Pretraining Language Models (part 1)

Wei Xu

(many slides from Greg Durrett)

This Lecture

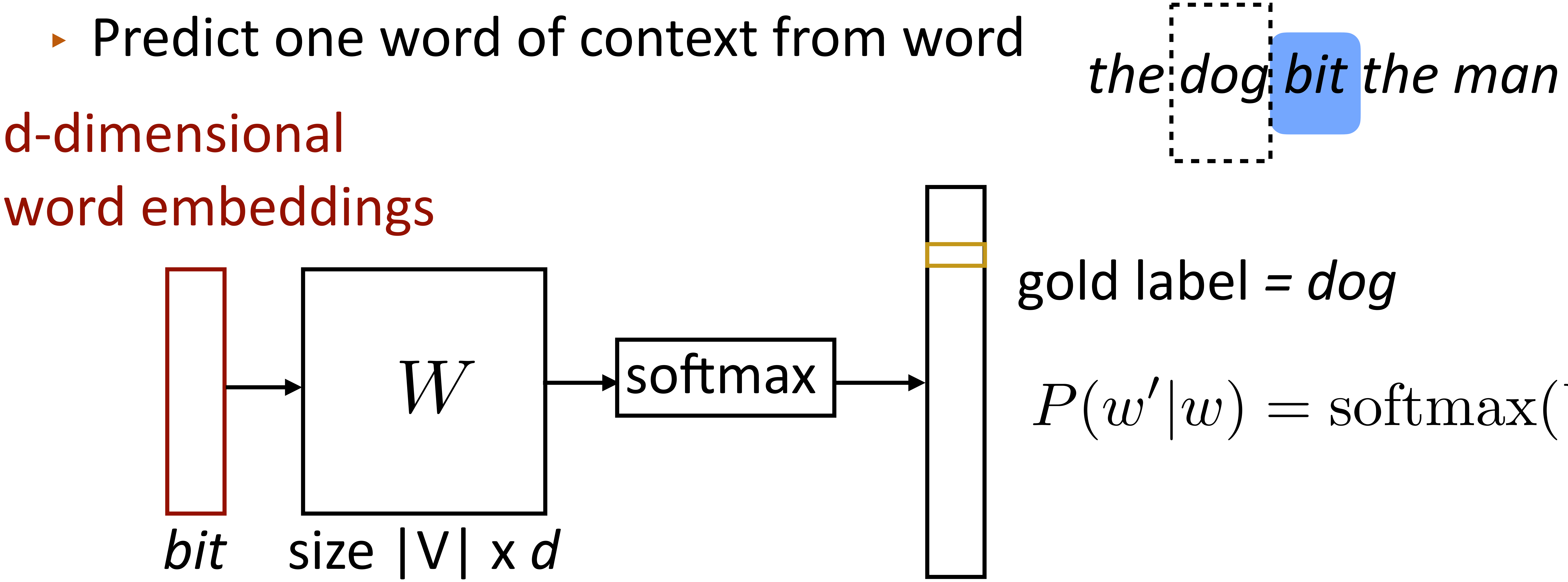
- ELMo
- BERT
- BERT Results, Extensions
- Analysis/Visualization of BERT

Readings

- Readings —
  - J+M 11
  - ELMo by Peters et al.
    https://aclanthology.org/N18-1202.pdf
  - BERT by Devlin et al.
    https://aclanthology.org/N19-1423.pdf

Recall: word2vec (Skip-Gram)

- Predict one word of context from word

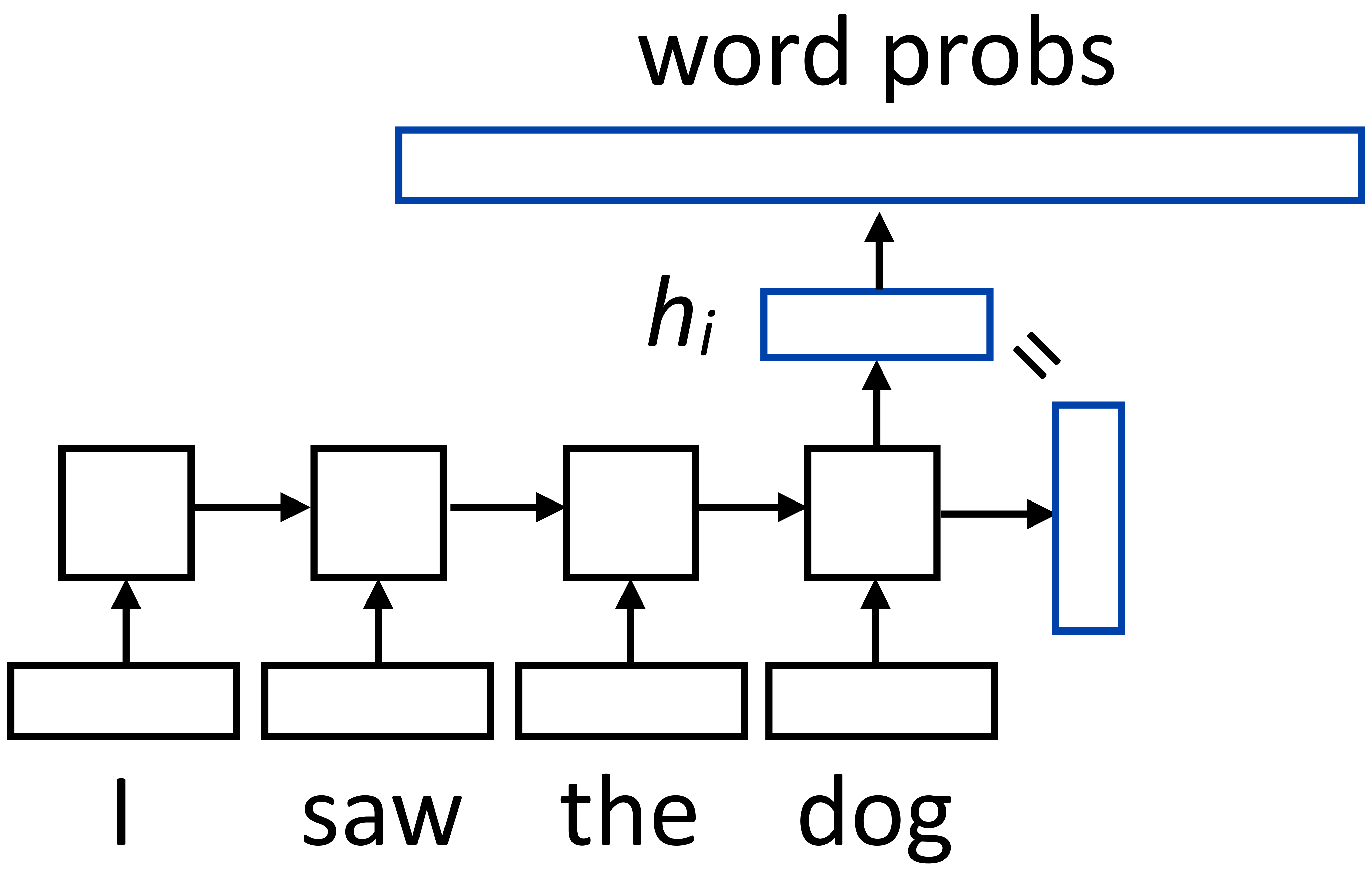


$$P(w'|w) = \mathrm{softmax}(We(w))$$

- Another training example: *bit -> the*
- Parameters: *d* x |V| vectors, |V| x *d* output parameters (W) (also usable as vectors!)

Mikolov et al. (2013)

Recall: RNN Language Modeling



$$P(w|\text{context}) = \text{softmax}(W\mathbf{h}_i)$$

- *W* is a (vocab size) x (hidden size) matrix

Recap: Neural Language Model

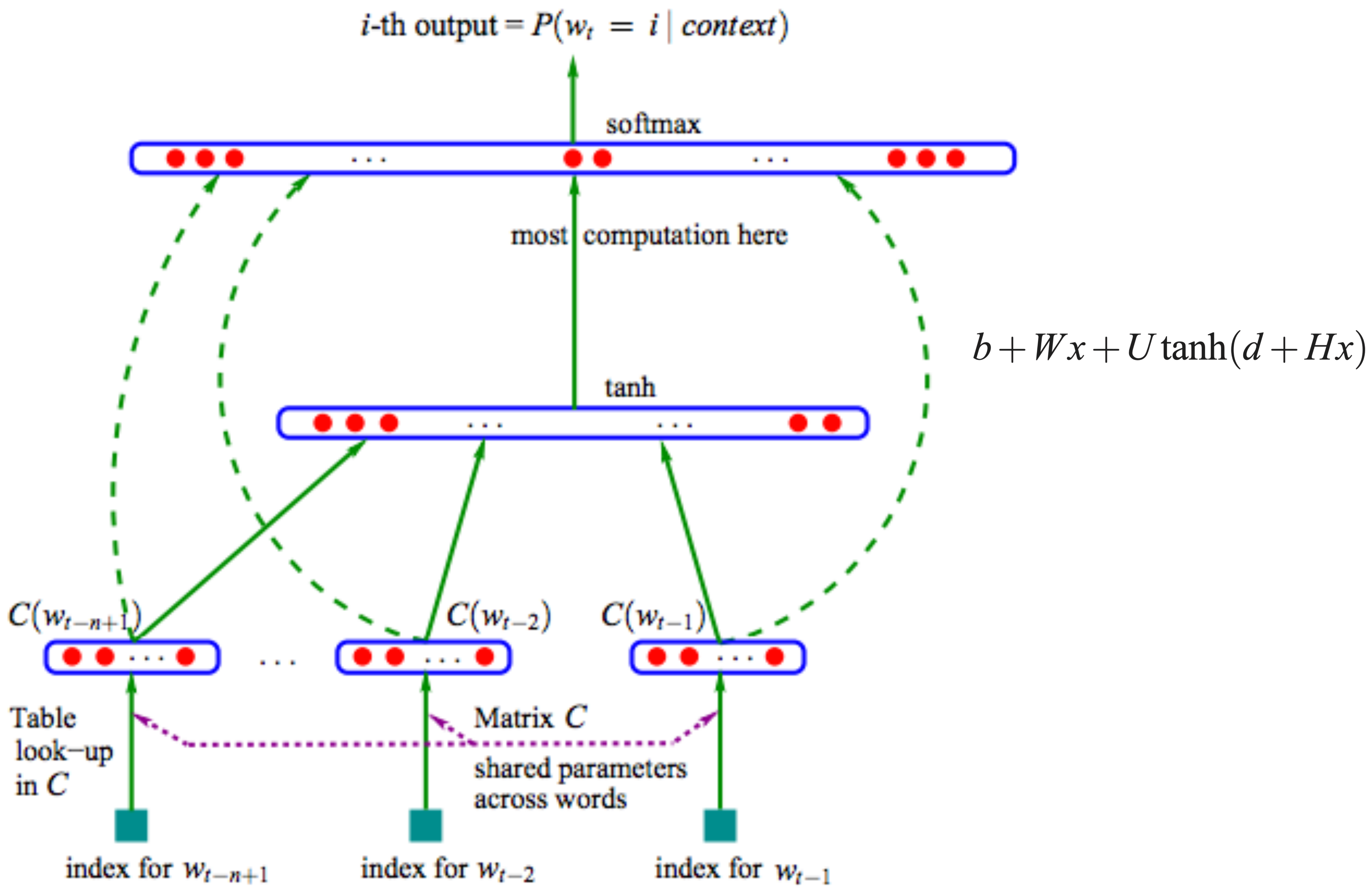
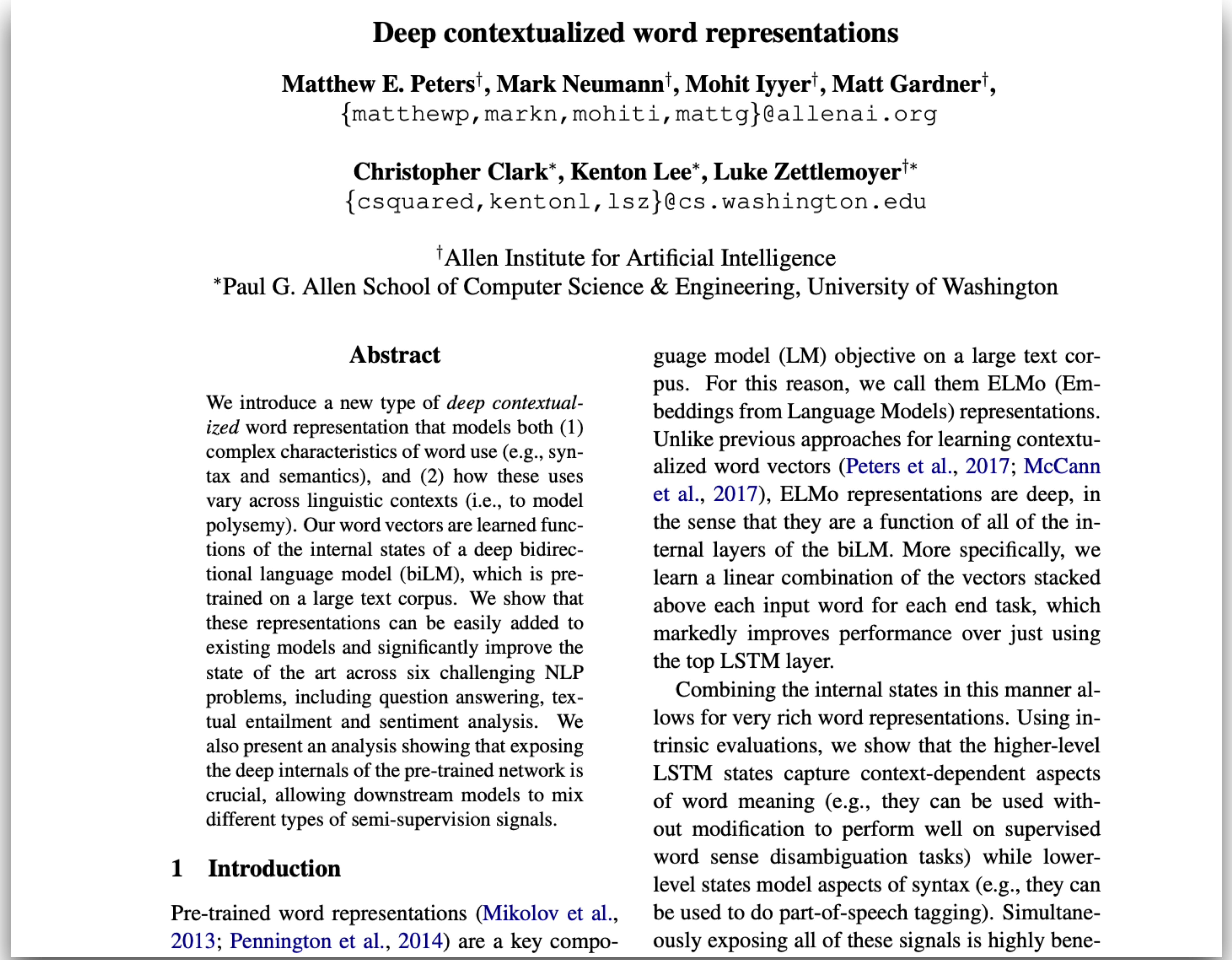


Figure 1: Neural architecture: $f(i, w_{t-1}, \cdots, w_{t-n+1}) = g(i, C(w_{t-1}), \cdots, C(w_{t-n+1}))$ where g is the neural network and $C(i)$ is the i-th word feature vector.

