Pretraining Language Models (part 1)

Wei Xu

(many slides from Greg Durrett)

This Lecture

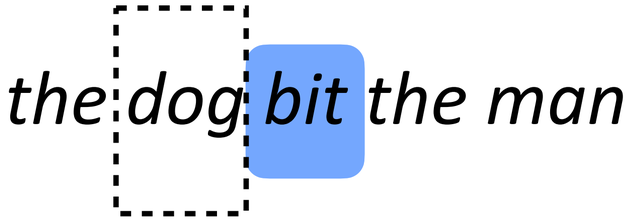
- ELMo
- BERT
- BERT Results, Extensions
- Analysis/Visualization of BERT

Readings

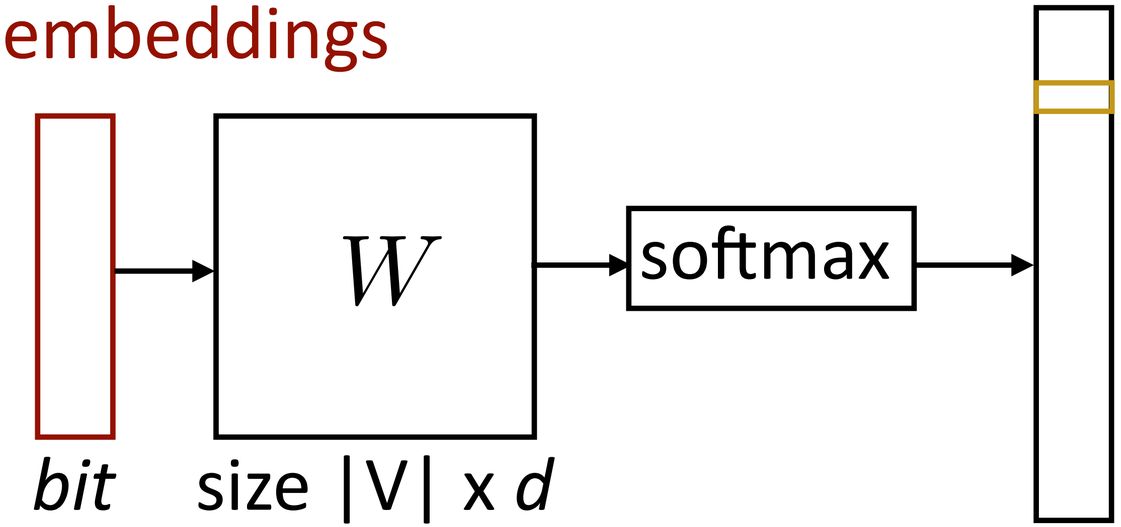
- Readings —
  - J+M 10
  - ELMo by Peters et al.
    https://aclanthology.org/N18-1202.pdf
  - BERT by Devlin et al.
    https://aclanthology.org/N19-1423.pdf

Recall: word2vec (Skip-Gram)

- Predict one word of context from word



d-dimensional
word embeddings



gold label = *dog*

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

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

Mikolov et al. (2013)

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.

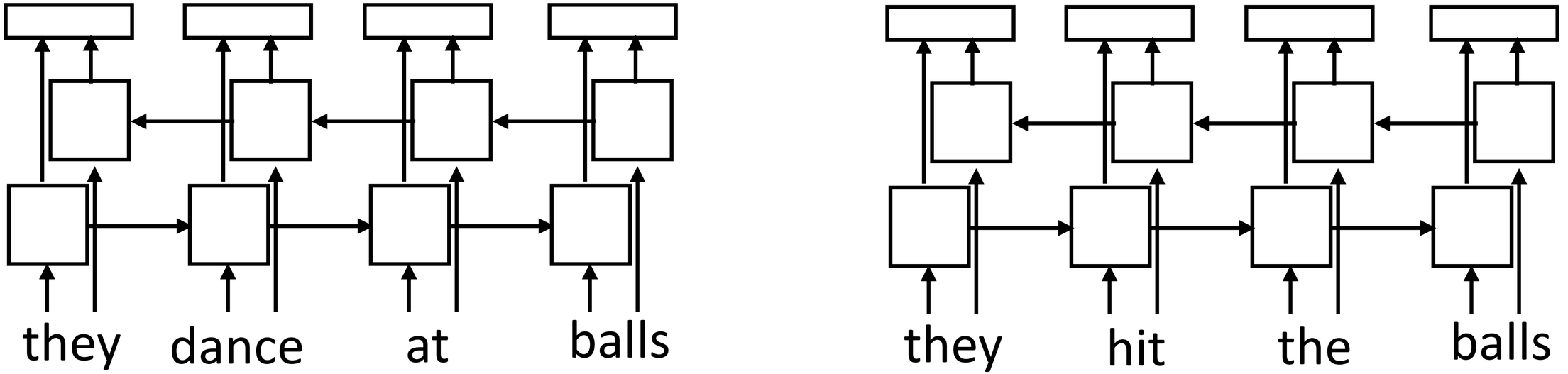
1 Introduction

Pre-trained word representations (Mikolov et al., 2013; Pennington et al., 2014) are a key component in many neural language understanding models. However, learning high quality representations can be challenging. They should ideally model 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). In this paper, we introduce a general purpose pre-trained neural language model that can be easily added to existing models for a wide range of language understanding problems. Our representations differ from traditional word type embeddings in that each token is assigned a representation that is a function of the entire input sentence. We use vectors derived from a bidirectional LSTM that is trained with a coupled language 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)

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)

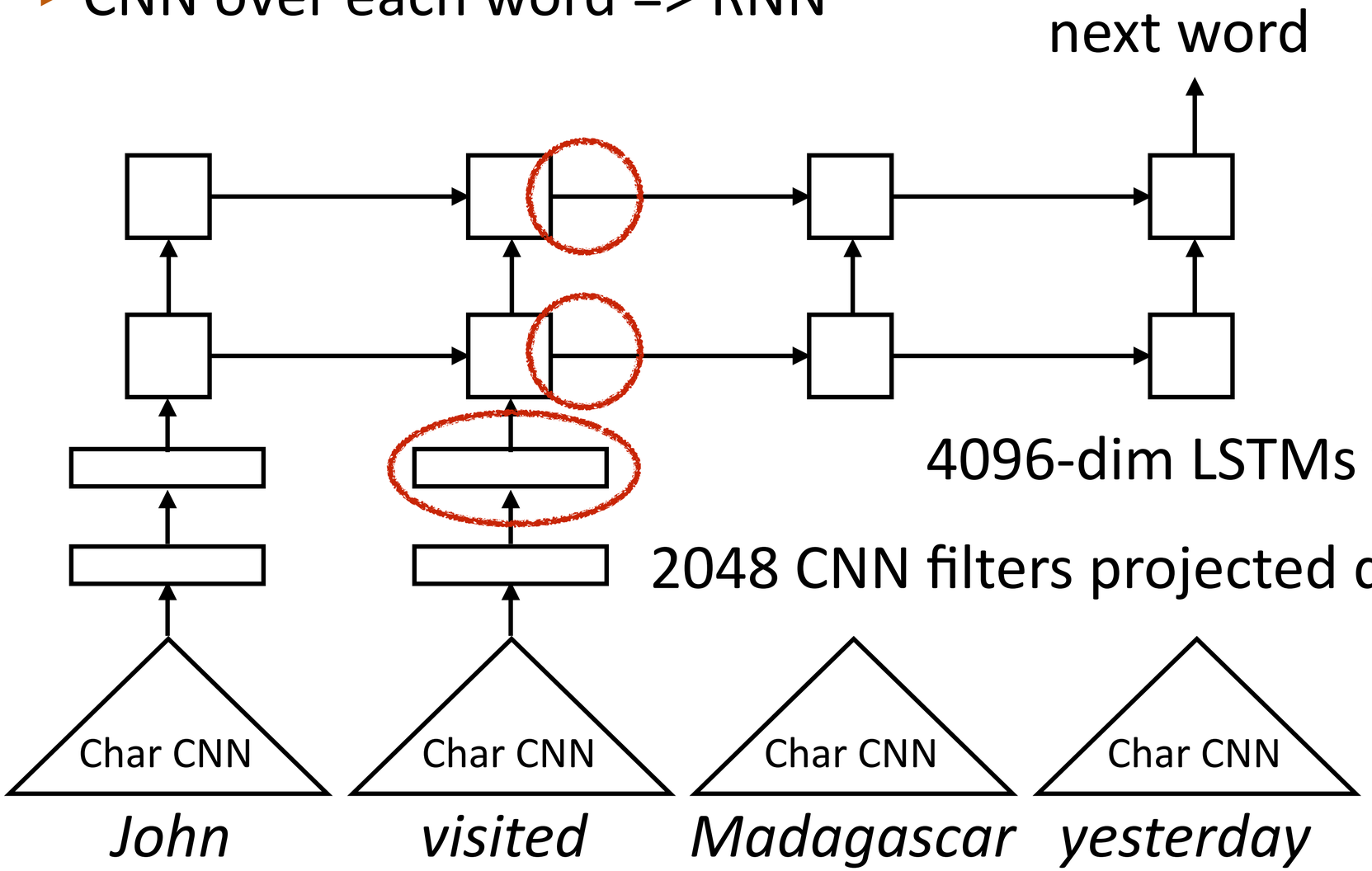
ELMo

- 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



Representation of *visited* (plus vectors from backwards LM)

