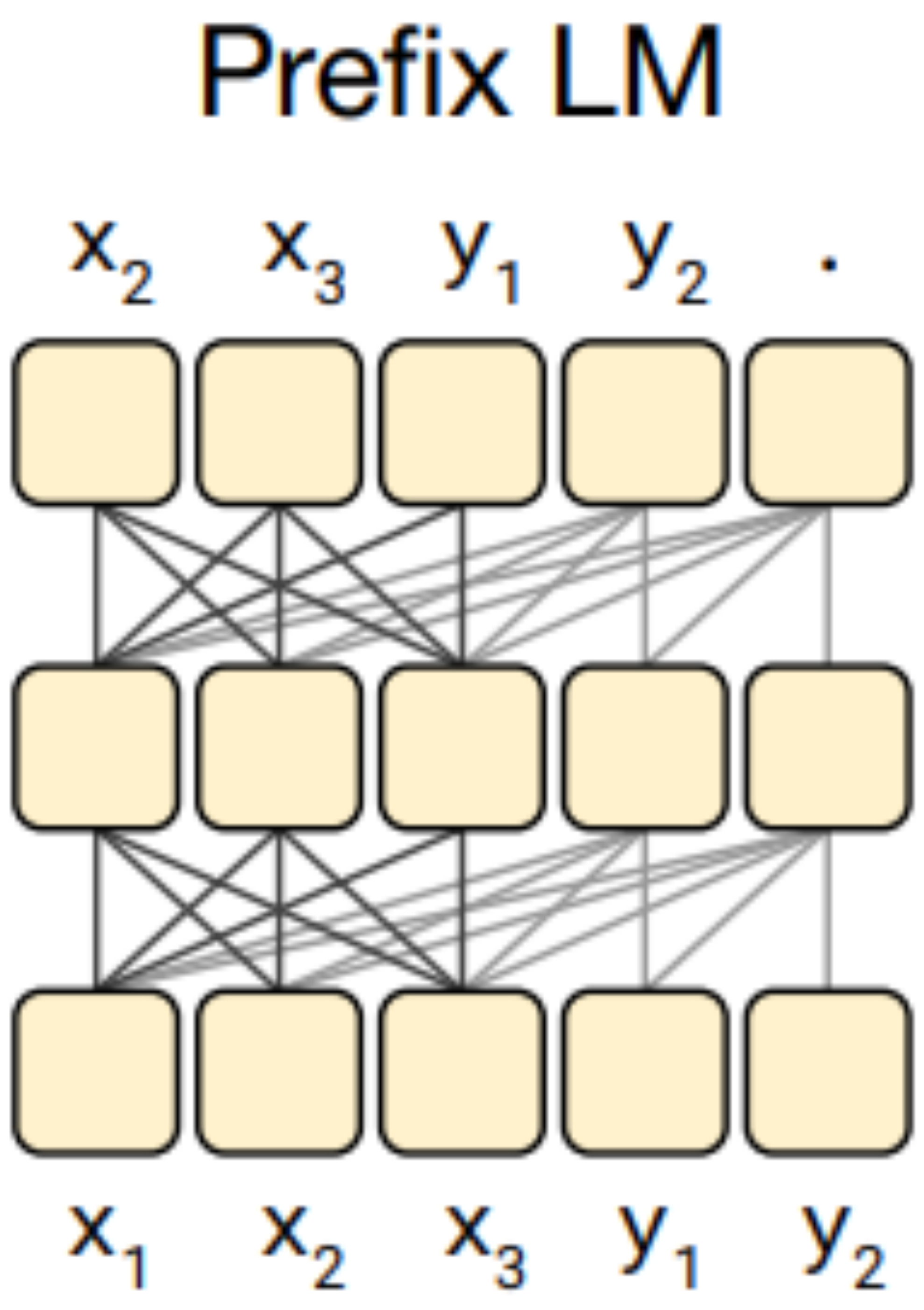
Can we use language model?

- Yes!
  - During training: "X Y"
  - During prediction: "X" and let it generate Y

- Prefix LM:
  - Let tokens in the input to attend to each other
  - Fully visible masking for input
  - Autoregressive masking for output



Problem?