Bengio et al. (2003)

ELMo

ELMo


Deep contextualized word representations

Matthew E. Peters[†], **Mark Neumann**[†], **Mohit Iyyer**[†], **Matt Gardner**[†],
{matthewp,markn,mohiti,mattg}@allenai.org

Christopher Clark[*], **Kenton Lee**[*], **Luke Zettlemoyer**[†*]
{csquared,kentonl,lsz}@cs.washington.edu

[†]Allen Institute for Artificial Intelligence
[*]Paul G. Allen School of Computer Science & Engineering, University of Washington

Abstract

We introduce a new type of *deep contextualized* word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy). Our word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus. We show that these representations can be easily added to existing models and significantly improve the state of the art across six challenging NLP problems, including question answering, textual entailment and sentiment analysis. We also present an analysis showing that exposing the deep internals of the pre-trained network is crucial, allowing downstream models to mix different types of semi-supervision signals.

1 Introduction

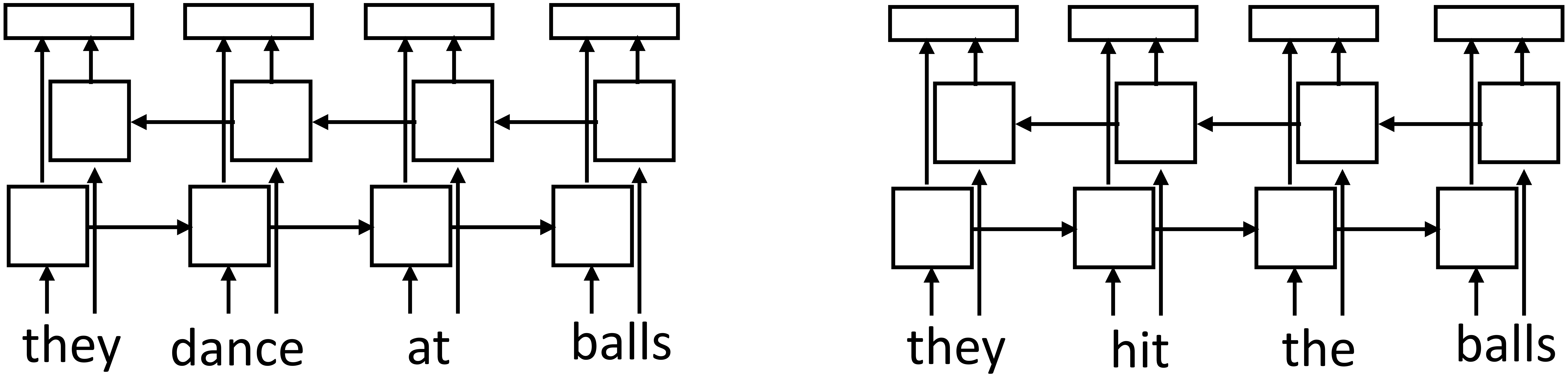
Pre-trained word representations (Mikolov et al., 2013; Pennington et al., 2014) are a key compo-

guage model (LM) objective on a large text corpus. For this reason, we call them ELMo (Embeddings from Language Models) representations. Unlike previous approaches for learning contextualized word vectors (Peters et al., 2017; McCann et al., 2017), ELMo representations are deep, in the sense that they are a function of all of the internal layers of the biLM. More specifically, we learn a linear combination of the vectors stacked above each input word for each end task, which markedly improves performance over just using the top LSTM layer.

Combining the internal states in this manner allows for very rich word representations. Using intrinsic evaluations, we show that the higher-level LSTM states capture context-dependent aspects of word meaning (e.g., they can be used without modification to perform well on supervised word sense disambiguation tasks) while lower-level states model aspects of syntax (e.g., they can be used to do part-of-speech tagging). Simultaneously exposing all of these signals is highly bene-

Context-dependent Embeddings

- How to handle different word senses? One vector for *balls*



- Train a neural language model to predict the next word given previous words in the sentence, use its internal representations as word vectors
- *Context-sensitive* word embeddings: depend on rest of the sentence
- *Huge* improvements across nearly all NLP tasks over word2vec & GloVe

ELMo - Peters et al. (2018)

ELMo

- Key idea: language models can allow us to form useful word representations in the same way word2vec did

- Take a powerful language model, train it on large amounts of data, then use those representations in downstream tasks

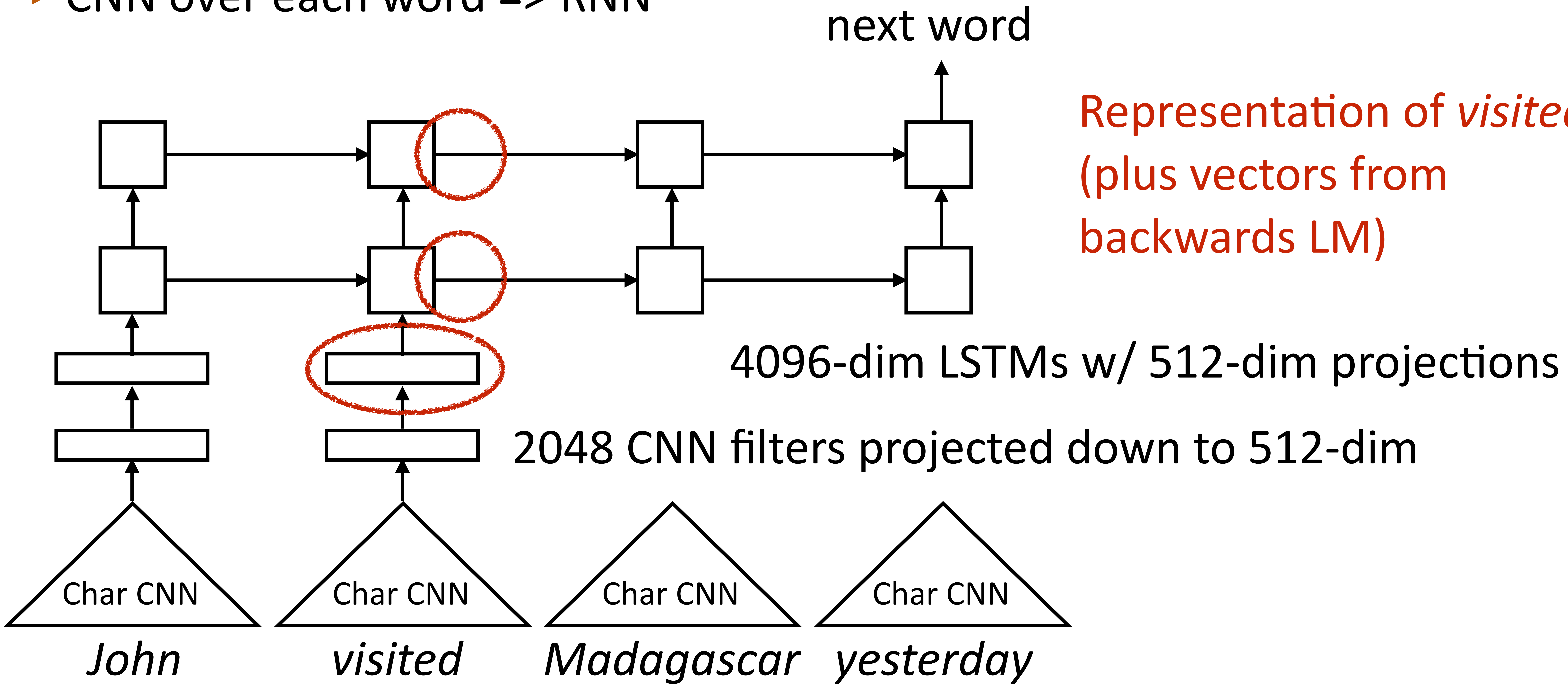
  - Data: Wikipedia, books, crawled stuff from the web, …

- What do we want our LM to look like?

Peters et al. (2018)

ELMo

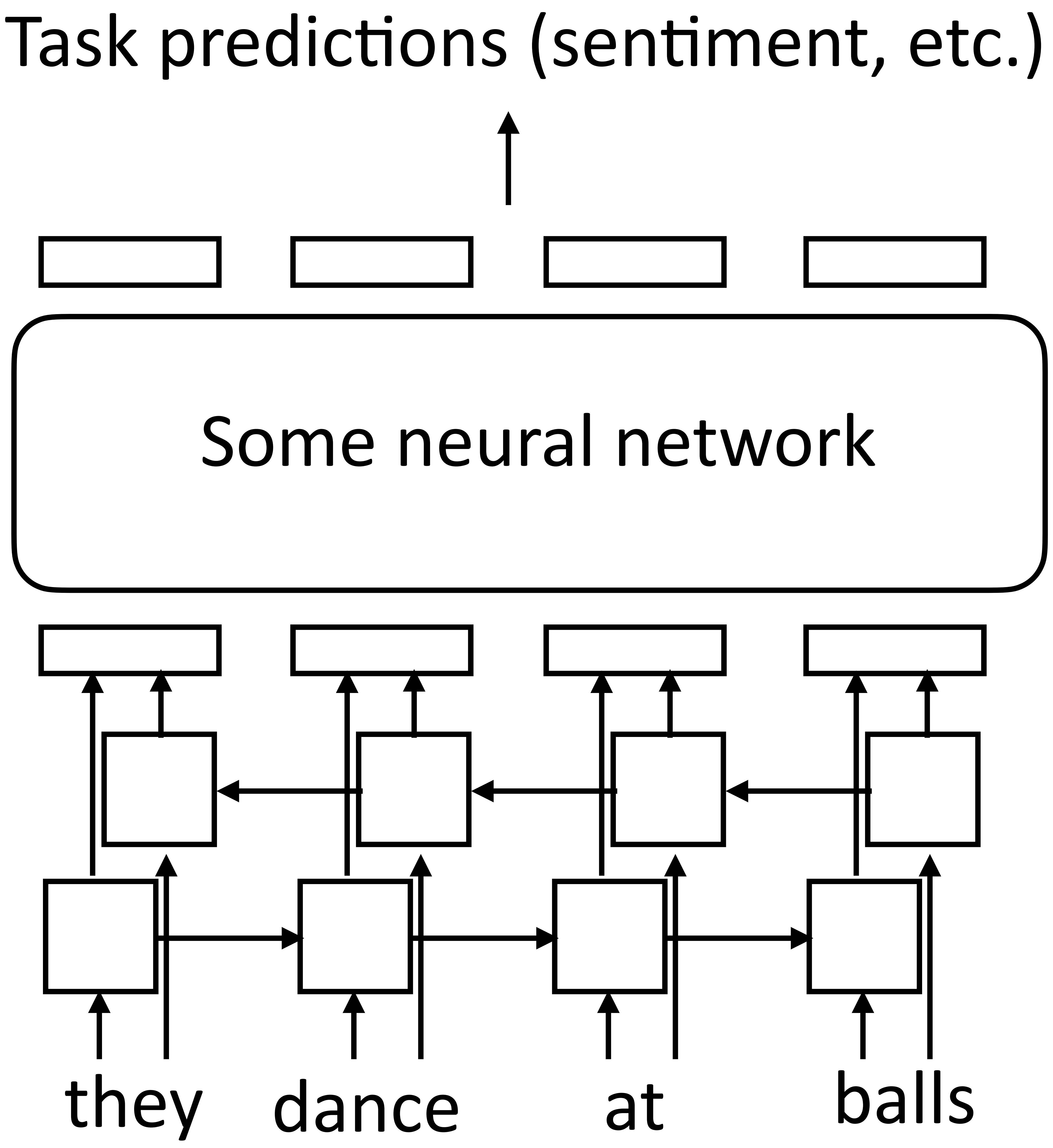
- CNN over each word => RNN



Peters et al. (2018)

How to apply 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
- *Fine-tuning*: backpropagate all the way into ELMo when training your model



Results: Frozen ELMo

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)

How to apply ELMo?

Pretraining	Adaptation	NER CoNLL 2003	SA SST-2	Nat. lang. inference MNLI	SICK-E	Semantic textual similarity SICK-R	MRPC	STS-B
Skip-thoughts	❄️	-	81.8	62.9	-	86.6	75.8	71.8
ELMo	❄️	91.7	**91.8**	**79.6**	**86.3**	**86.1**	**76.0**	**75.9**
	🔥	**91.9**	91.2	76.4	83.3	83.3	74.7	75.5
	Δ=🔥-❄️	0.2	-0.6	-3.2	-3.3	-2.8	-1.3	-0.4

- How does frozen (❄️) vs. fine-tuned (🔥) compare?
- Recommendations:

Conditions: Pretrain	Adapt.	Task	Guidelines
Any	❄️	Any	Add many task parameters
Any	🔥	Any	Add minimal task parameters ⚠️ Hyper-parameters
Any	Any	Seq. / clas.	❄️ and 🔥 have similar performance
ELMo	Any	Sent. pair	use ❄️
BERT	Any	Sent. pair	use 🔥

Peters, Ruder, Smith (2019)

Why did this take time to catch on?

- Earlier version of ELMo by the same authors in 2017, but it was only evaluated on tagging tasks, gains were 1% or less
- Required: training on lots of data, having the right architecture, significant hyperparameter tuning (e.g., GPT-3, T5 …)

OpenReview

OpenReview.net Search OpenReview... Login

Deep contextualized word representations

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer

02 Feb 2018 (modified: 02 Mar 2025) ICLR 2018 Conference Blind Submission Readers: Everyone Show Bibtex Show Revisions

Keywords: representation learning, contextualized word embeddings

TL;DR: We introduce a new type of deep contextualized word representation that significantly improves the state of the art for a range of challenging NLP tasks.

Abstract: We introduce a new type of deep contextualized word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy). Our word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pretrained on a large text corpus. We show that these representations can be easily added to existing models and significantly improve the state of the art across six challenging NLP problems, including question answering, textual entailment and sentiment analysis. We also present an analysis showing that exposing the deep internals of the pretrained network is crucial, allowing downstream models to mix different types of semi-supervision signals.

Community Implementations: 43 code implementations

Withdrawal: Confirmed

Reply Type: all Author: everybody Visible To: all readers Hidden From: nobody **17 Replies**

NLP conference

ICLR 2018 Conference Paper759 Authors

02 Feb 2018, 21:17 ICLR 2018 Conference Paper759 Official Comment Readers: Everyone Show Revisions

Comment: As suggested by the meta reviewer, the empirical results in this paper are better suited for a NLP conference then ICLR. As a result, we are withdrawing it from ICLR. Thank you to the Chairs for your consideration and paper acceptance.

ICLR 2018 Conference Acceptance Decision

ICLR 2018 Conference Program Chairs

29 Jan 2018, 13:11 (modified: 29 Jan 2018, 16:08) ICLR 2018 Conference Acceptance Decision Readers: Everyone Show Revisions

Decision: Accept (Poster)

Comment: This is a good paper that presents state-of-the-art results on a number of challenging NLP tasks. The idea is fairly simple and clean, I therefore expect it to get adopted in the community. It also seems to work across several tasks, which is nice. At the same time it is fairly simple (train an LSTM language model and use representations from all levels of the LSTM in the input or output layer of a supervised task of interest) and hence may be more appropriate for an NLP conference than ICLR?