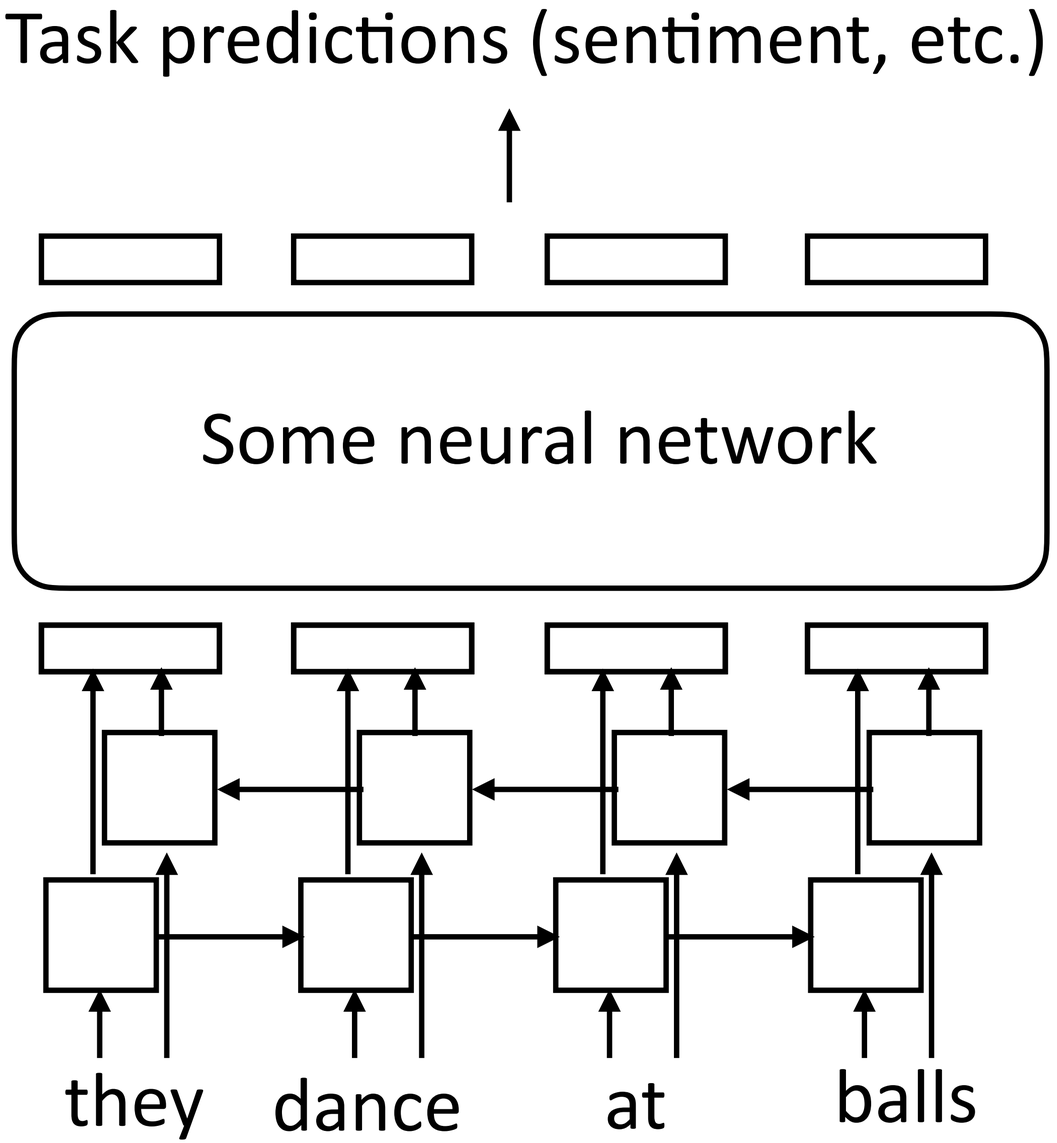
4096-dim LSTMs w/ 512-dim projections

2048 CNN filters projected down to 512-dim

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



How to apply ELMo?

Pretraining	Adaptation	NER CoNLL 2003	SA SST-2	Nat. lang. inference MNLI	Nat. lang. inference SICK-E	Semantic textual similarity SICK-R	Semantic textual similarity MRPC	Semantic textual similarity 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			Guidelines
Pretrain	Adapt.	Task	
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

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)

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.

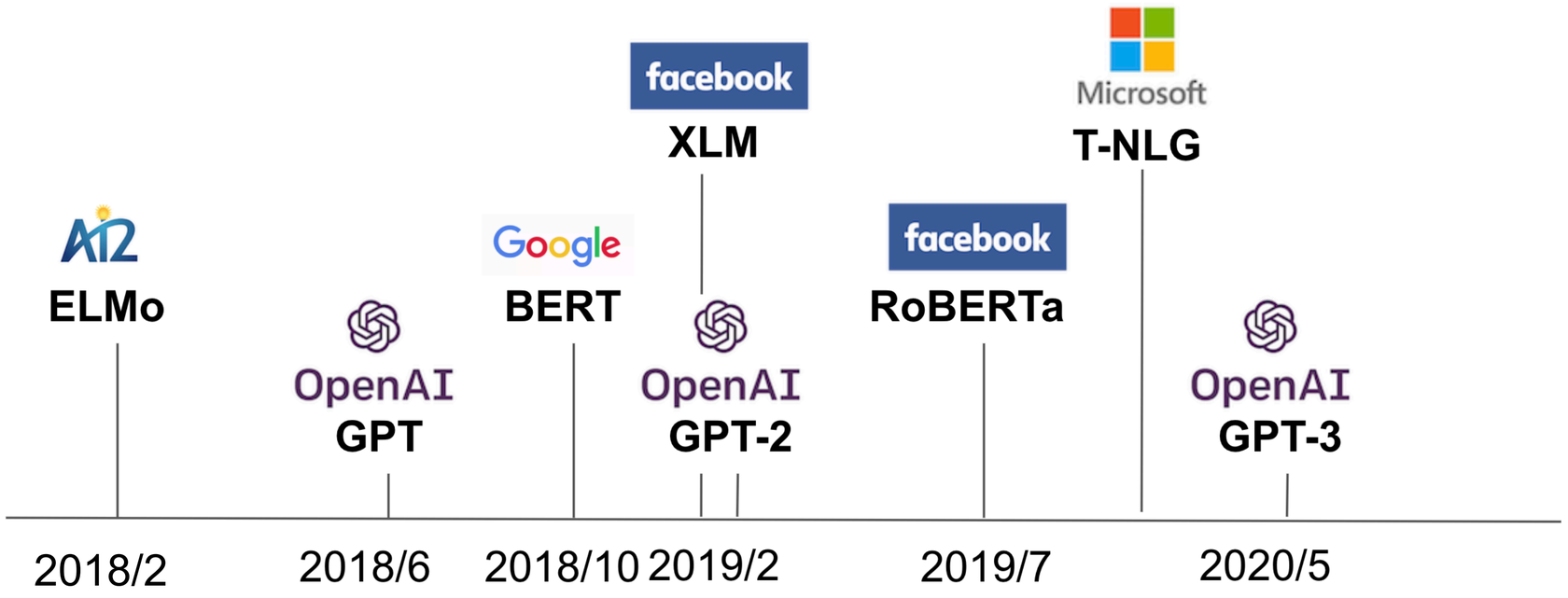
BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering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