OpenAI GPT/GPT2

- "ELMo with transformers" (works better than ELMo)
- Train a single **unidirectional** transformer LM on long contexts
- GPT2: trained on 40GB of text collected from upvoted links from reddit
- 1.5B parameters — by far the largest of these models trained as of March 2019

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

- Because it's a language model, we can **generate** from it

Radford et al. (2019)

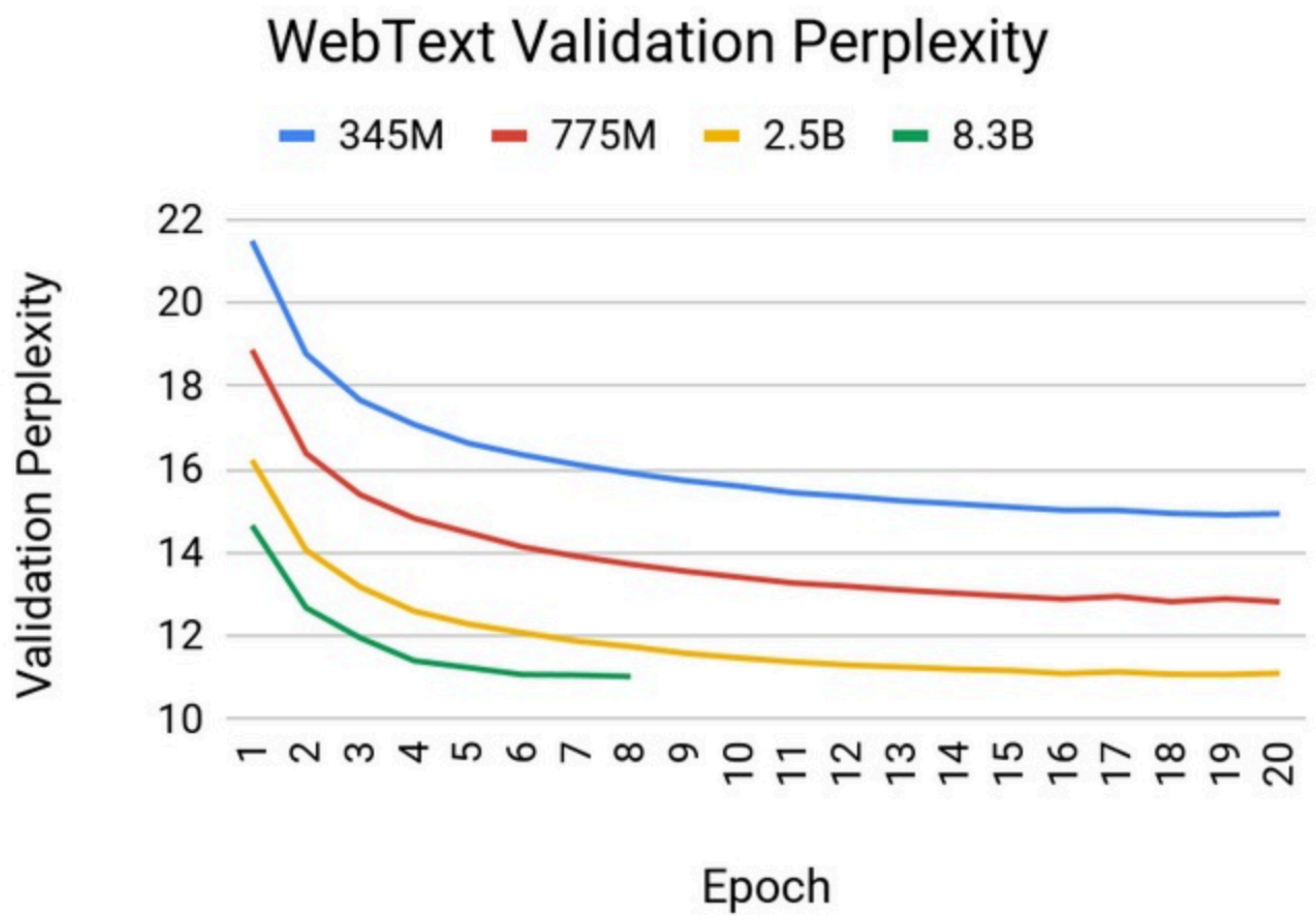
OpenAI GPT2

SYSTEM PROMPT (HUMAN-WRITTEN)	*Miley Cyrus was caught shoplifting from Abercrombie and Fitch on Hollywood Boulevard today.*
MODEL COMPLETION (MACHINE-WRITTEN, SECOND TRY)	The 19-year-old singer was caught on camera being escorted out of the store by security guards. The singer was wearing a black hoodie with the label 'Blurred Lines' on the front and 'Fashion Police' on the back. Scroll down for video Shoplifting: Miley Cyrus was caught shoplifting from Abercrombie and Fitch on Hollywood Boulevard today (pictured) The singer was wearing a black hoodie with the label 'Blurred Lines' on the front and 'Fashion Police' on the back The singer was also wearing a pair of black-rimmed glasses, a black jacket, black jeans and black sandals. She was carrying a pair of black and white striped gloves and a small black bag.