It is a good paper and it will get cited even though the ML contributions are modest. Accept due to the strong empirical results.

arXiv

arXiv > cs > arXiv:1802.05365

Search... | All fields | Search

Help | Advanced Search

Computer Science > Computation and Language

[Submitted on 15 Feb 2018 (v1), last revised 22 Mar 2018 (this version, v2)]

Deep contextualized word representations

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer

We introduce a new type of deep contextualized word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy). Our word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus. We show that these representations can be easily added to existing models and significantly improve the state of the art across six challenging NLP problems, including question answering, textual entailment and sentiment analysis. We also present an analysis showing that exposing the deep internals of the pre-trained network is crucial, allowing downstream models to mix different types of semi-supervision signals.

Comments: NAACL 2018. Originally posted to openreview 27 Oct 2017. v2 updated for NAACL camera ready

Subjects: **Computation and Language (cs.CL)**

Cite as: arXiv:1802.05365 **[cs.CL]**

(or arXiv:1802.05365v2 **[cs.CL]** for this version)

https://doi.org/10.48550/arXiv.1802.05365

Submission history

From: Matthew Peters [view email]

[v1] Thu, 15 Feb 2018 00:05:11 UTC (135 KB)

[v2] Thu, 22 Mar 2018 21:59:40 UTC (140 KB)

Access Paper:

- View PDF
- TeX Source
- Other Formats

view license

Current browse context:
cs.CL

< prev | next >

new | recent | 2018-02

Change to browse by:
cs

References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

22 blog links (what is this?)

DBLP - CS Bibliography

listing | bibtex

Matthew E. Peters
Mark Neumann
Mohit Iyyer
Matt Gardner
Christopher Clark
...

Export BibTeX Citation

BERT

BERT

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

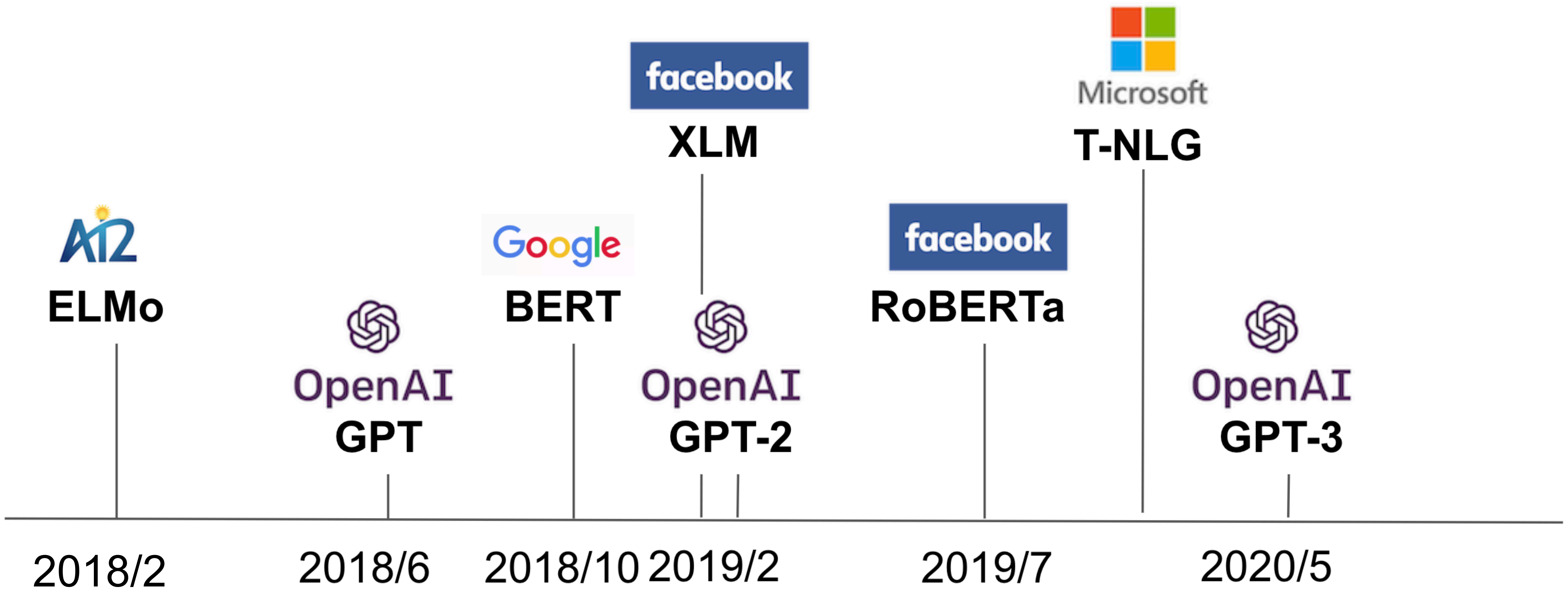
BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

We argue that current techniques restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models are unidirectional, and this limits the choice of architectures that can be used during pre-training. For example, in OpenAI GPT, the authors use a left-to-right architecture, where every token can only attend to previous tokens in the self-attention layers

Context-dependent Embeddings

- AI2 released ELMo in 2017-2018, GPT was released in summer 2018, BERT came out October 2018
- aka Pre-trained Language Models



and many more:

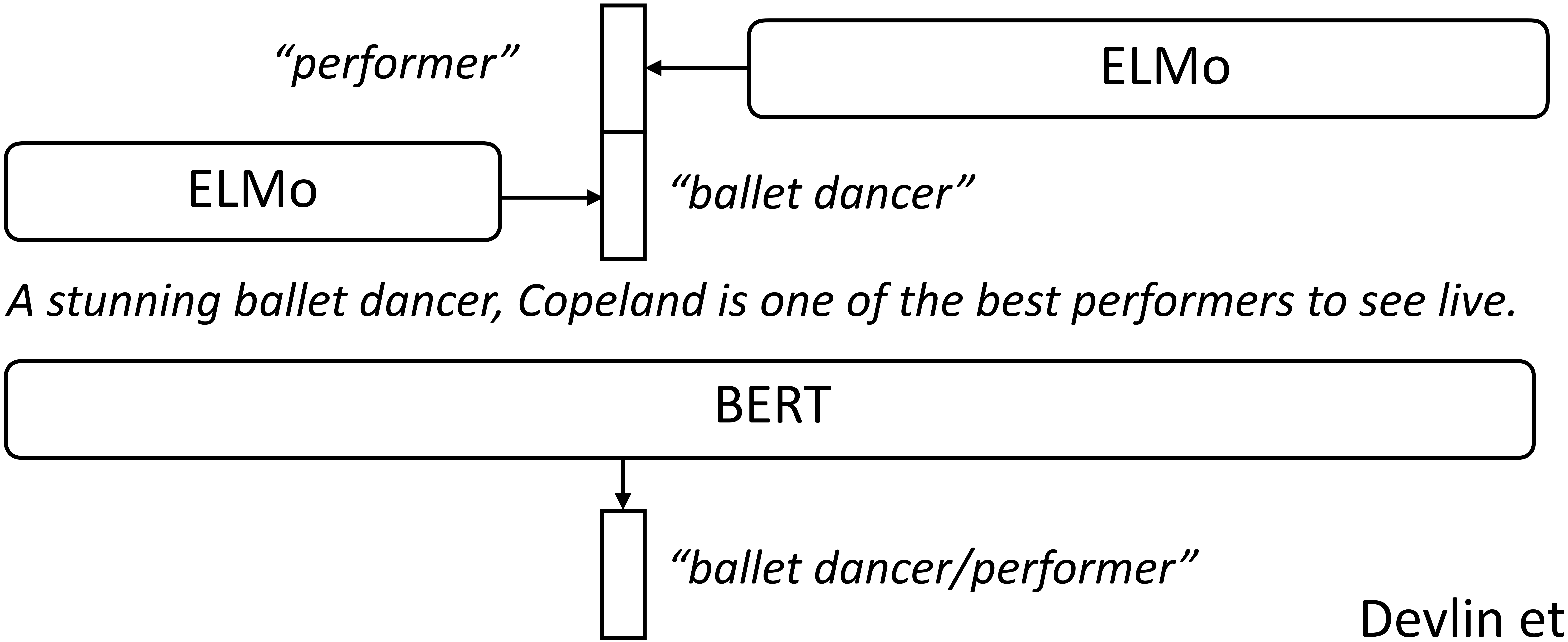
- BART
- DialoGPT
- GPT-J
- T5
- T0
- OPT
- PaLM
- ChatGPT
- Llama
- Mistral …

Contextual Word Embeddings

- AI2 released ELMo in spring 2018, GPT (transformer-based) was released in summer 2018, BERT came out October 2018
- BERT's four major changes compared to ELMo:
  - Transformers instead of LSTMs (transformers in GPT as well)
  - "Truely" Bidirectional <=> Masked LM objective instead of standard LM
  - Fine-tune instead of freeze at test time
  - Uses word pieces (subword tokenization)

BERT

- ELMo is a unidirectional model: we can concatenate two unidirectional models, but is this the right thing to do?
- ELMo looks at each direction in isolation; BERT looks at them jointly

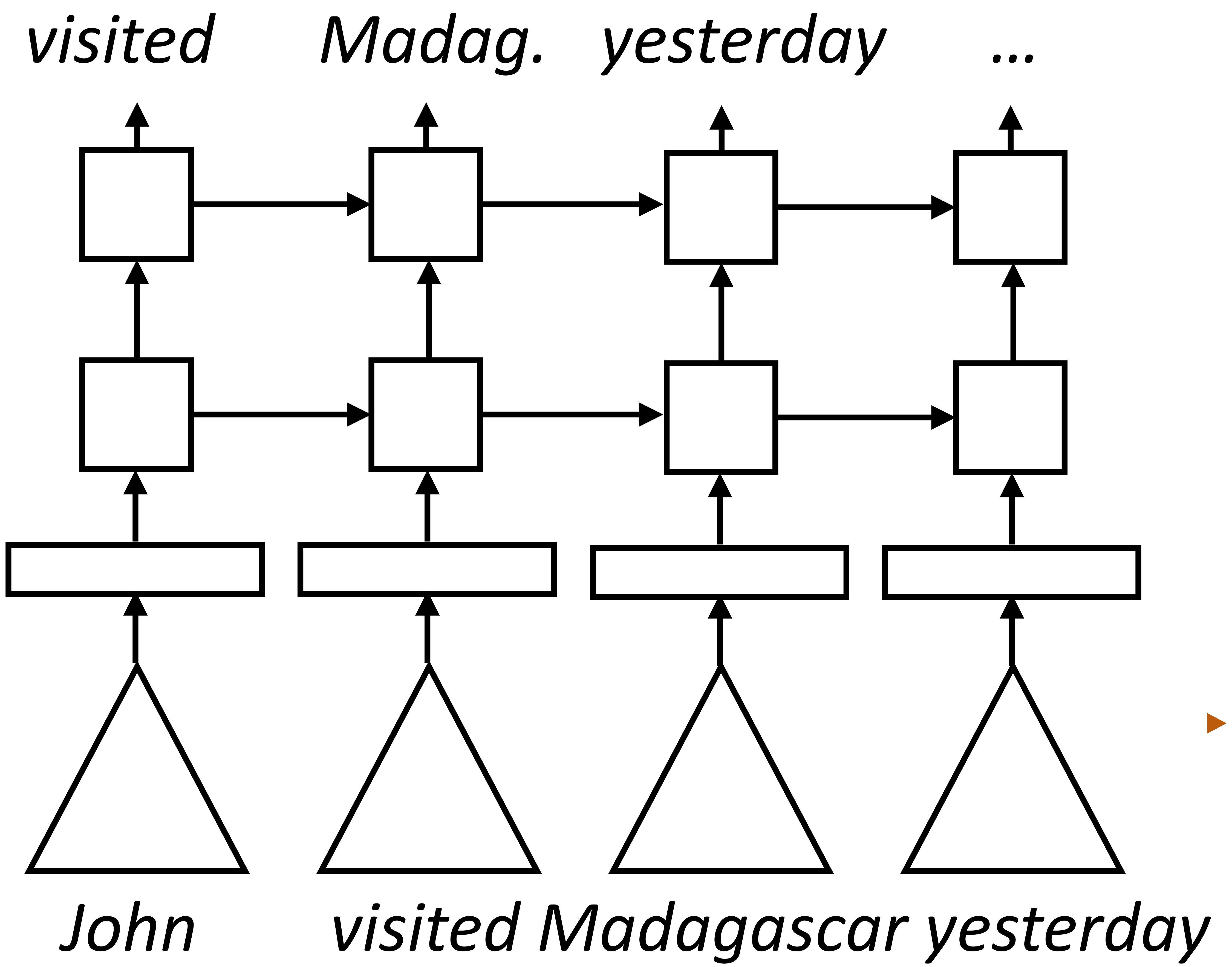


Devlin et al. (2019)

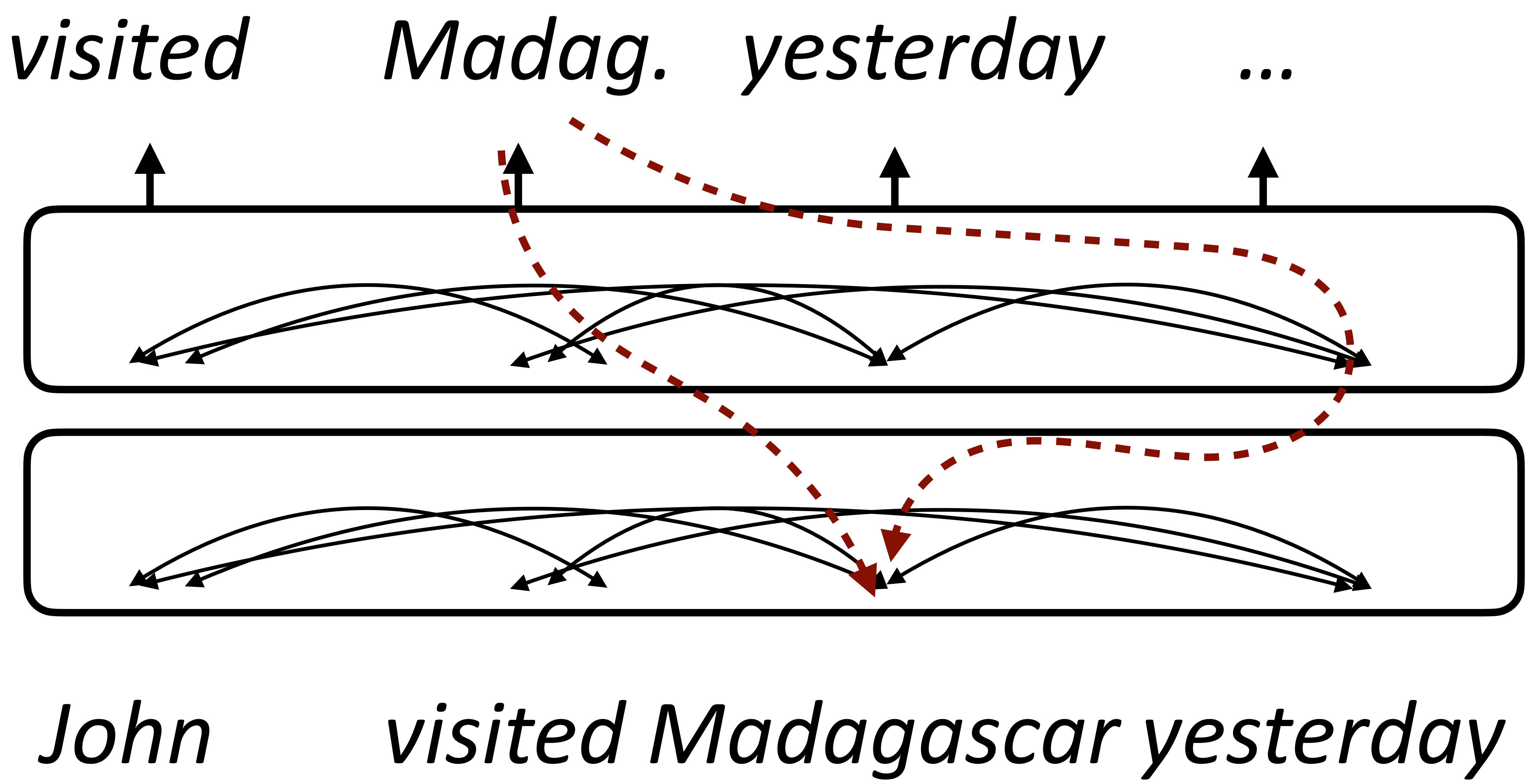
BERT

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

ELMo (Language Modeling)