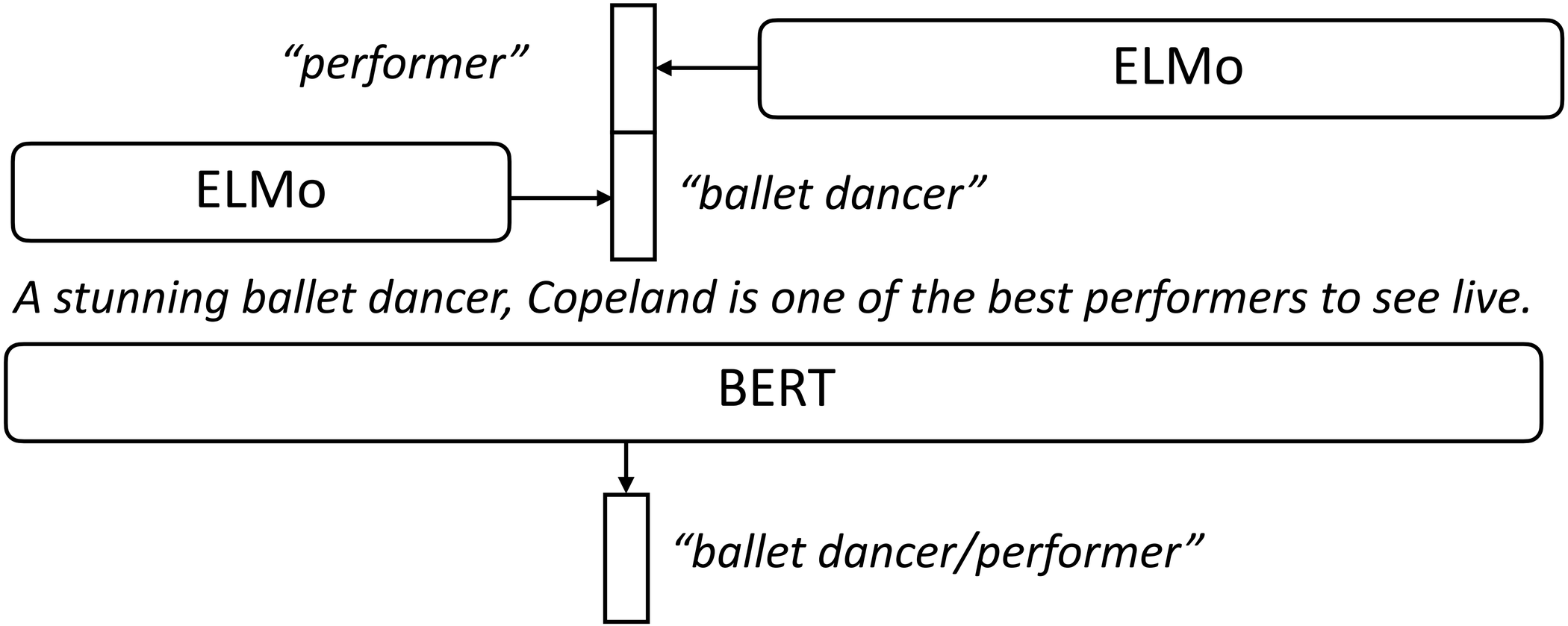
and many more …

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?



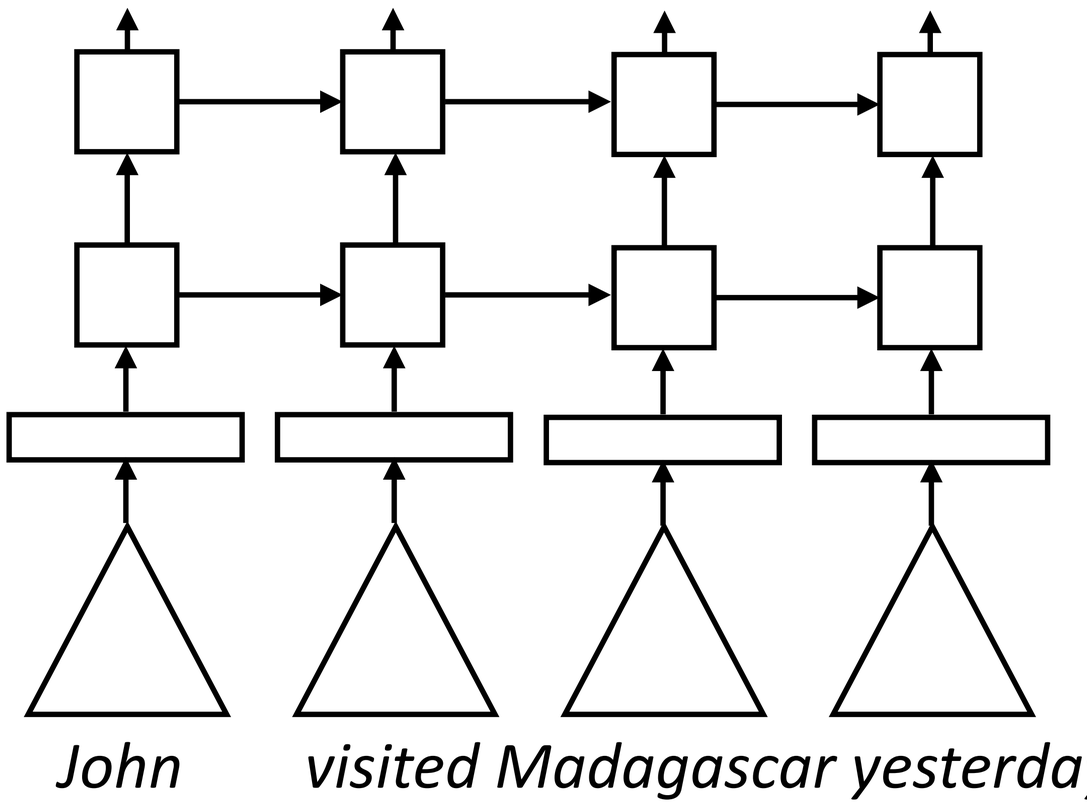
- ELMo looks at each direction in isolation; BERT looks at them jointly

Devlin et al. (2019)

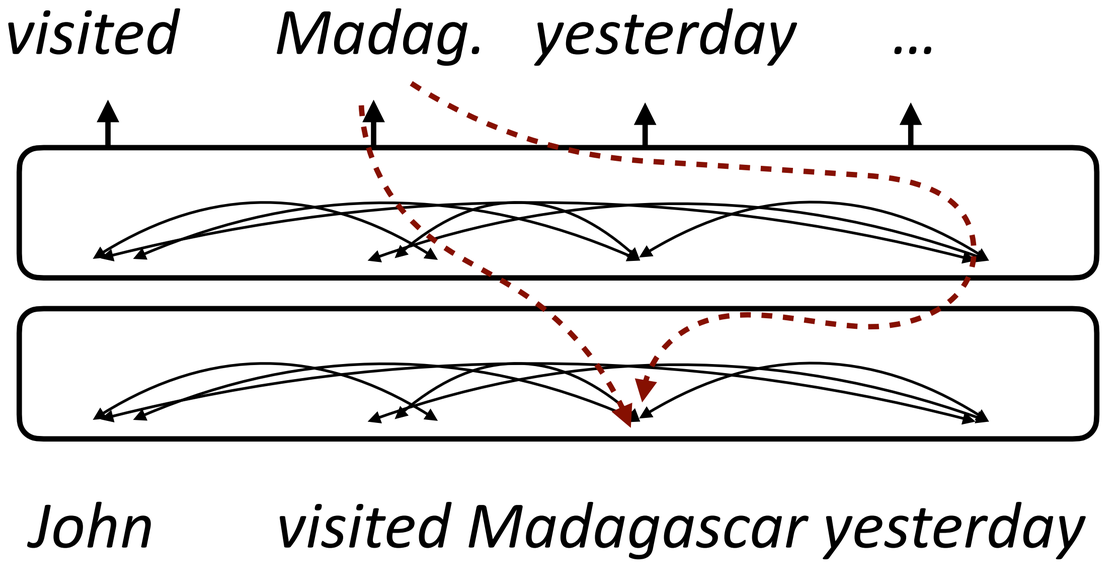
BERT

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

ELMo (Language Modeling)