slide credit: OpenAI

GPT-3

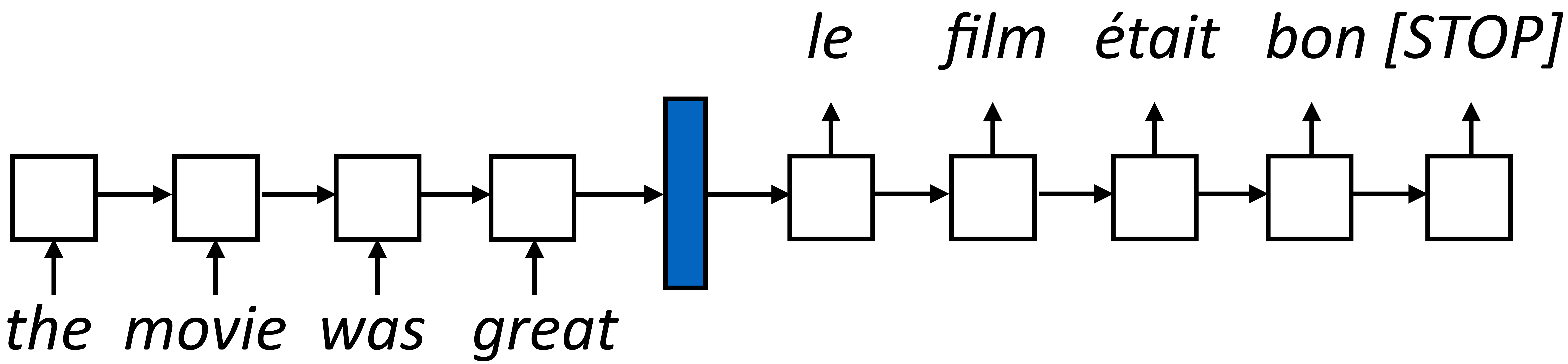
- Same architecture as GPT-2, just larger
  - 1.3B -> 175B parameters
  - Trained on 570GB of Common Crawl
  - Requires 400GB to store parameters!



NVIDIA blog (Narasimhan, August 2019)

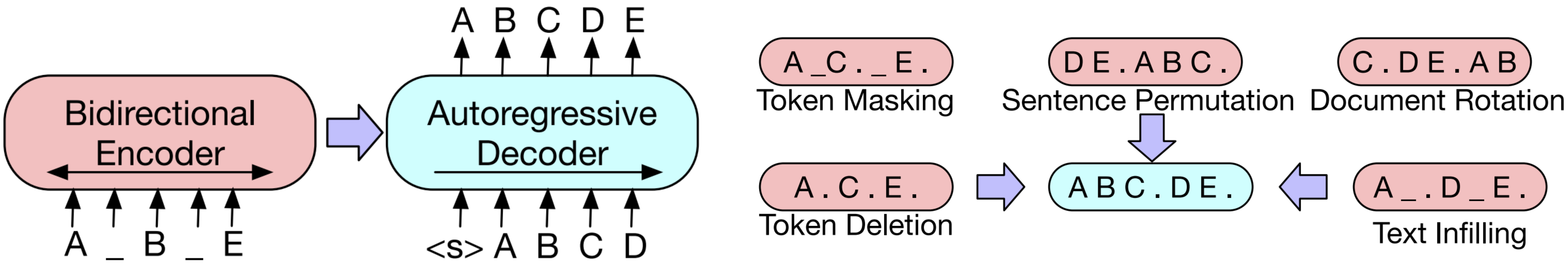
Recap: Seq2Seq Model

- Input: a sequence of tokens
- Output: a sequence of tokens (of ***arbitrary*** length)