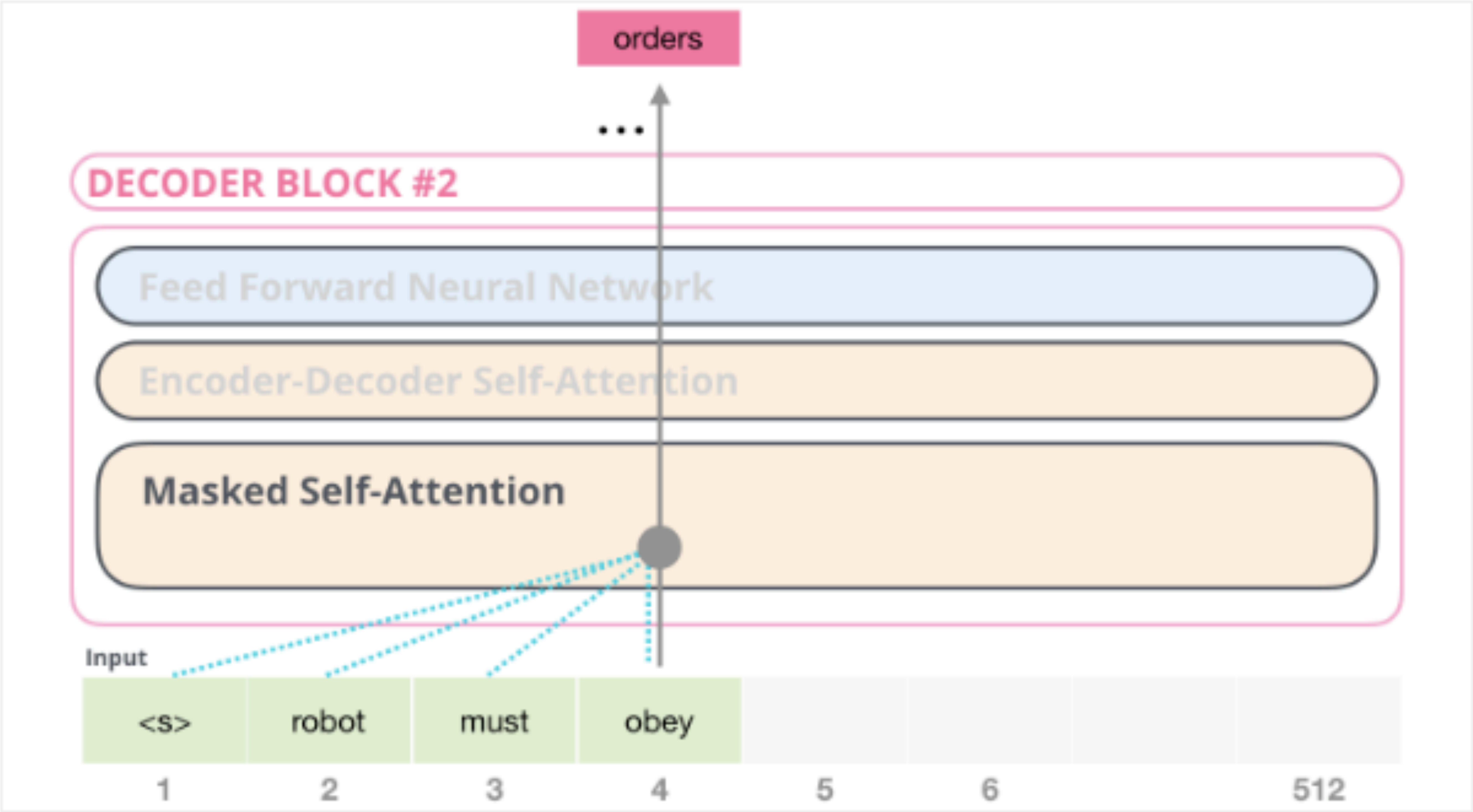
BERT



- Can do this by "one-sided" Transformer (masked self-attention), but "two-sided" Transformer encoder can cheat

GPT (preview)

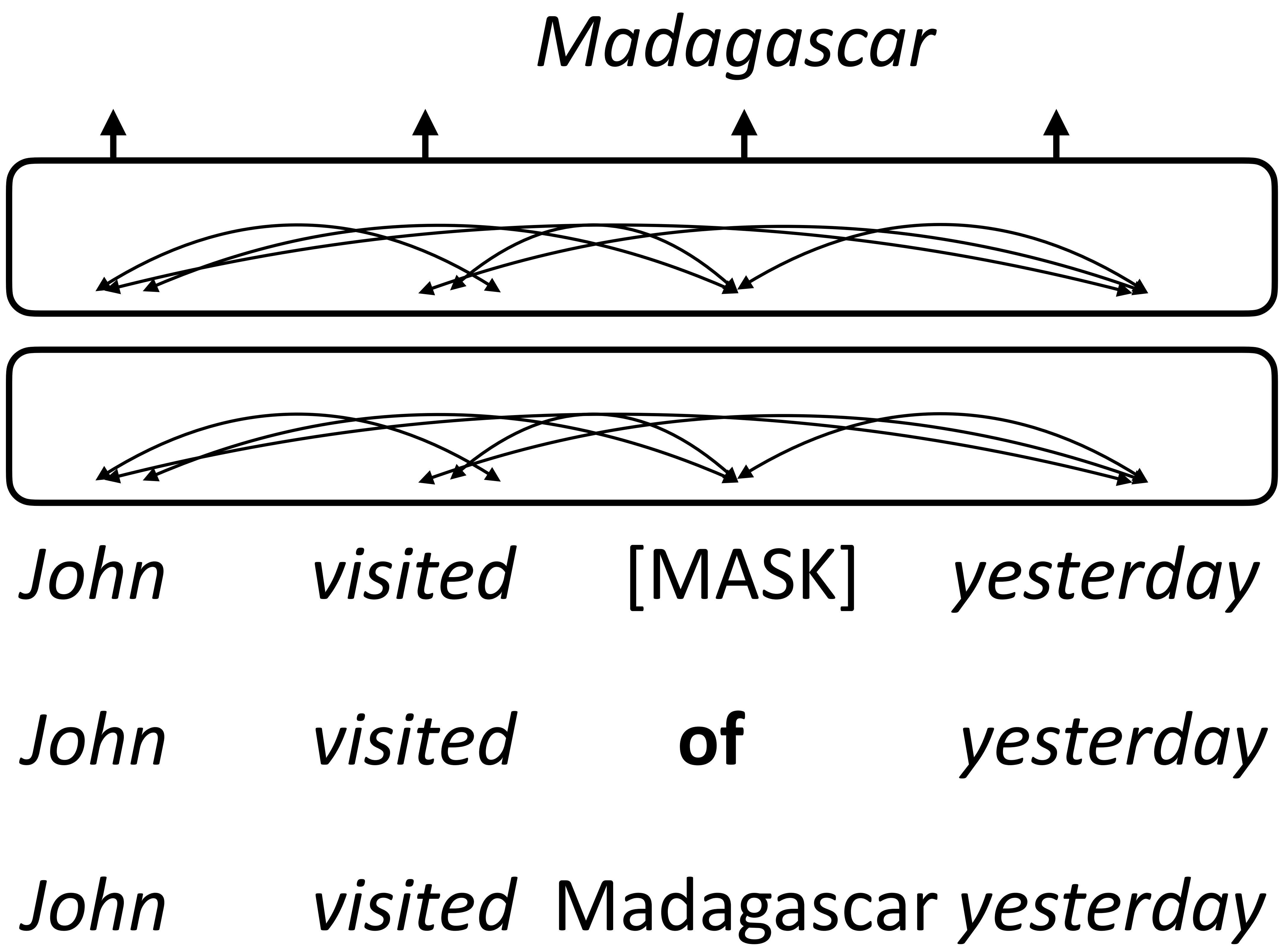
- Modified Transformer (masked self-attention): each token can only attend to past tokens



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, predict 15% of the tokens
  - For 80% (of the 15%), replace the input token with [MASK]
  - For 10%, replace w/random
  - For 10%, keep same



Devlin et al. (2019)

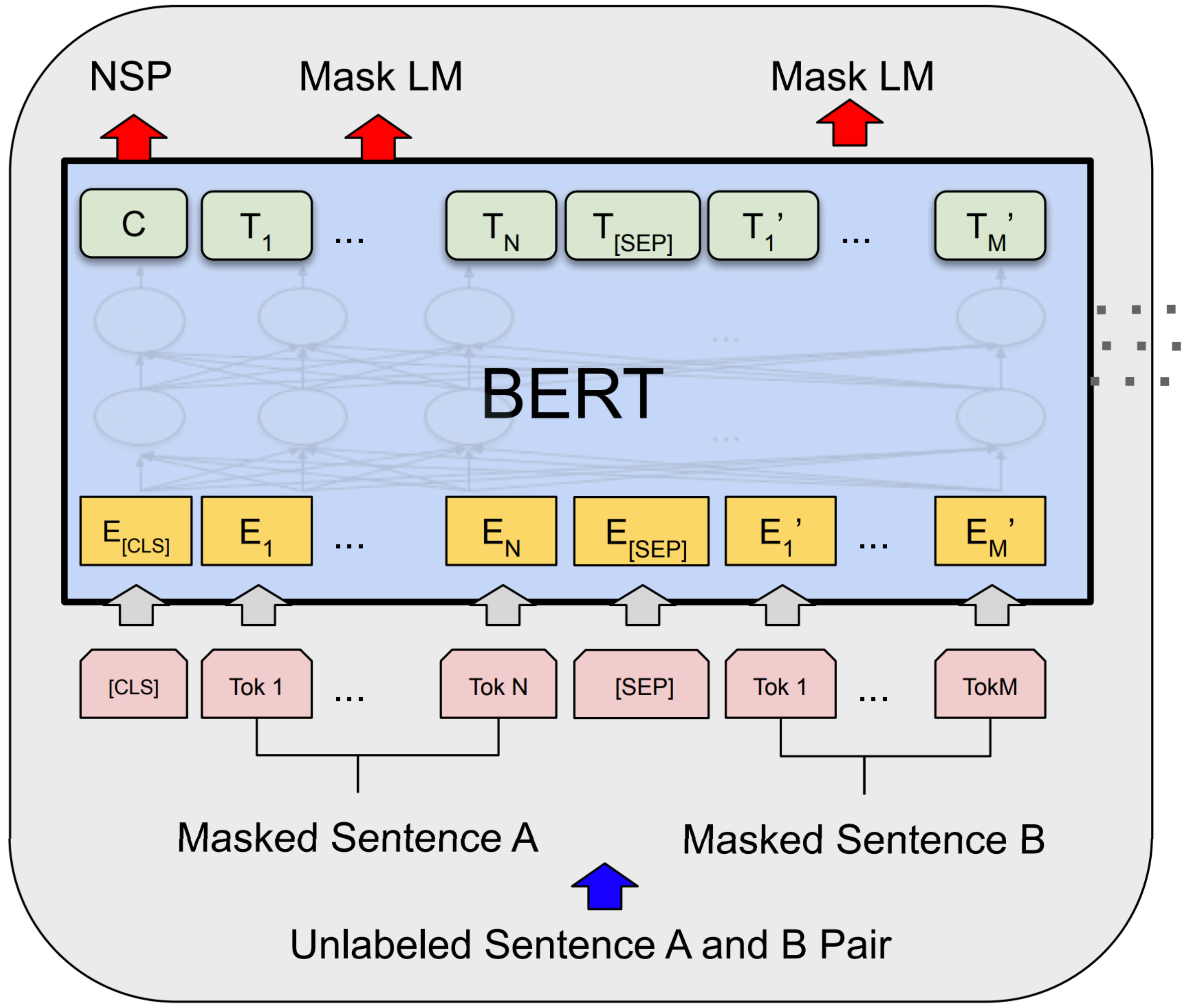
Next “Sentence” Prediction

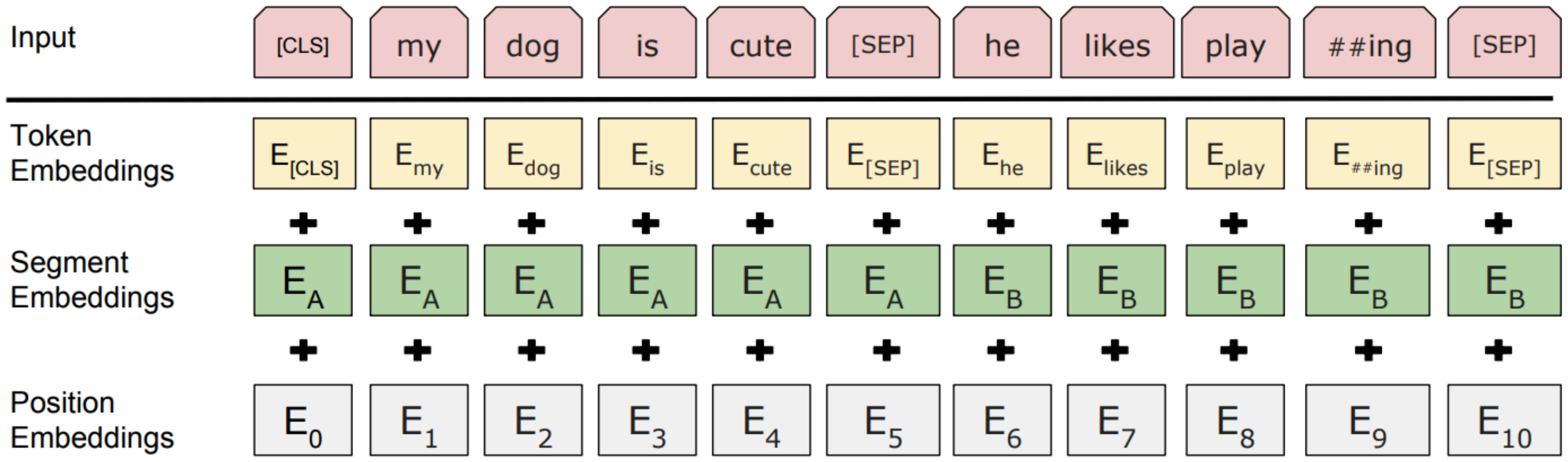
- Input: [CLS] Text chunk 1 [SEP] Text chunk 2
- 50% of the time, take the true next chunk of text, 50% of the time take a random other chunk. Predict whether the next chunk is the “true” next
- BERT objective: masked LM + next sentence prediction

Devlin et al. (2019)