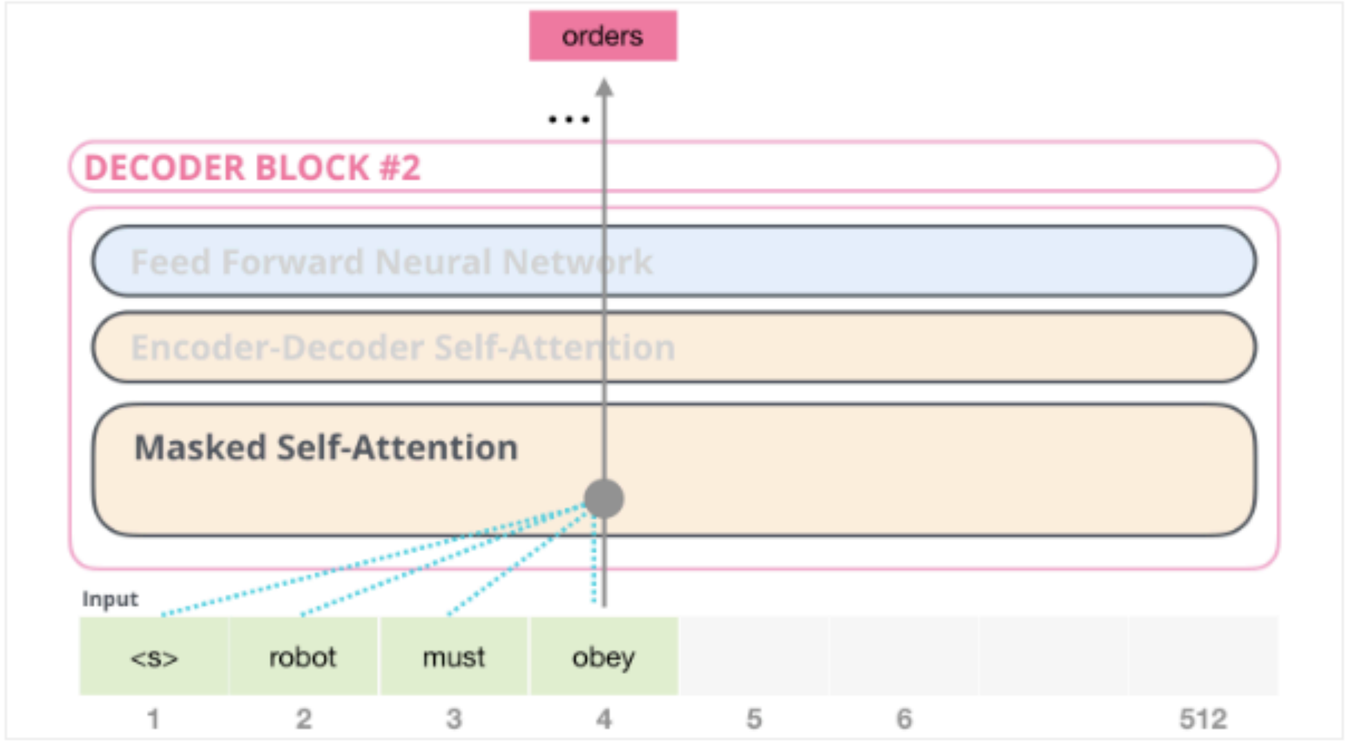
BERT



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

GPT (preview)

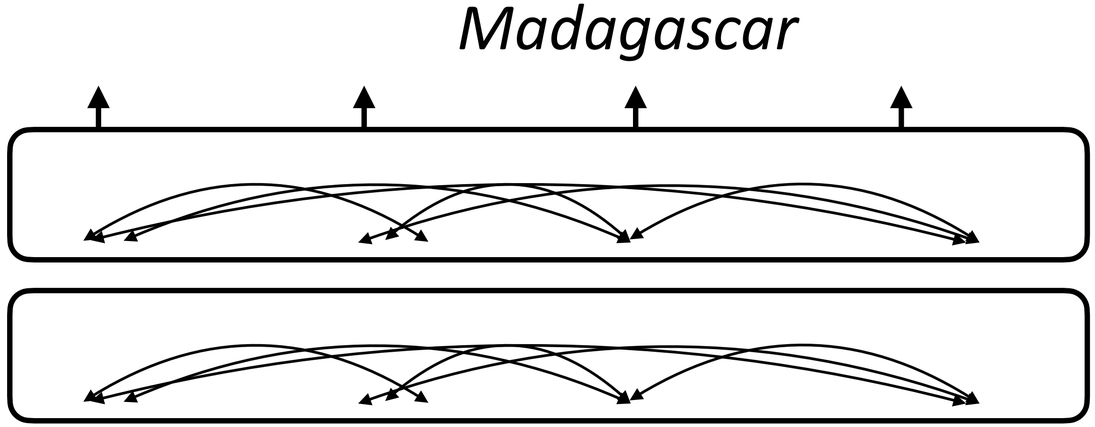
- Transformer with masked self-attention: each token can only attend to past tokens



Masked Language Modeling (MLM)

- 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



John *visited* [MASK] *yesterday*

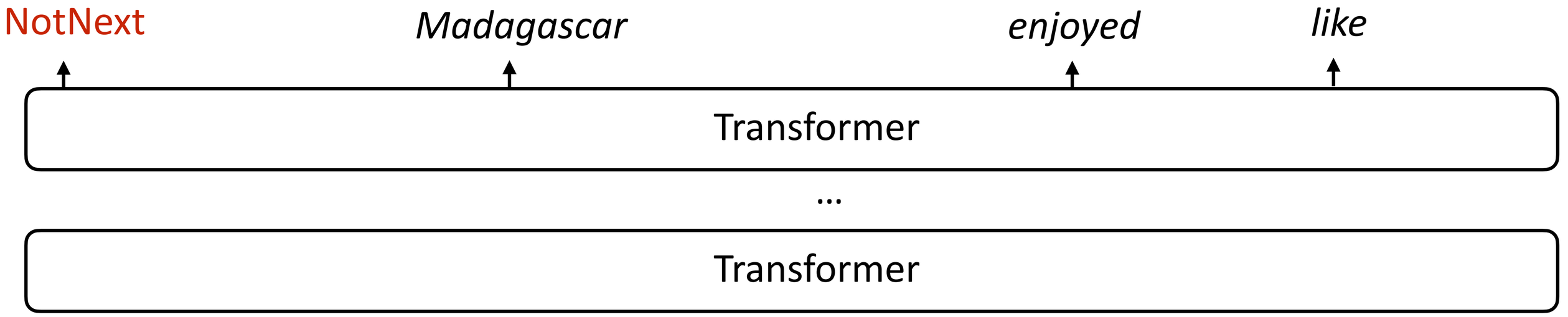
John *visited* **of** *yesterday*

John *visited* Madagascar *yesterday*

Devlin et al. (2019)

Next “Sentence” Prediction

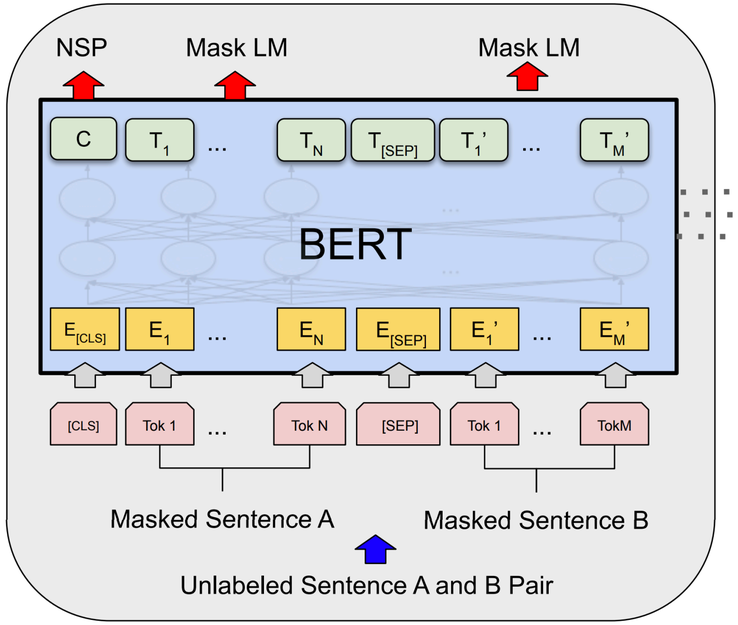
- Input: [CLS] Text chunk 1 [SEP] Text chunk 2
- 50% of the time, take the true next chunk of text, 50% of the time take a random other chunk. Predict whether the next chunk is the “true” next
- BERT objective: masked LM + next sentence prediction

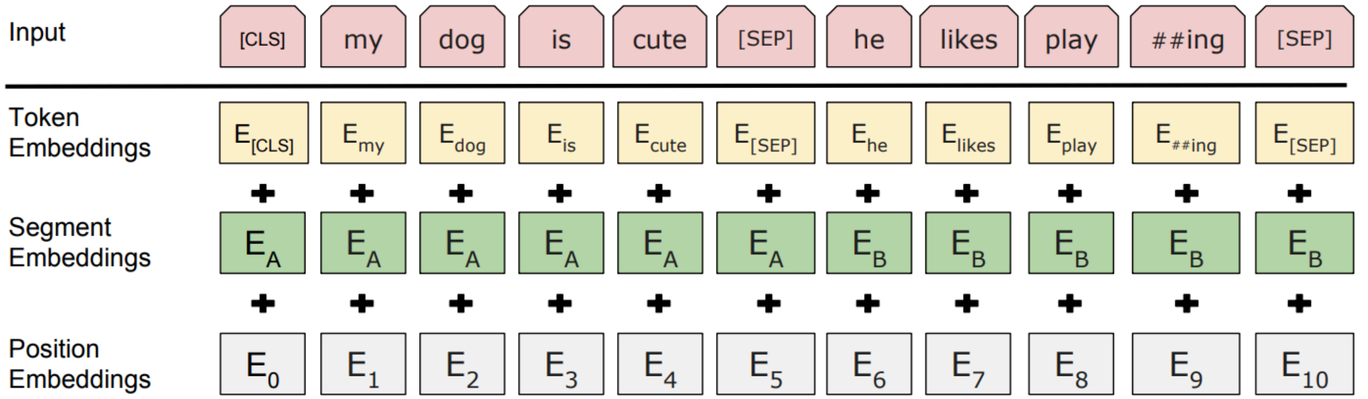


Devlin et al. (2019)