BART

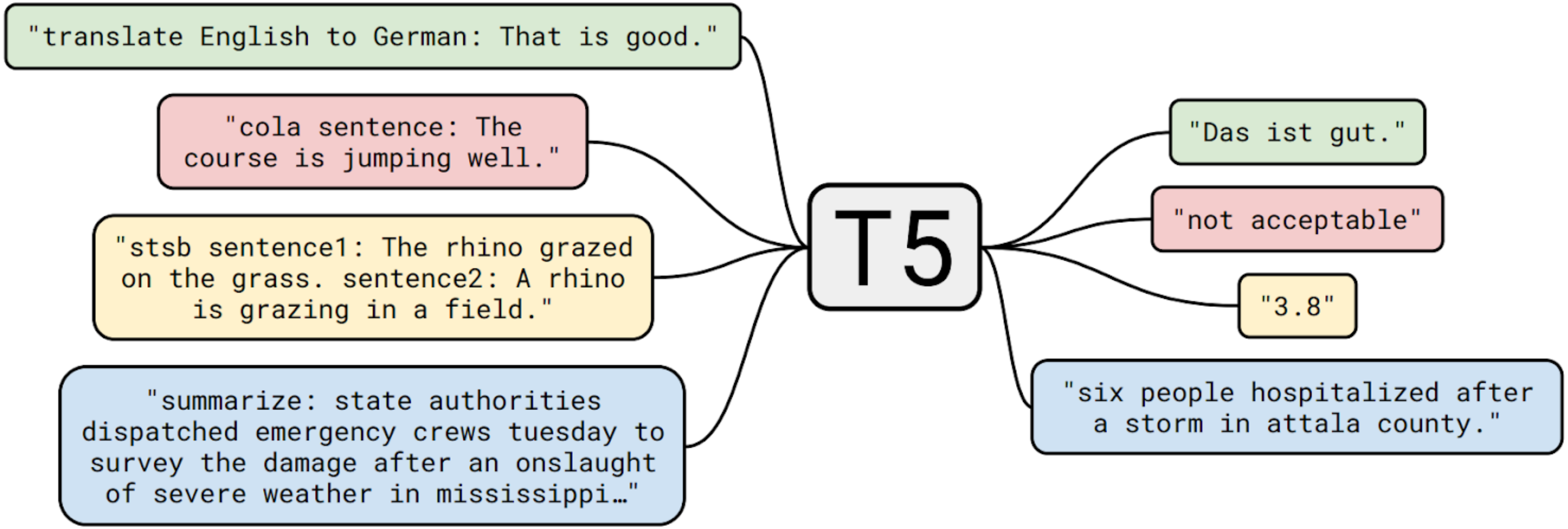
- Objective: Re-construct (corrupted) input sequence



[Lewis et al, 2019]