BERT Architecture

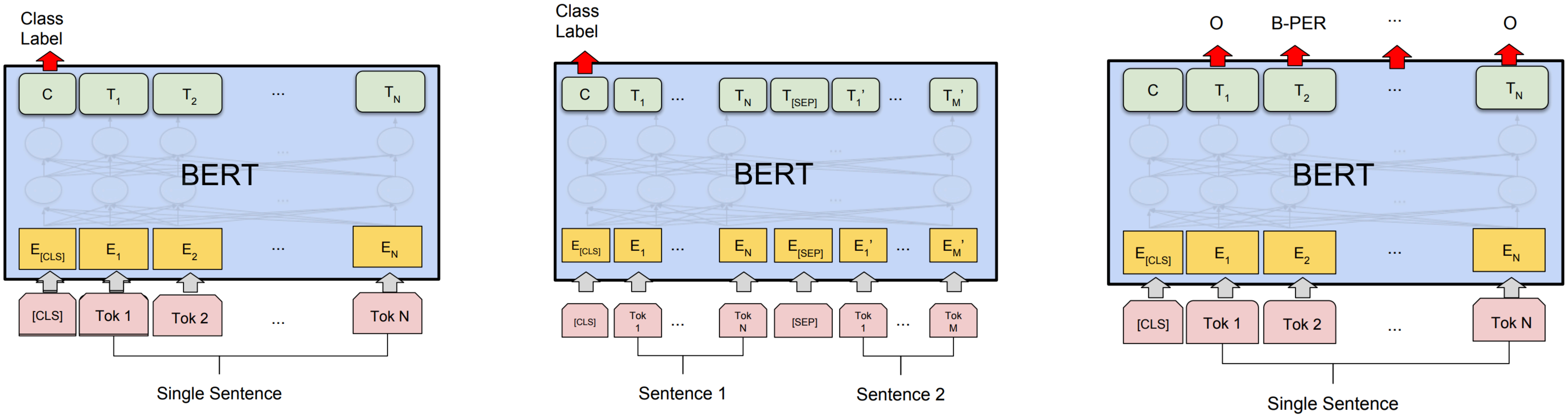
- BERT Base: 12 layers, 768-dim, 12 heads. Total params = 110M
- BERT Large: 24 layers, 1024-dim, 16 heads. Total params = 340M
- Positional embeddings and segment embeddings, 30k word pieces
- This is the model that gets **pre-trained** on a large corpus





Devlin et al. (2019)

What can BERT do?



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

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

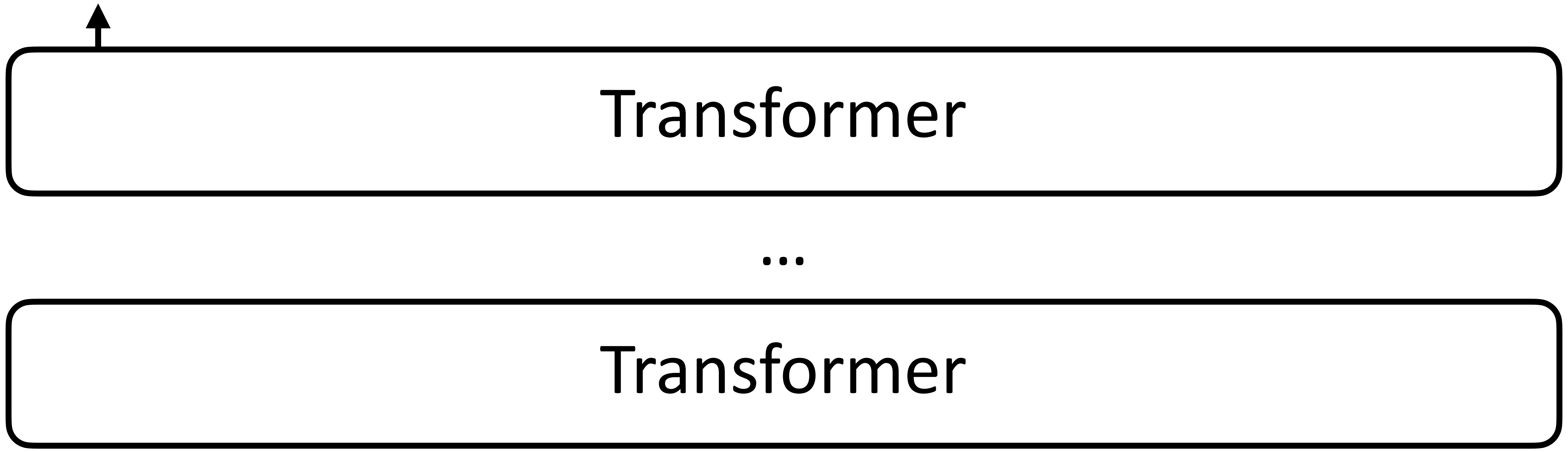
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

- CLS token is used to provide classification decisions
- Sentence pair tasks (entailment): feed both sentences into BERT
- BERT can also do tagging by predicting tags at each word piece

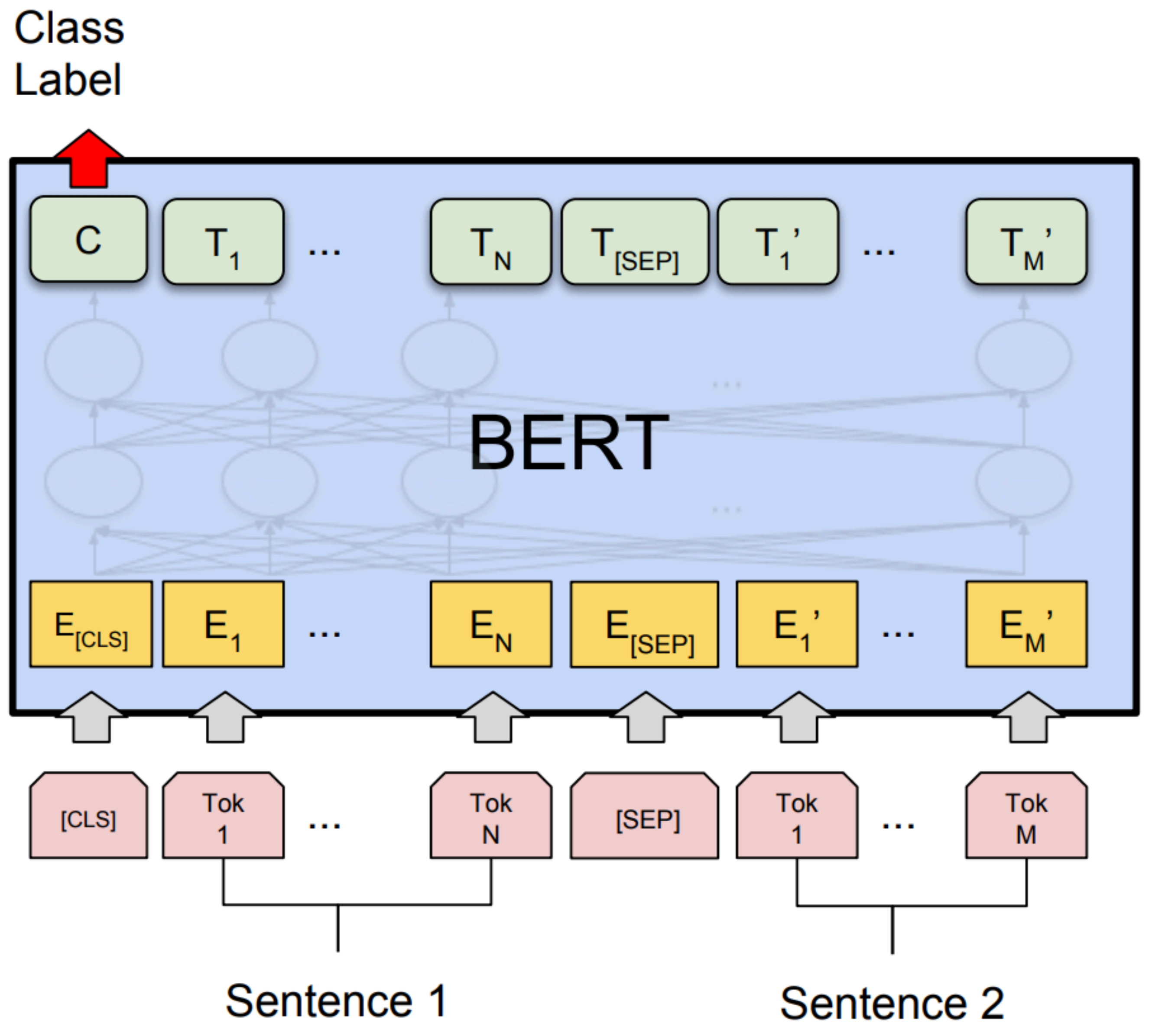
Devlin et al. (2019)

What can BERT do?

Entails (first sentence implies second one 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 this sentence pair stuff?
- Transformers can capture interactions between the two sentences, (even though the NSP objective doesn't really cause this to happen).

Devlin et al. (2019)

Recap: Natural Language Inference

Premise		Hypothesis
A boy plays in the snow	*entails*	A boy is outside
A man inspects the uniform of a figure	*contradicts*	The man is sleeping
An older and younger man smiling	*neutral*	Two men are smiling and laughing at cats playing

- Long history of this task: "Recognizing Textual Entailment" challenge in 2006 (Dagan, Glickman, Magnini)
- Early datasets: small (hundreds of pairs), very ambitious (lots of world knowledge, temporal reasoning, etc.)

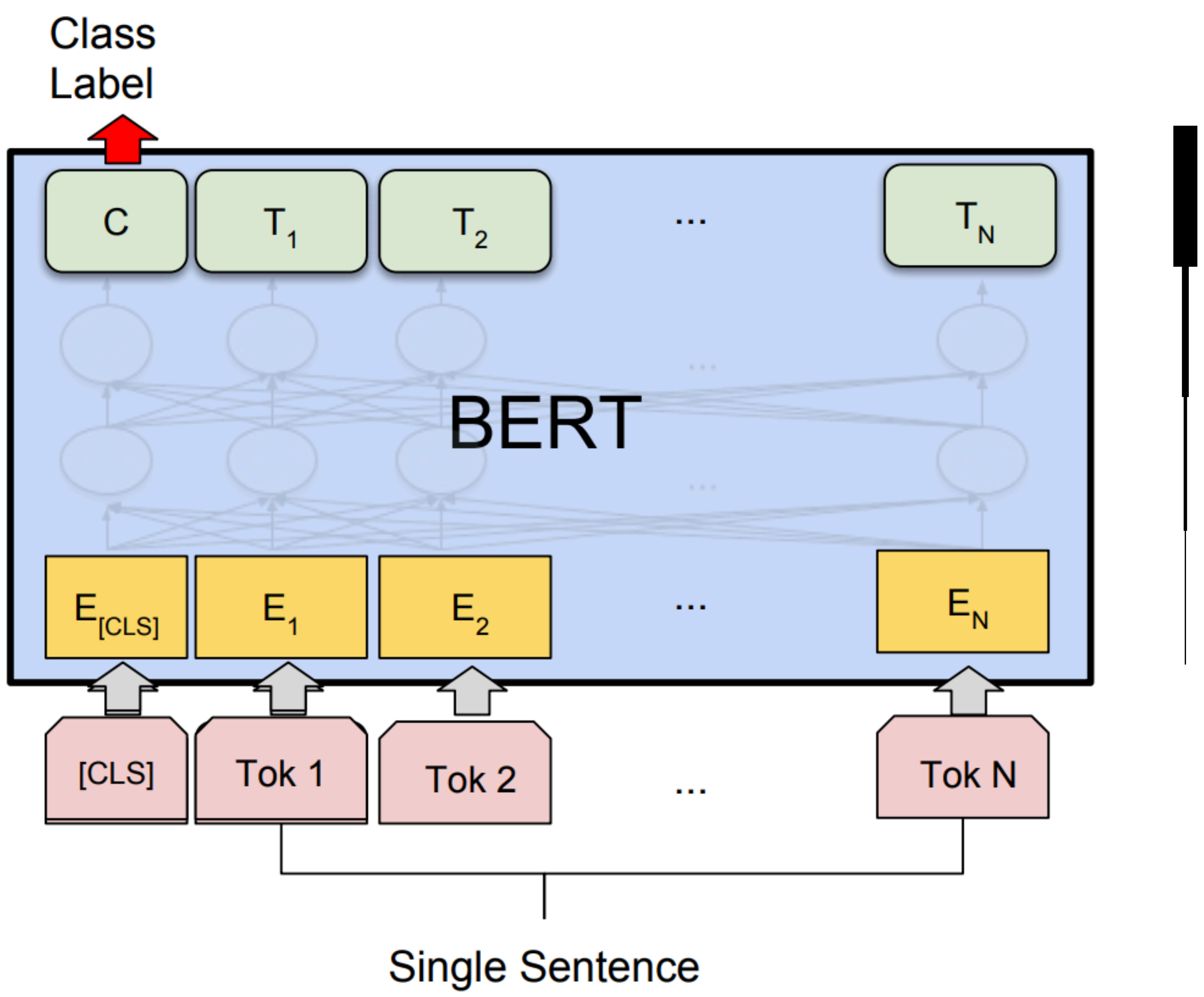
What can BERT NOT do?

- BERT **cannot** generate text (at least not in an obvious way)
  - Can fill in [MASK] tokens, but can't generate left-to-right (you can put [MASK] at the end, then predict repeatedly, but this is slow)
- Masked language models are intended to be used primarily for "analysis" tasks, e.g., sequential tagging, semantic similarity between two sentences, …

BERT Results, Extensions

Fine-tuning BERT

- Fine-tune for 1-3 epochs, small learning rate (e.g. 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
- More complex “triangular learning rate” schemes exist

Fine-tuning BERT

- How does frozen (❄️) vs. fine-tuned (🔥) compare?