BERT Architecture

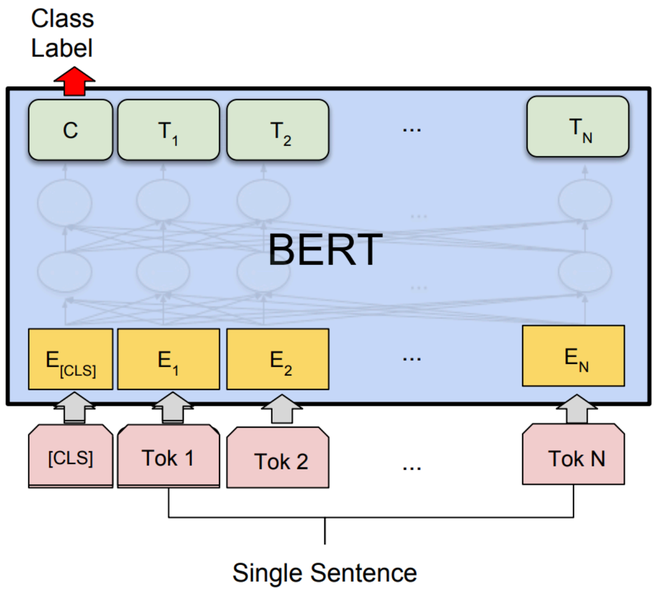
- BERT Base: 12 layers, 768-dim, 12 heads. Total params = 110M
- BERT Large: 24 layers, 1024-dim, 16 heads. Total params = 340M
- Positional embeddings and segment embeddings, 30k word pieces
- This is the model that gets **pre-trained** on a large corpus



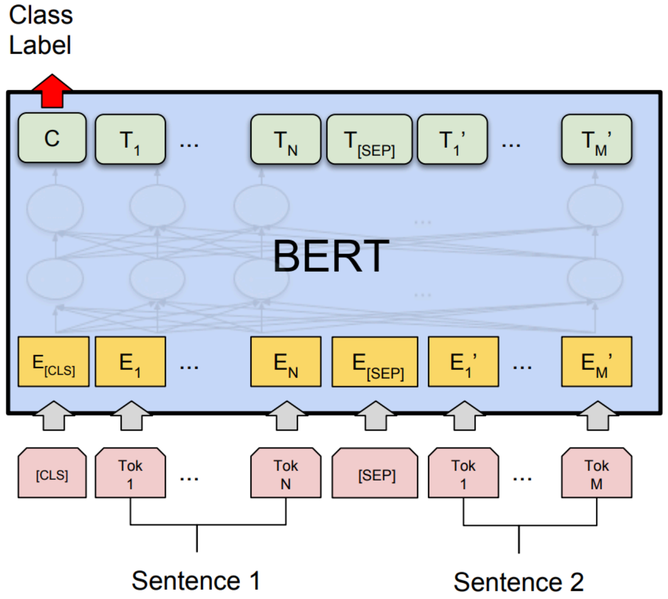


Devlin et al. (2019)

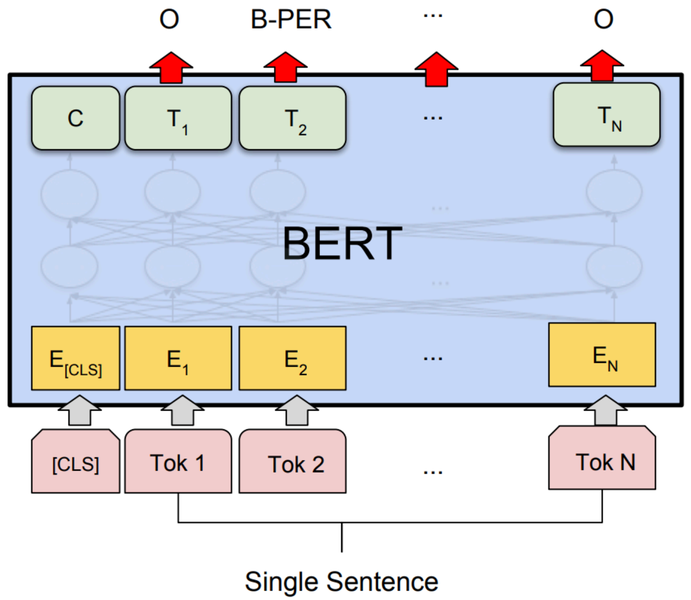
What can BERT do?



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



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



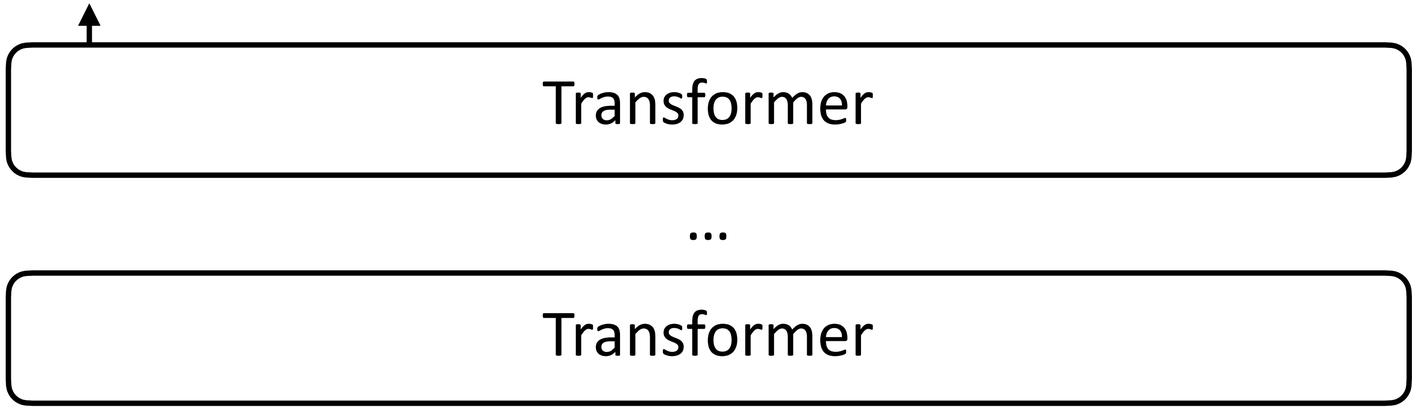
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

- 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

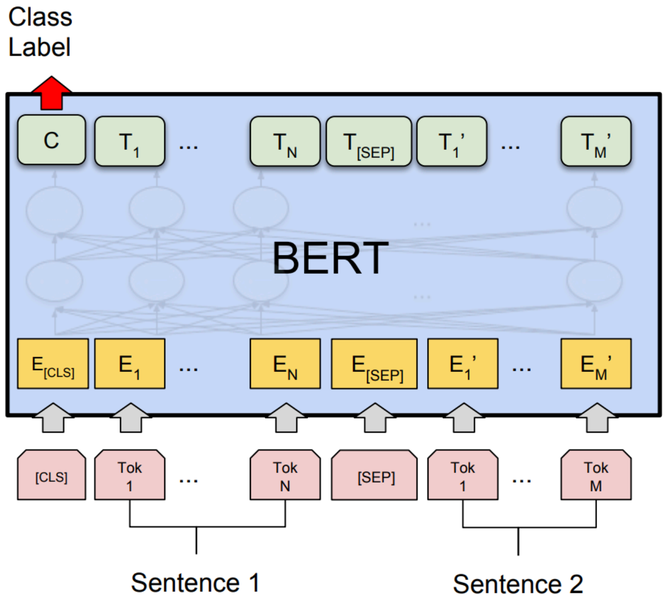
Devlin et al. (2019)

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)