T5: Text-to-Text Transfer Transformer

- Train on multiple tasks

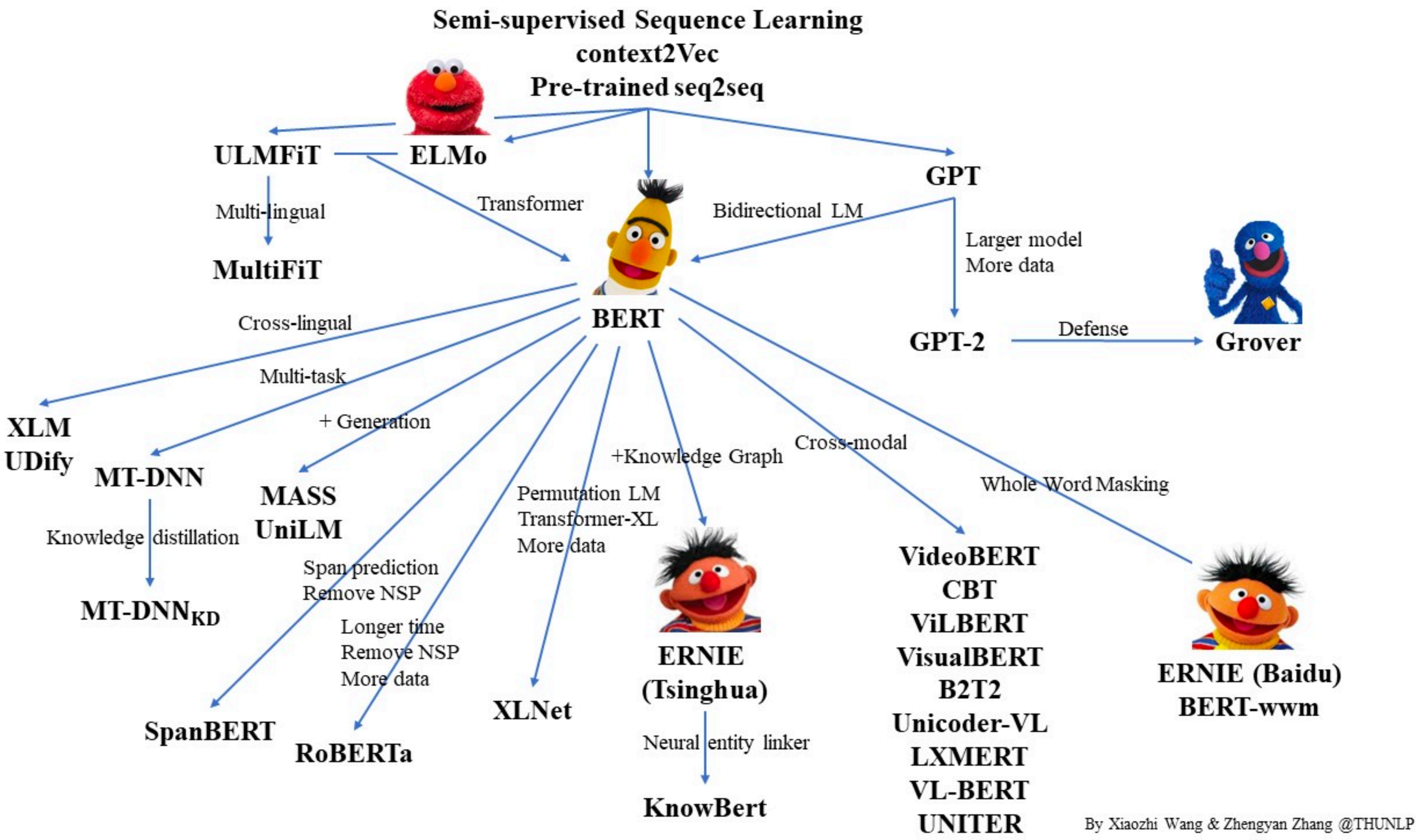


Masked LM vs. SeqSeq

- Considerable win on generation tasks such as summarization, translation
  - Compared to BERT based encoder, randomly initialized decoder
    - 2-3 ROUGE score gains (summarization task metric)

- On classification tasks?
  - Roughly comparable performances, sometimes better, sometimes worse.

many, many variants



Using BERT

- Huggingface Transformers: big open-source library with most pre-trained architectures implemented, weights available

- Lots of standard models…

and "community models"

Model architectures

🤗 Transformers currently provides the following NLU/NLG architectures:

1. **BERT** (from Google) released with the paper BERT: Pre-training of Deep Understanding by Jacob Devlin, Ming-Wei Chang, Kenton Lee and Krist
2. **GPT** (from OpenAI) released with the paper Improving Language Under Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever.
3. **GPT-2** (from OpenAI) released with the paper Language Models are Un Jeffrey Wu*, Rewon Child, David Luan, Dario Amodei** and Ilya Sutskev
4. **Transformer-XL** (from Google/CMU) released with the paper Transform Fixed-Length Context by Zihang Dai*, Zhilin Yang*, Yiming Yang, Jaime
5. **XLNet** (from Google/CMU) released with the paper XLNet: Generalized Understanding by Zhilin Yang*, Zihang Dai*, Yiming Yang, Jaime Carbon
6. **XLM** (from Facebook) released together with the paper Cross-lingual L and Alexis Conneau.
7. **RoBERTa** (from Facebook), released together with the paper a Robustly

…

`mrm8488/spanbert-large-finetuned-tacred` ★

`mrm8488/xlm-multi-finetuned-xquadv1` ★

`nlpaueb/bert-base-greek-uncased-v1` ★

`nlptown/bert-base-multilingual-uncased-sentiment` ★

`patrickvonplaten/reformer-crime-and-punish` ★

`redewiedergabe/bert-base-historical-german-rw-cased` ★

`roberta-base` ★

`severinsimmler/literary-german-bert` ★

`seyonec/ChemBERTa-zinc-base-v1` ★

…

Preview: Next class

You can visit the cemetery where famous Russian composers are buried daily except Thursday

Toronto law to protect squirrels hit by mayor.