Pretraining	Adaptation	NER CoNLL 2003	SA SST-2	Nat. lang. inference MNLI	 SICK-E	Semantic textual similarity SICK-R	 MRPC	 STS-B
Skip-thoughts	❄️	-	81.8	62.9	-	86.6	75.8	71.8
ELMo	❄️	91.7	**91.8**	**79.6**	**86.3**	**86.1**	**76.0**	**75.9**
	🔥	**91.9**	91.2	76.4	83.3	83.3	74.7	75.5
	Δ=🔥-❄️	0.2	-0.6	-3.2	-3.3	-2.8	-1.3	-0.4
BERT-base	❄️	92.2	93.0	**84.6**	84.8	86.4	78.1	82.9
	🔥	**92.4**	**93.5**	**84.6**	**85.8**	**88.7**	**84.8**	**87.1**
	Δ=🔥-❄️	0.2	0.5	0.0	1.0	2.3	6.7	4.2

- BERT is typically better if the whole network is fine-tuned, unlike ELMo

Peters, Ruder, Smith (2019)

Evaluation: GLUE

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

Wang et al. (2019)

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)

Subsequent Improvements to BERT

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

epoch 2

epoch 1

… John visited Madagascar yesterday …

- Whole word masking: don't mask out parts of words (word pieces)

… _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

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

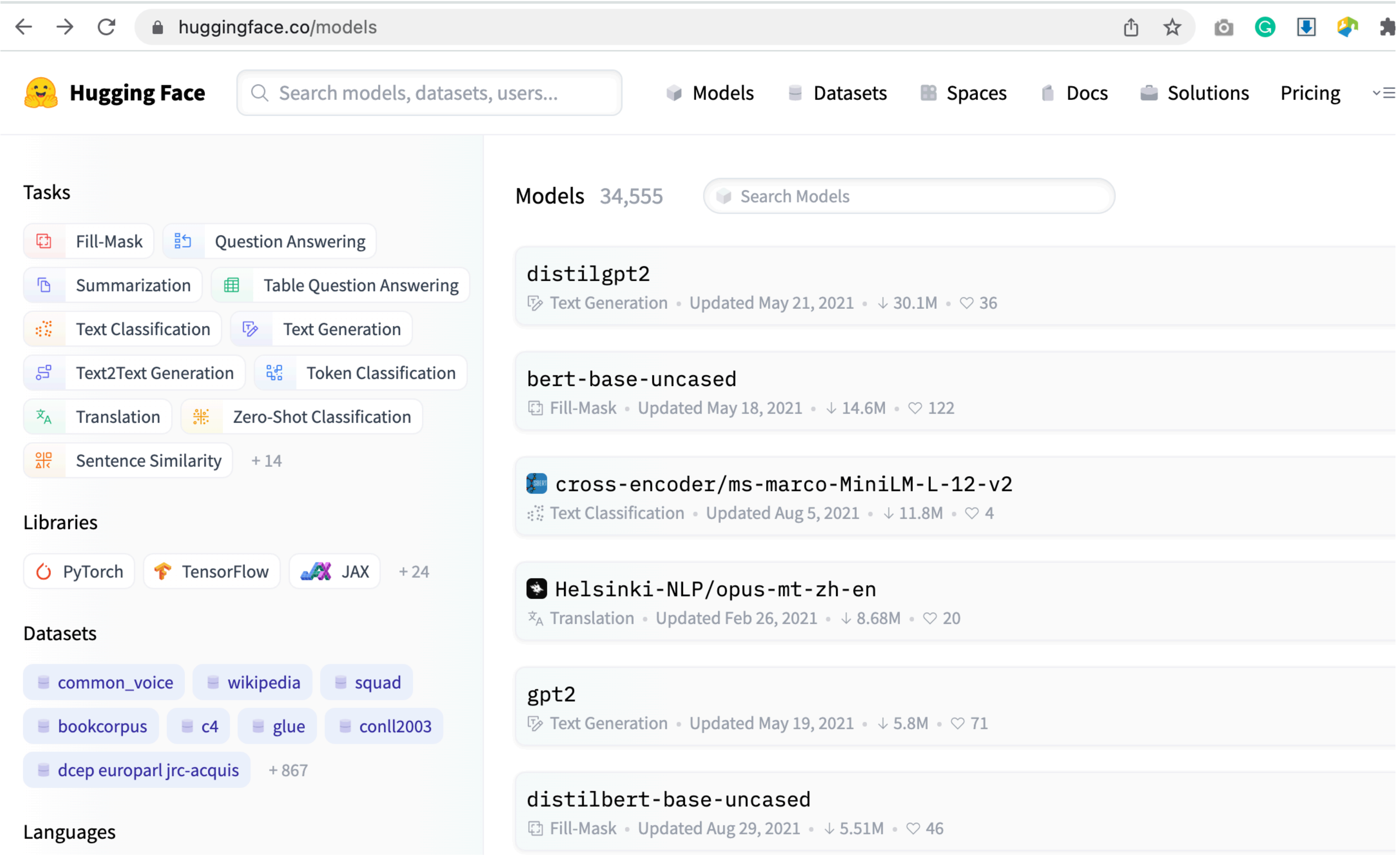
- New training + more data = better performance

- For this and more: check out Huggingface or fairseq

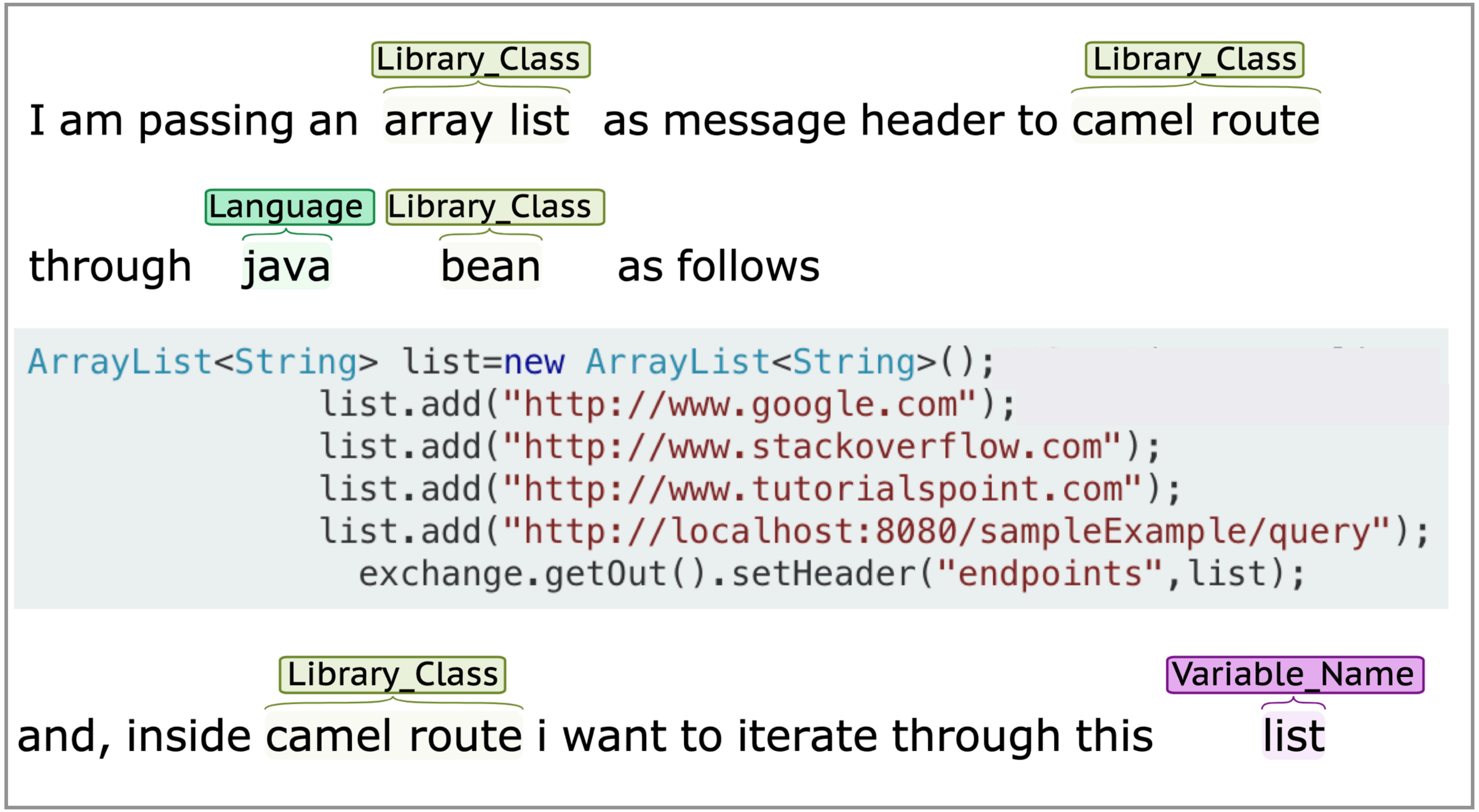
Liu et al. (2019)

many BERT variations

- For specific text domains (e.g. StackOverflow), or specific languages



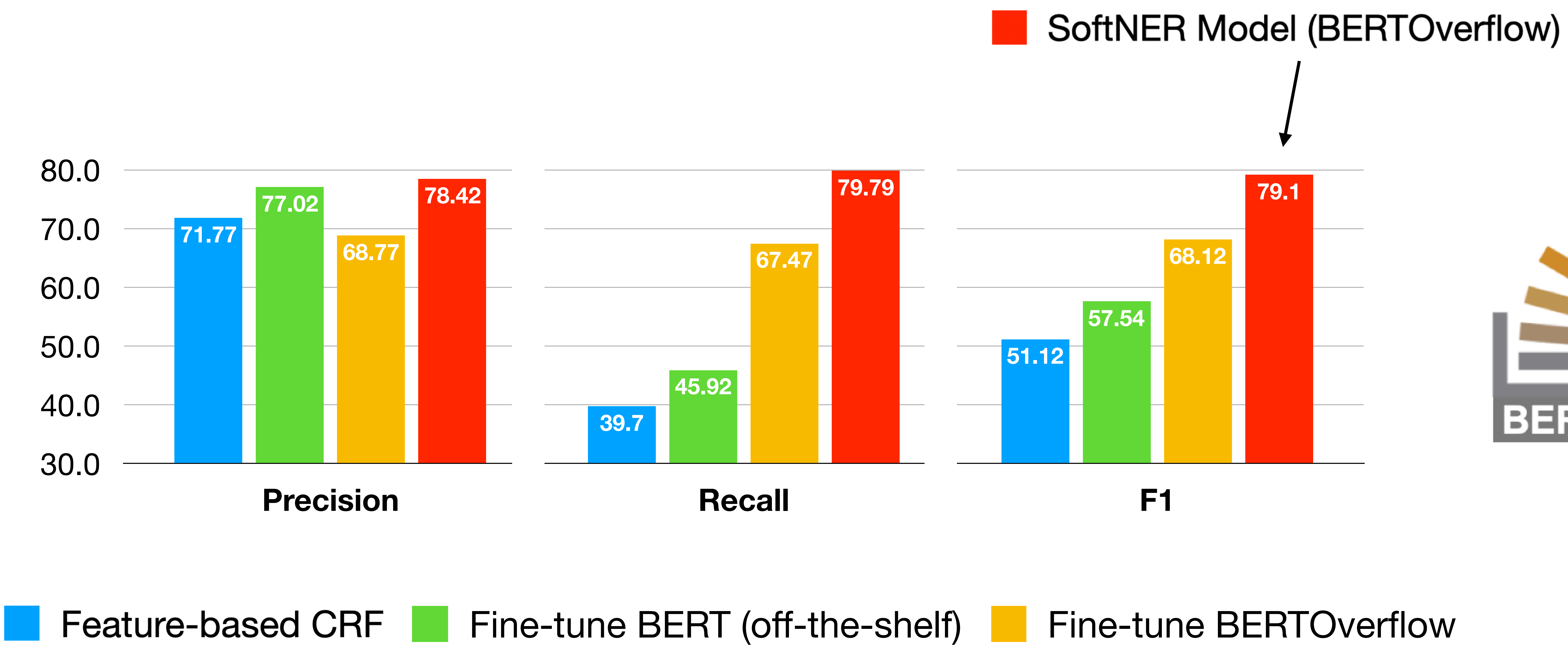
NER in StackOverflow



Jeniya Tabassum, Mounica Maddela, Alan Ritter, Wei Xu. “Code and Named Entity Recognition in StackOverflow” (ACL 2020)

NER in StackOverflow

- A domain-specific BERT model that pre-trained on 10-year StackOverflow data (152M sentences; ~2B tokens).



Jeniya Tabassum, Mounica Maddela, Alan Ritter, Wei Xu. "Code and Named Entity Recognition in StackOverflow" (ACL 2020)

Bilingual BERT

- A customized bilingual BERT for Arabic NLP and English-to-Arabic zero-shot transfer learning

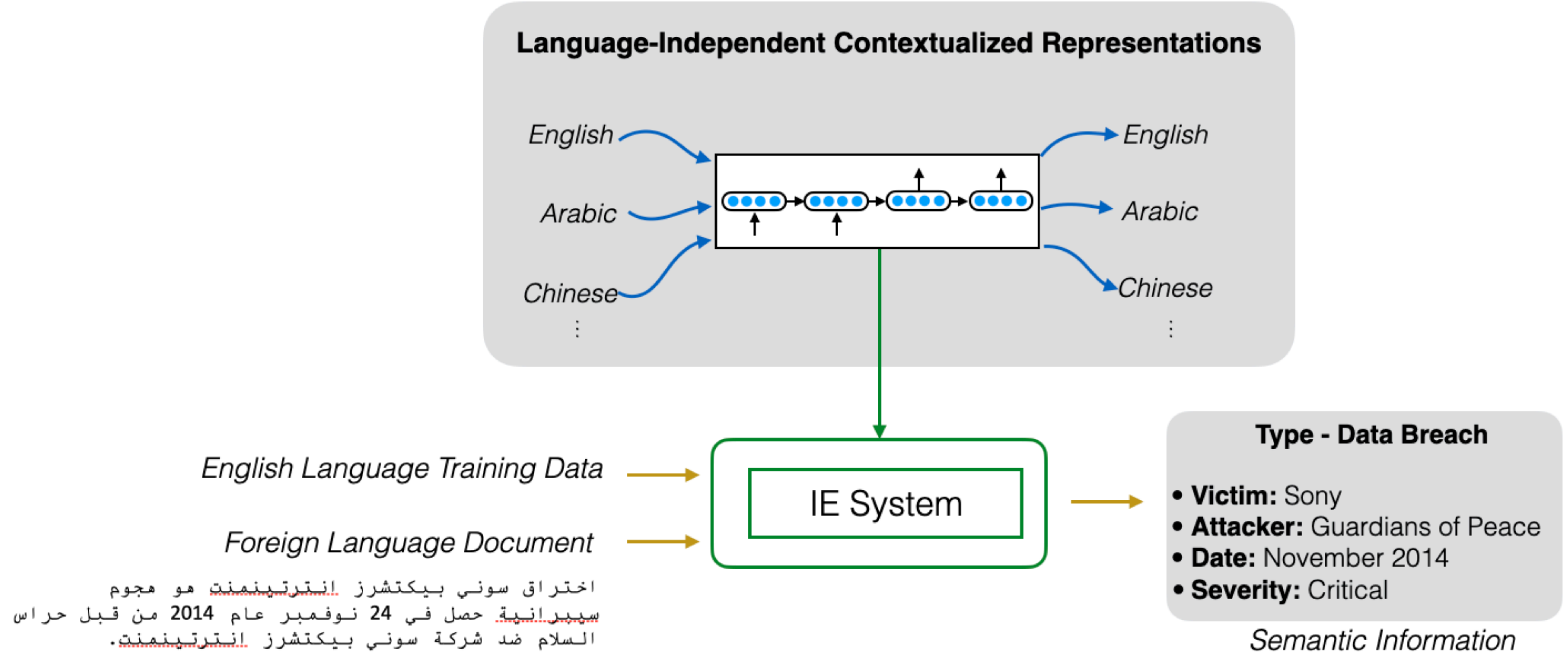
	Data Source	**Data Size** (All / English / Arabic)	**IE Performance** (F1 score)
AraBERT (AUBeirut 2019)	News	2.5B / 0B / 2.5B	97.1 / —
mBERT (Google 2018)	Wiki	21.9B / 2.5B / 0.15B	75.3 / 30.1
XLM-RoBERTa (Facebook 2019)	Common Crawl	295B / 55.6B / 2.9B	79.2 / 40.4
GigaBERT (our work)	News, Wiki, Common Crawl	10.4B / 6.1B / 4.3B	84.3 / 48.2

supervised learning   zero-shot transfer learning

Lan et al. (2020)