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

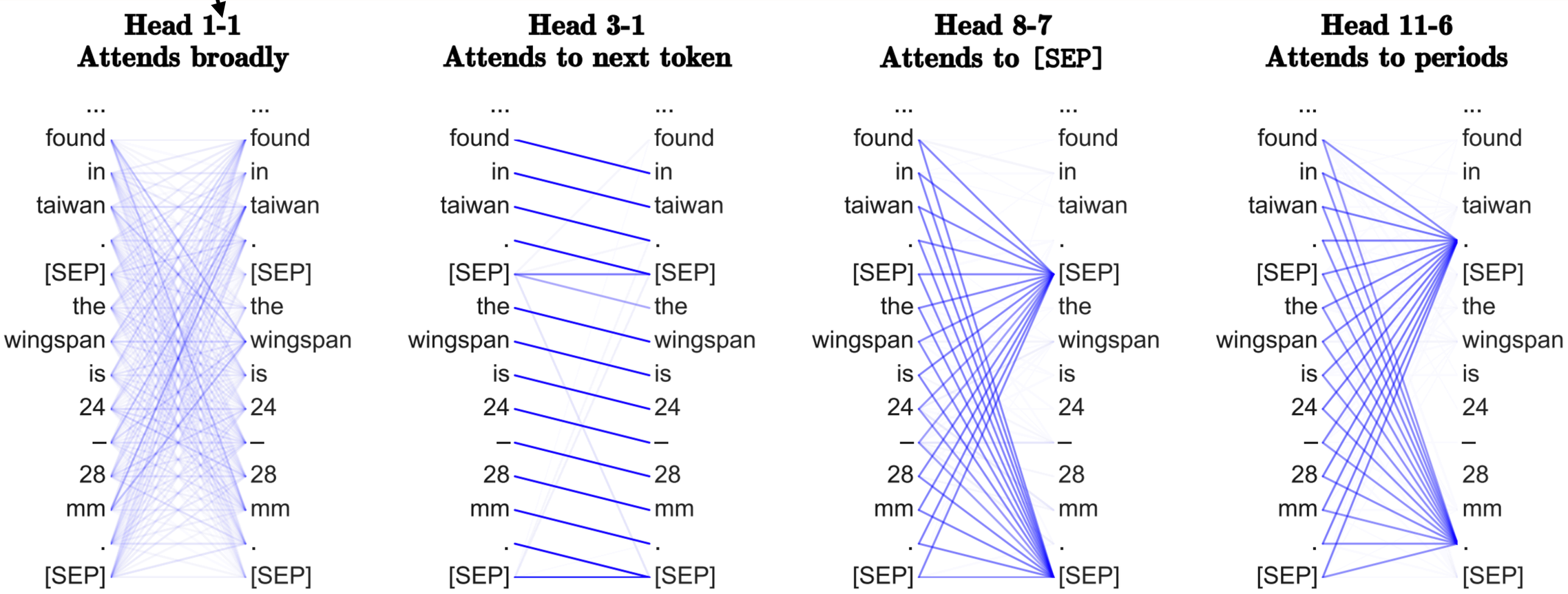
- Long history of this task: “Recognizing Textual Entailment” challenge in 2006 (Dagan, Glickman, Magnini)
- Early datasets: small (hundreds of pairs), very ambitious (lots of world knowledge, temporal reasoning, etc.)

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, …

What does BERT learn?

layer-head



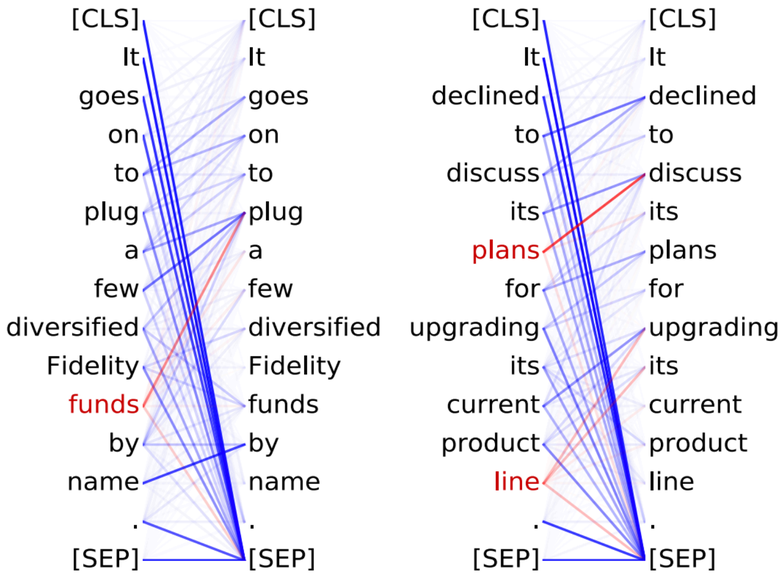
- Heads on transformers learn interesting and diverse things: content heads (attend based on content), positional heads (based on position), etc.

Clark et al. (2019)

What does BERT learn?

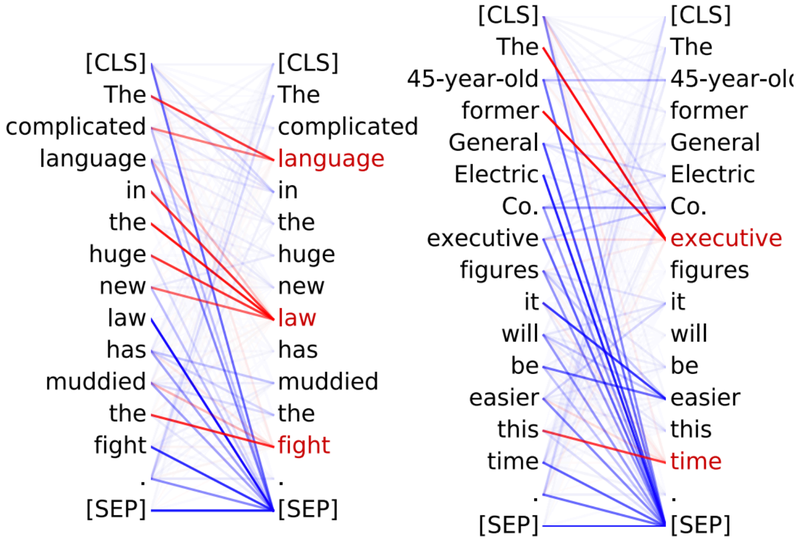
Head 8-10

- **Direct objects** attend to their verbs
- 86.8% accuracy at the `dobj` relation



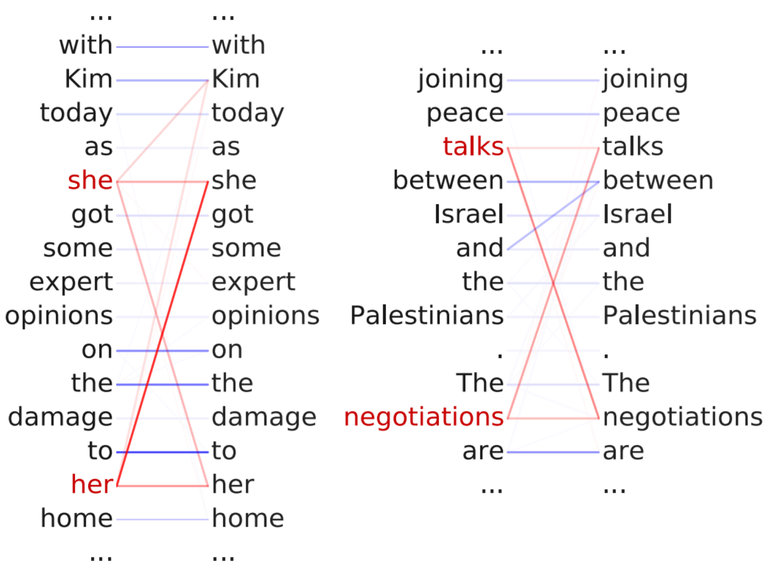
Head 8-11

- **Noun modifiers** (e.g., determiners) attend to their noun
- 94.3% accuracy at the `det` relation



Head 5-4

- **Coreferent** mentions attend to their antecedents
- 65.1% accuracy at linking the head of a coreferent mention to the head of an antecedent