Bilingual BERT

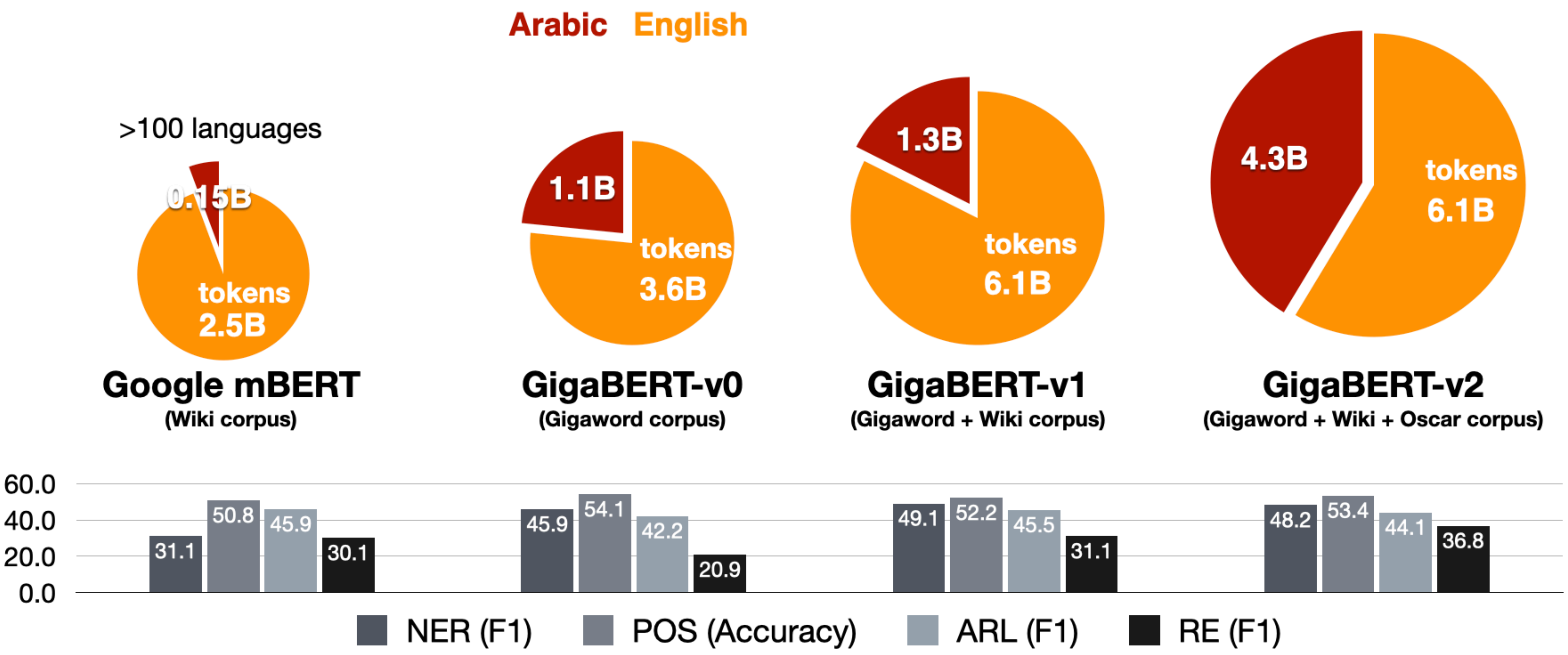
- A customized bilingual BERT for Arabic NLP and English-to-Arabic zero-shot transfer learning
- i.e., Information extraction models, trained on annotated English data, directly apply to non-English texts to extract entities and events.



Lan et al. (2020)

Bilingual BERT

- A customized bilingual BERT for Arabic NLP and English-to-Arabic zero-shot transfer learning



Lan et al. (2020)

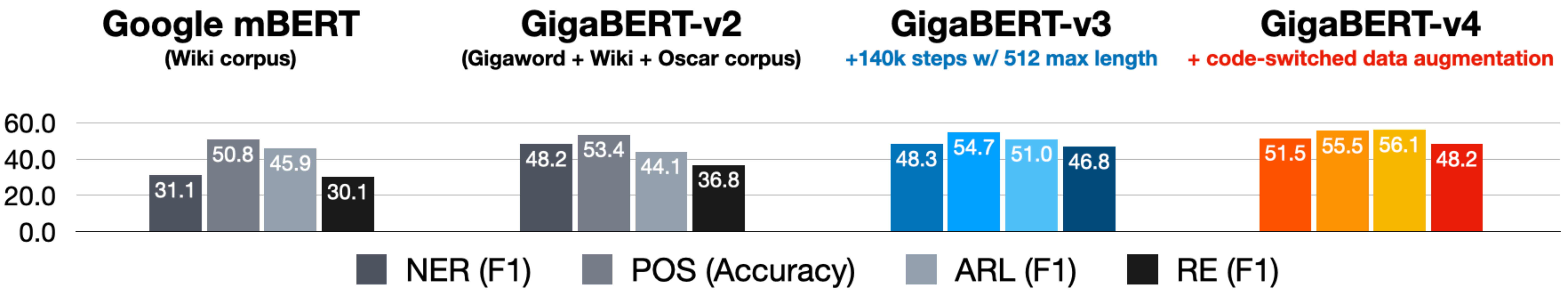
Bilingual BERT

- A customized bilingual BERT for Arabic NLP and English-to-Arabic zero-shot transfer learning

(English Translation — Official: China reaches the five-year plan target of cutting emissions)

مسؤول: **الصين** تبلغ هدف **الخطة** الخمسية المتعلق بخفض **الانبعاثات**

مسؤول: China تبلغ هدف plan الخمسية المتعلق بخفض emissions



Lan et al. (2020)

GigaBERT

- A customized bilingual BERT for Arabic NLP and English-to-Arabic zero-shot transfer learning

	Data Source	Data Size (All / English / Arabic)	IE Performance (F1 score)
AraBERT (AUBeirut 2019)	News	2.5B / 0B / 2.5B	97.1 / —
mBERT (Google 2018)	Wiki	21.9B / 2.5B / 0.15B	75.3 / 30.1
XLM-RoBERTa (Facebook 2019)	Common Crawl	295B / 55.6B / 2.9B	79.2 / 40.4
GigaBERT (our work)	News, Wiki, Common Crawl	10.4B / 6.1B / 4.3B	84.3 / 48.2

supervised learning   zero-shot transfer learning

Lan et al. (2020)

BERT/MLMs

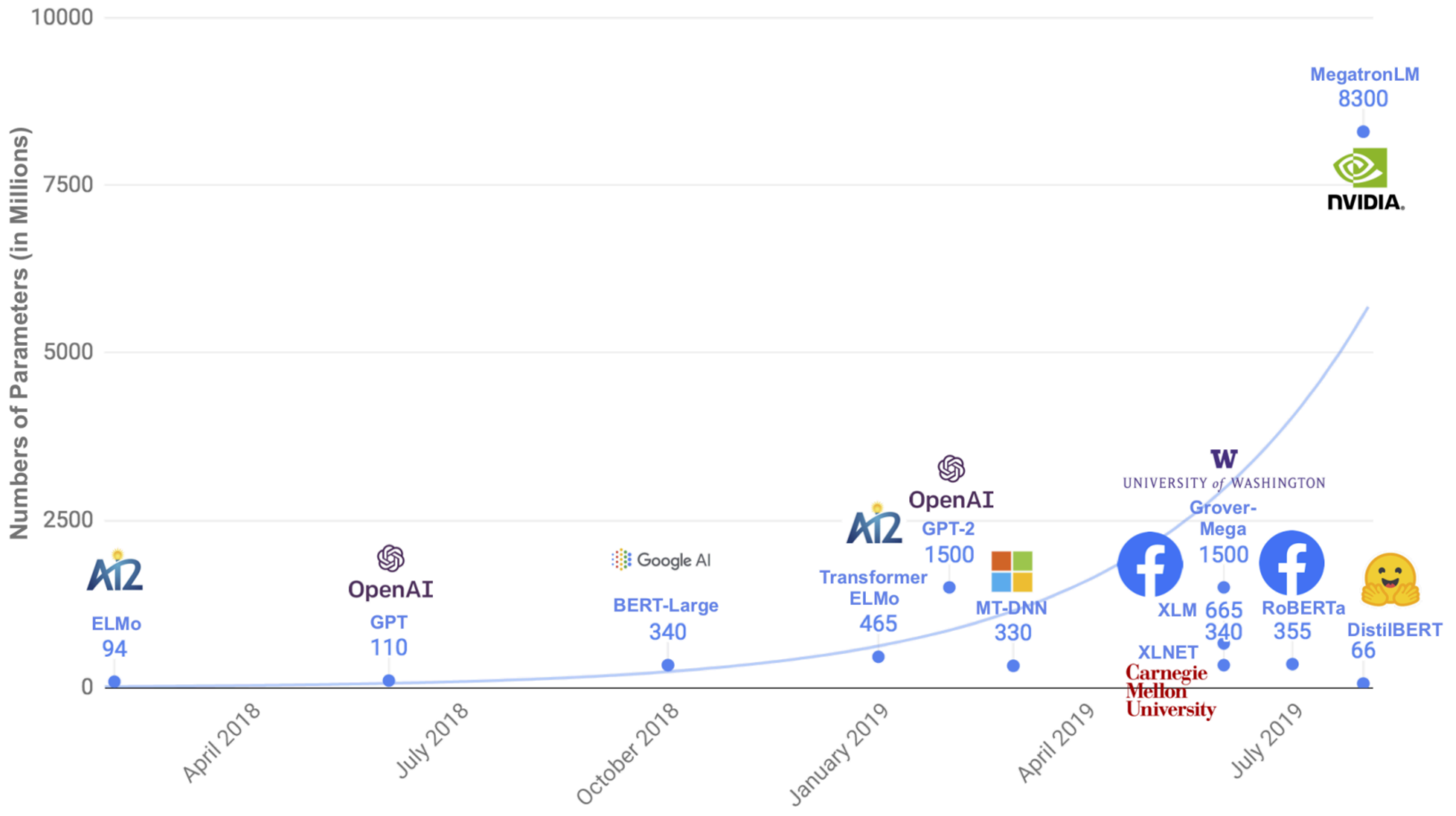
- There are lots of ways to train these models!
- Key factors:
  - Big enough model
  - Big enough data
  - Well-designed “self-supervised” objective (something like language modeling). Needs to be a hard enough problem!

Compressing BERT

DistilBERT

- Remove 60+% of BERT's heads post-training with minimal drop in performance

- DistilBERT (Sanh et al., 2019): nearly as good with half the parameters of BERT (via knowledge distillation)



Michel et al. (2019)

ALBERT

- A Lite BERT (18x fewer parameters, 1.7x faster training than BERT)
- Factorized embedding matrix to save parameters, model context-independent words with fewer parameters

  Ordinarily |V| x H — |V| is 30k-90k, H is >1000

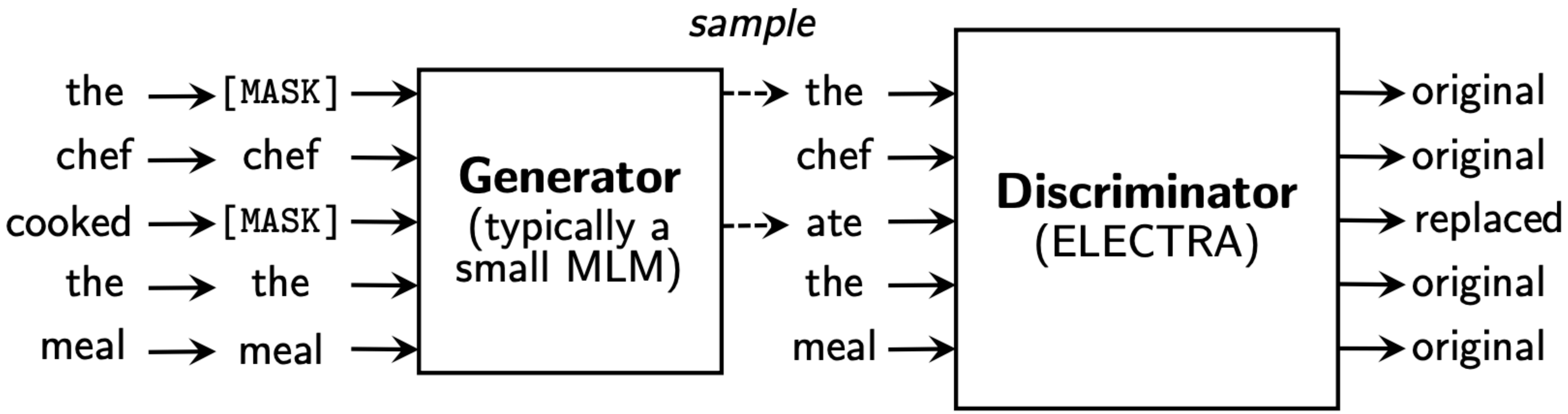
  Factor into two matrices with a low-rank approximation

  Now: |V| x E and E x H — E is 128 in their implementation

- Additional cross-layer parameter sharing

Lan et al. (2020)

ELECTRA



- No need to necessarily have a generative model (predicting words)
- This objective is more computationally efficient (trains faster) than the standard BERT objective

Clark et al. (2020)

Takeaways

- BERT-based systems are state-of-the-art for nearly every major text analysis task
- Transformers + lots of data + self-supervision seems to do very well
- Lots of work studying and analyzing these, but few “deep” conclusions have emerged
- Next time: BART/T5, GPT/GPT-2/GPT-3, etc.

Why is language modeling a good objective?

- "Impossible" problem but bigger models seem to do better and better at distributional modeling (no upper limit yet)
- Successfully predicting next words requires modeling lots of different effects in text

Context: My wife refused to allow me to come to Hong Kong when the plague was at its height and –" "Your wife, Johanne? You are married at last ?" Johanne grinned. "Well, when a man gets to my age, he starts to need a few home comforts.

Target sentence: After my dear mother passed away ten years ago now, I became _____.

Target word: lonely

- LAMBADA dataset (Papernot et al., 2016): explicitly targets world knowledge and very challenging LM examples

Recall: Winograd Schema

- Hector Levesque (2011): “Winograd schema challenge” (named after Terry Winograd, the creator of SHRDLU 1968-1972)

they advocated

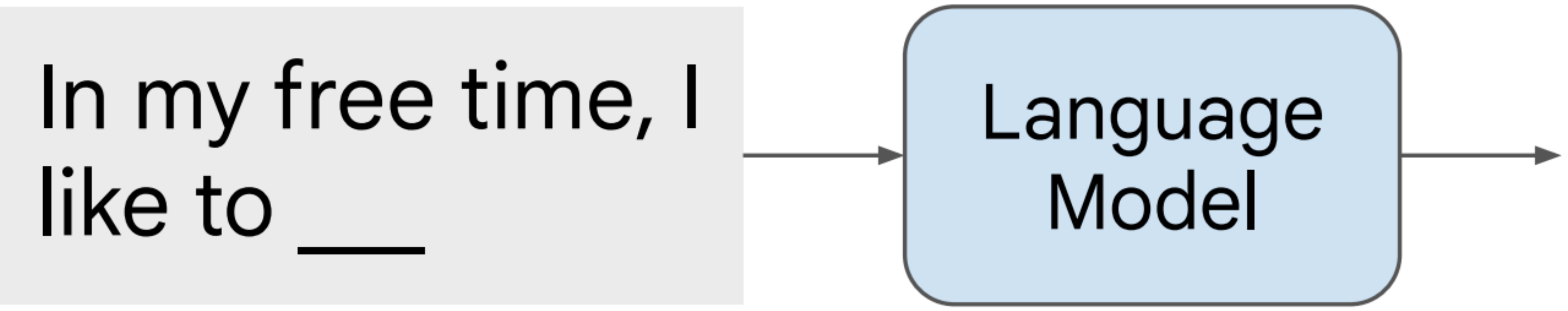
The city council refused the demonstrators a permit because they _______ violence

they feared

- This is so complicated that it’s an AI challenge problem! (AI-complete)
- Referential/semantic ambiguity

What do language model learn?