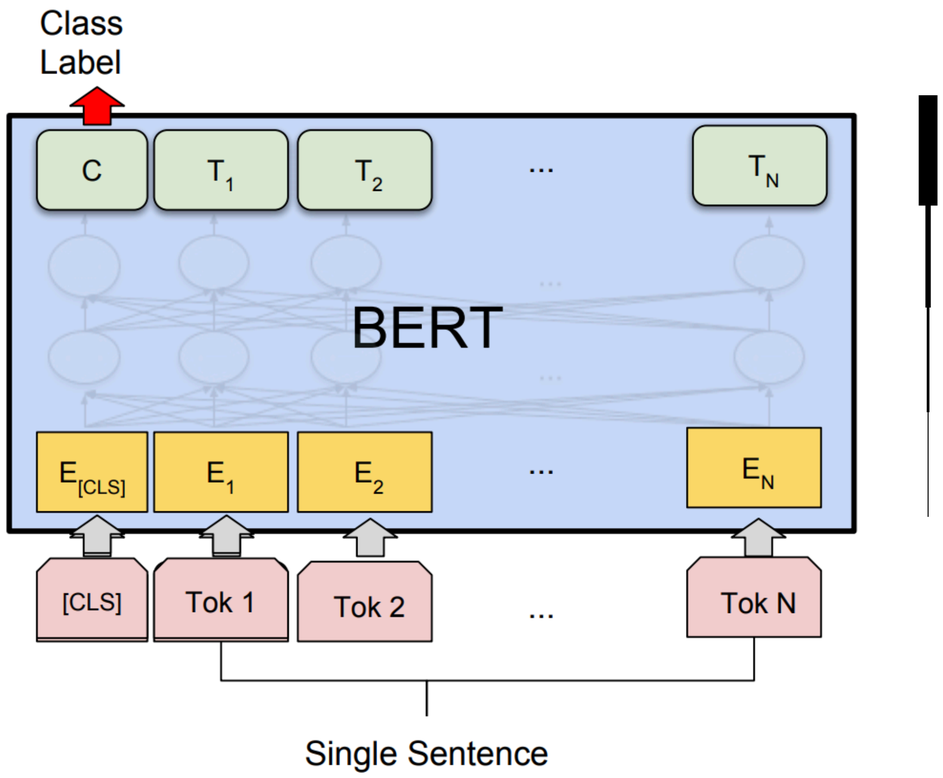
- Still way worse than what supervised systems can do, but interesting that this is learned organically

Clark et al. (2019)

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	Nat. lang. inference SICK-E	Semantic textual similarity SICK-R	Semantic textual similarity MRPC	Semantic textual similarity 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) 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	79.9	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

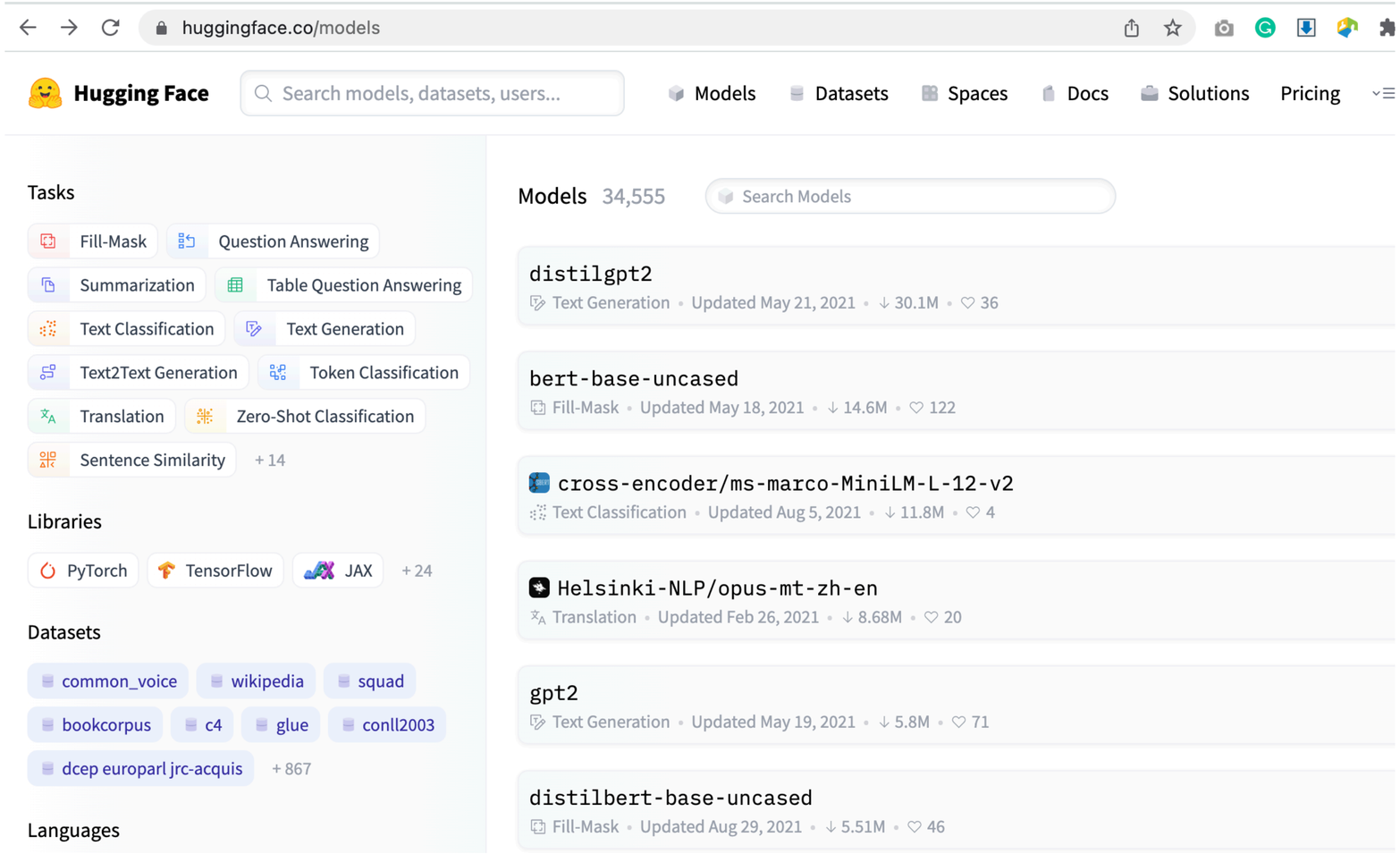
Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$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

I am passing an array list (Library_Class) as message header to camel route (Library_Class)

through java (Language) bean (Library_Class) as follows

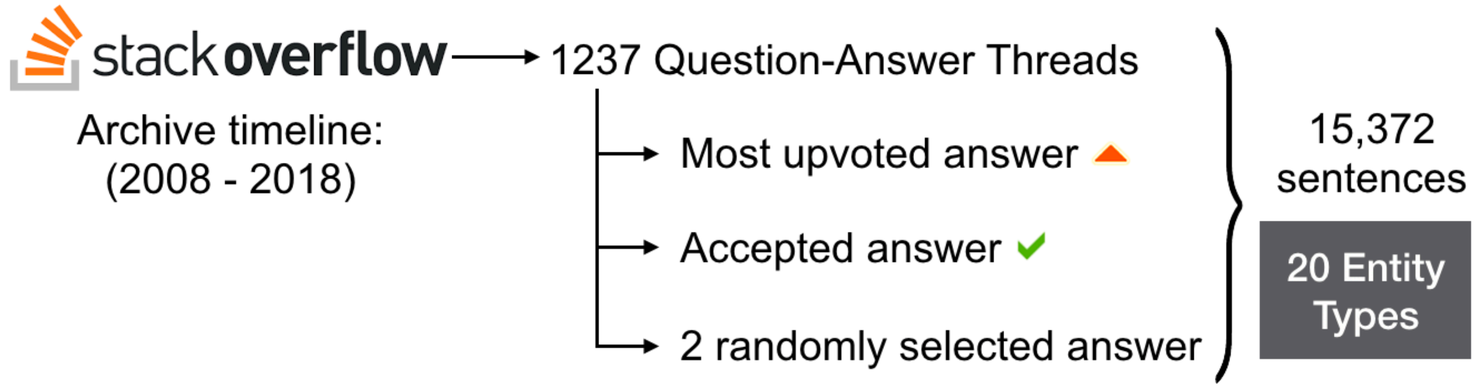
```
ArrayList<String> list=new ArrayList<String>();
                list.add("http://www.google.com");
                list.add("http://www.stackoverflow.com");
                list.add("http://www.tutorialspoint.com");
                list.add("http://localhost:8080/sampleExample/query");
                  exchange.getOut().setHeader("endpoints",list);
```

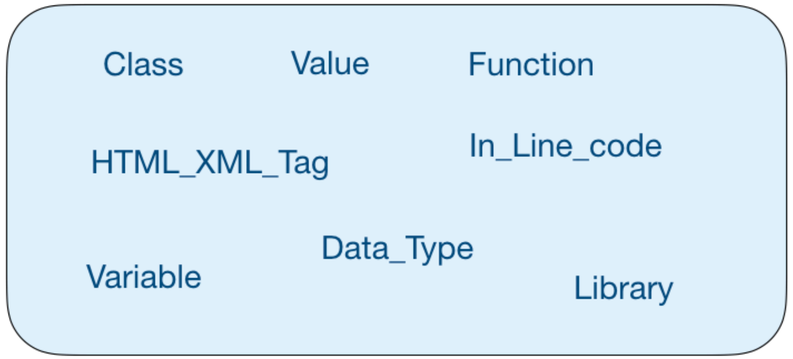
and, inside camel route (Library_Class) i want to iterate through this list (Variable_Name)