- Grammar



Word	**Probability**
a	
…	
banana	*0.00001*
…	
run	*0.7*
…	
zucchini	

The next word is probably not a noun

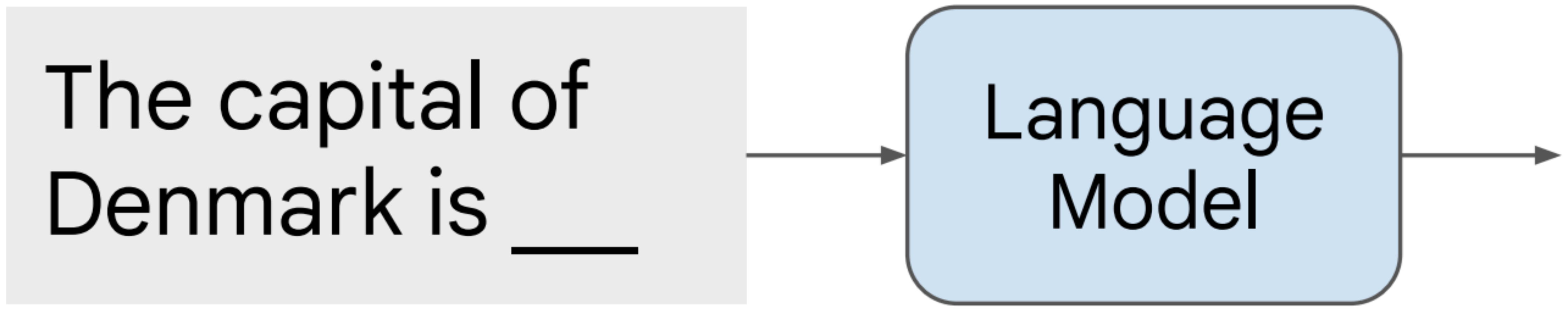
The next word is probably a verb

(hypothetical)

Image from Jason Wei

What do language model learn?

- Facts about the world



Word	Probability
a	
…	
Copenhagen	*0.9*
…	
London	*0.05*
…	
zucchini	

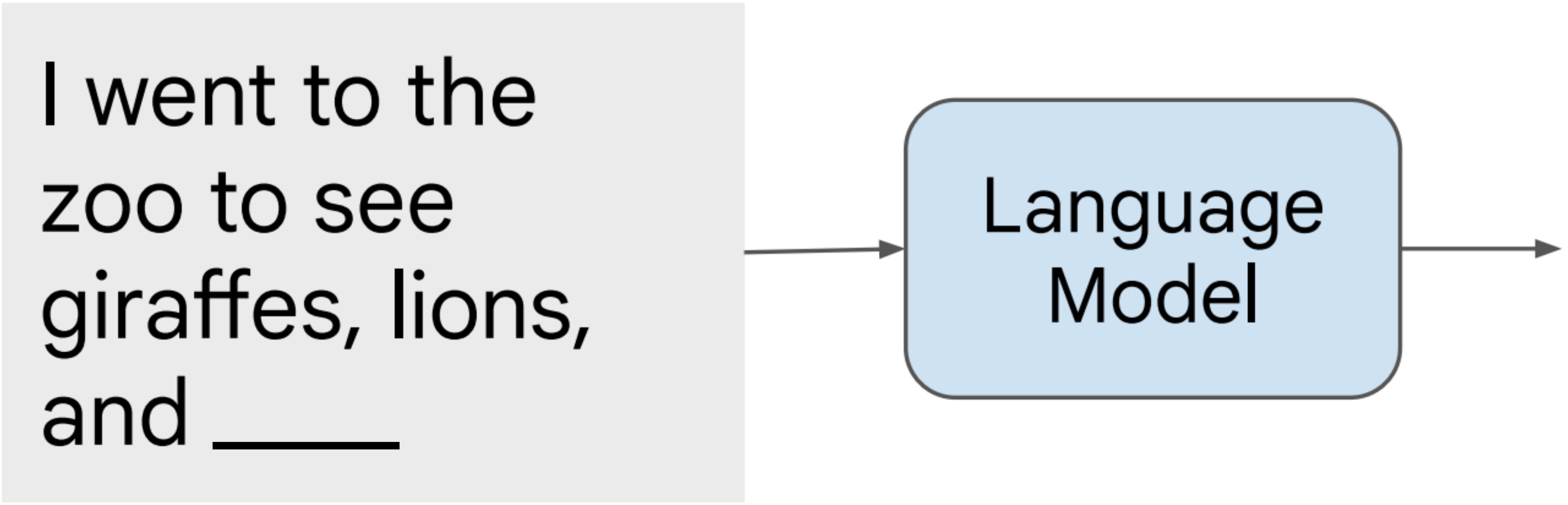
Associations between words!

(hypothetical)

Image from Jason Wei

What do language model learn?

- Lexical semantics



Word	Probability
a	
…	
spoon	*0.00001*
…	
zebras	*0.6*
…	
zucchini	

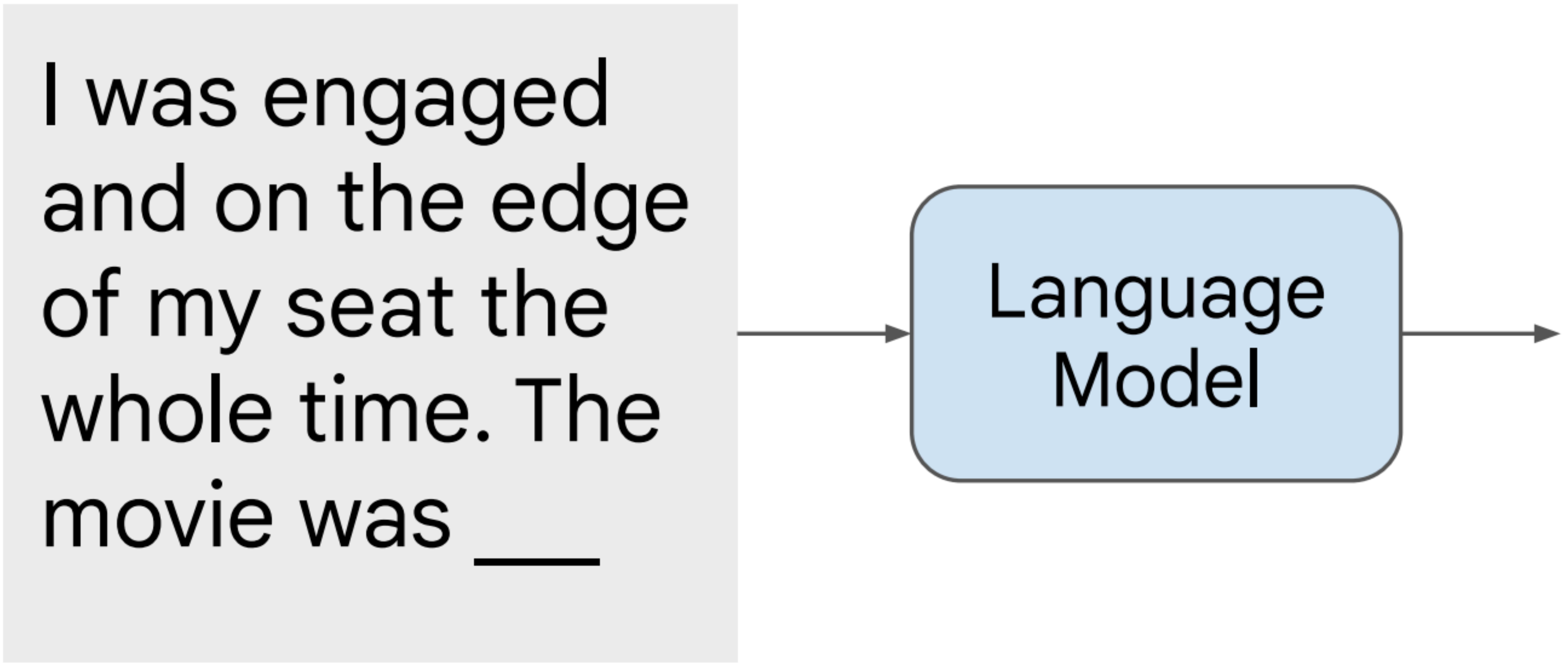
The next word is probably related to giraffes and lions

(hypothetical)

Image from Jason Wei

What do language model learn?

- Sentiment analysis



Word	**Probability**
a	
…	
bad	*0.1*
…	
good	*0.9*
…	
zucchini	

Well, “engaged” is pretty indicative of a positive sentiment

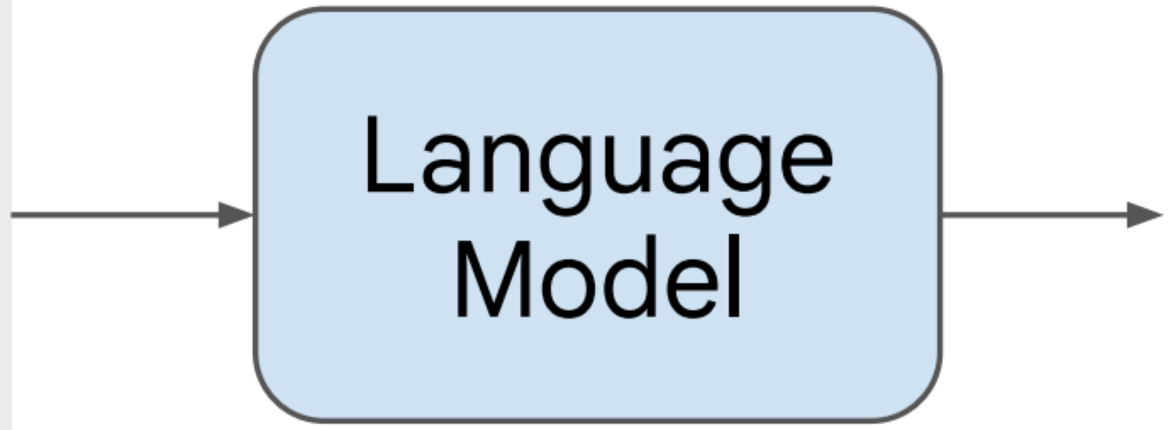
(hypothetical)

Image from Jason Wei

What do language model learn?

- Harder sentiment analysis

Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was ___



Word	Probability
a	
...	
bad	0.7
...	
good	0.3
...	
zucchini	

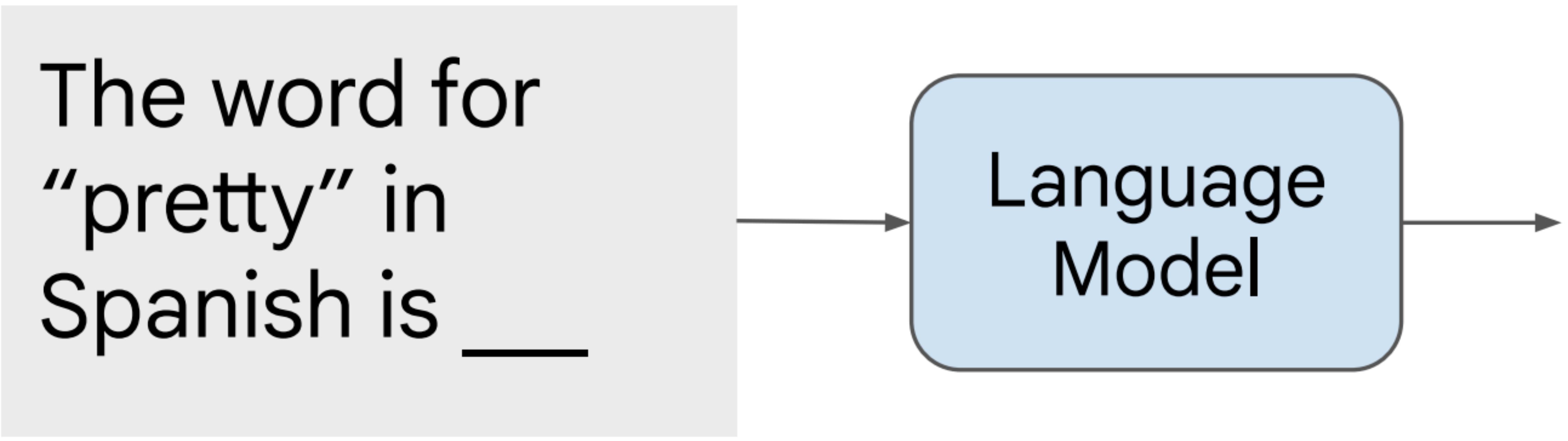
Some more-complex understanding needed

(hypothetical)

Image from Jason Wei

What do language model learn?

- Translation



Word	Probability
a	
...	
bonita	*0.8*
...	
hola	*0.03*
...	
zucchini	

Understanding of multiple languages

(hypothetical)

Image from Jason Wei

What do language model learn?

- Spatial reasoning

Iroh went into the kitchen to make some tea. Standing next to Iroh, Zuko pondered his destiny. Zuko left the ____

→ Language Model →

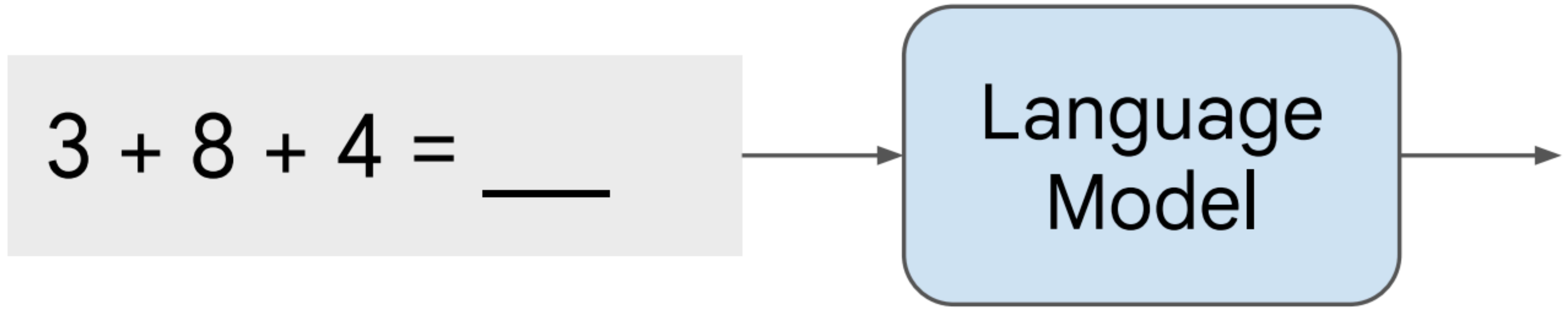
Word	**Probability**
a	
...	
...	
kitchen	*0.8*
...	
zucchini	

(hypothetical)

Image from Jason Wei

What do language model learn?

- Easy arithmetic



Word	**Probability**
a	
...	
14	*0.1*
15	*0.7*
...	
zucchini	

(hypothetical)

Understanding (or memorization) or addition?

Image from Jason Wei