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

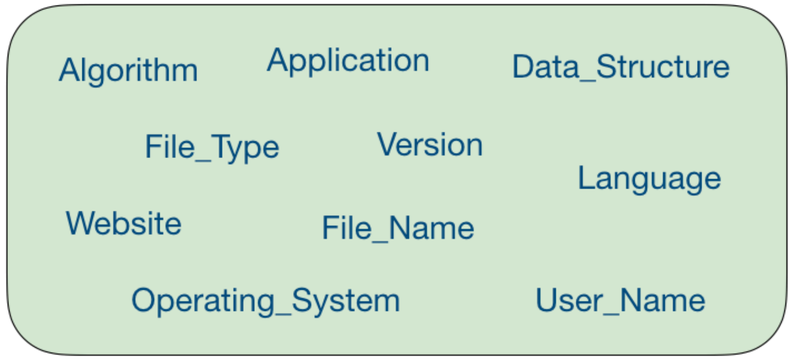
NER in StackOverflow

StackOverflow NER Corpus





Code Entity Types



Natural Language Entity

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

NER in StackOverflow

Two Main Challenges

(1) Polysemy — e.g., “key”, “windows”.
(2) Inline code — code-switch between human and programming languages.

Before adding element to array, check if key is numeric is is_numeric($key) function. If it return false, then, covert key to integer using typecasting, (int)$key.

Now, the array will have numeric keys only and can be ordered.

share improve this answer follow

answered Oct 23 '15 at 9:16

Ravneet

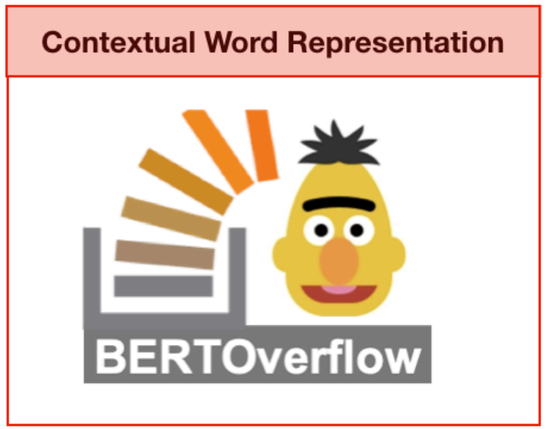
300 ● 1 ● 5

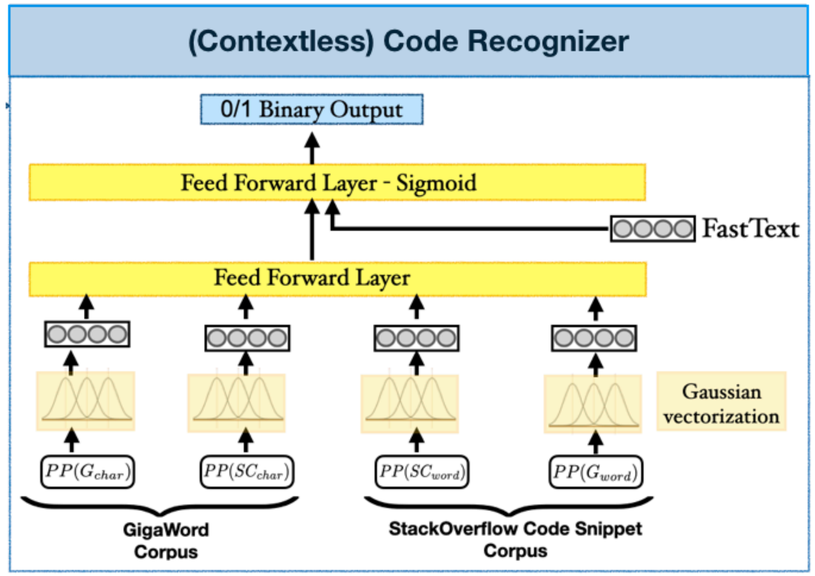
Jeniya Tabassum, Mounica Maddela, Alan Ritter, Wei Xu. “Code and Named Entity Recognition in StackOverflow” (ACL 2020)

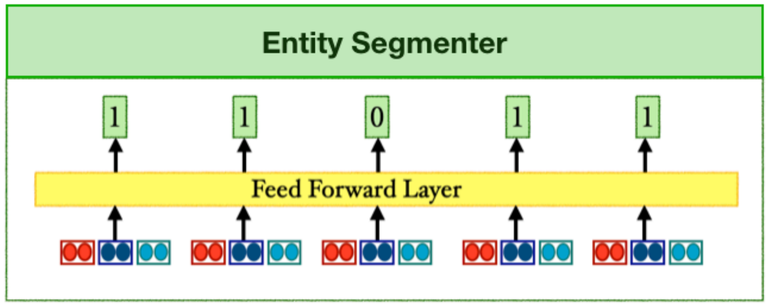
NER in StackOverflow

SoftNER Model

Combines BERTOverflow with domain-specific embeddings (Code Recognizer & Entity Segmenter) via attention.





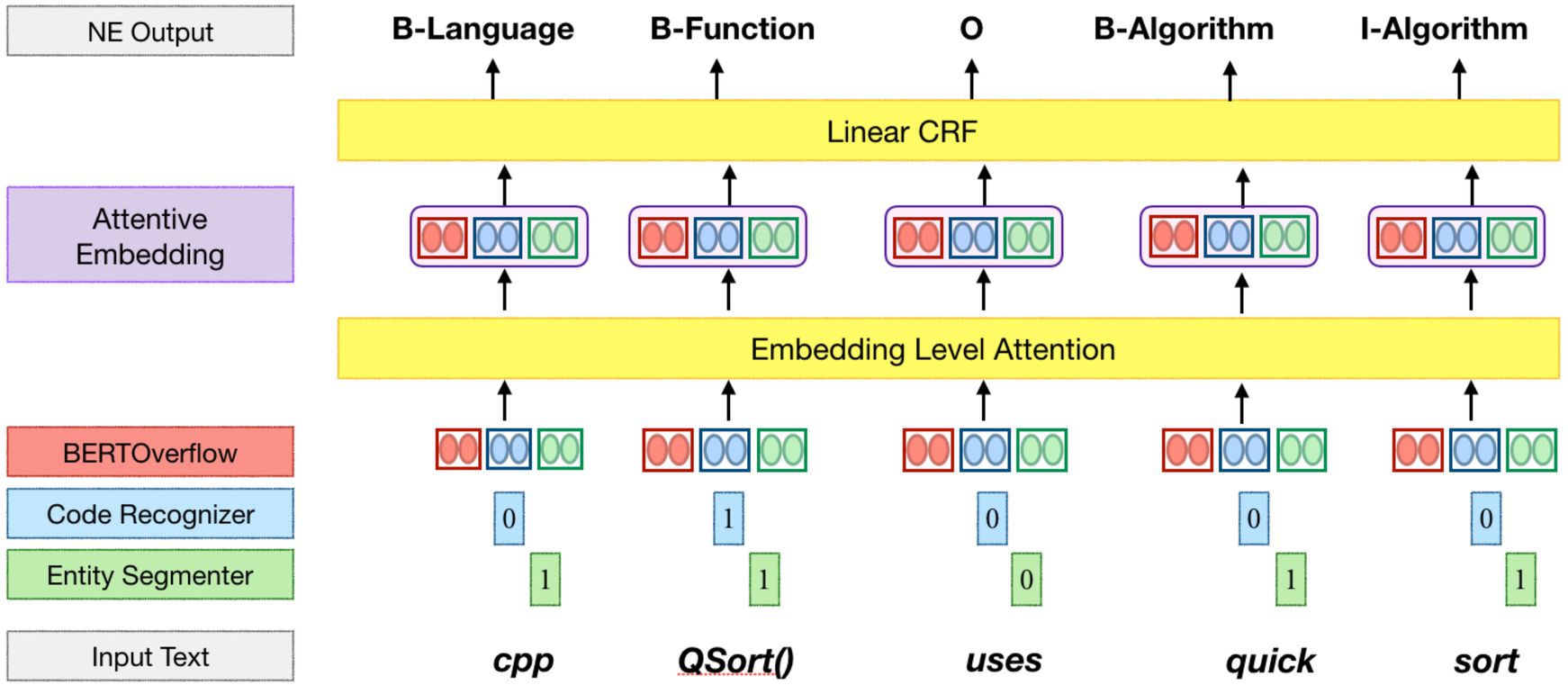


Jeniya Tabassum, Mounica Maddela, Alan Ritter, Wei Xu. “Code and Named Entity Recognition in StackOverflow” (ACL 2020)

NER in StackOverflow

SoftNER Model

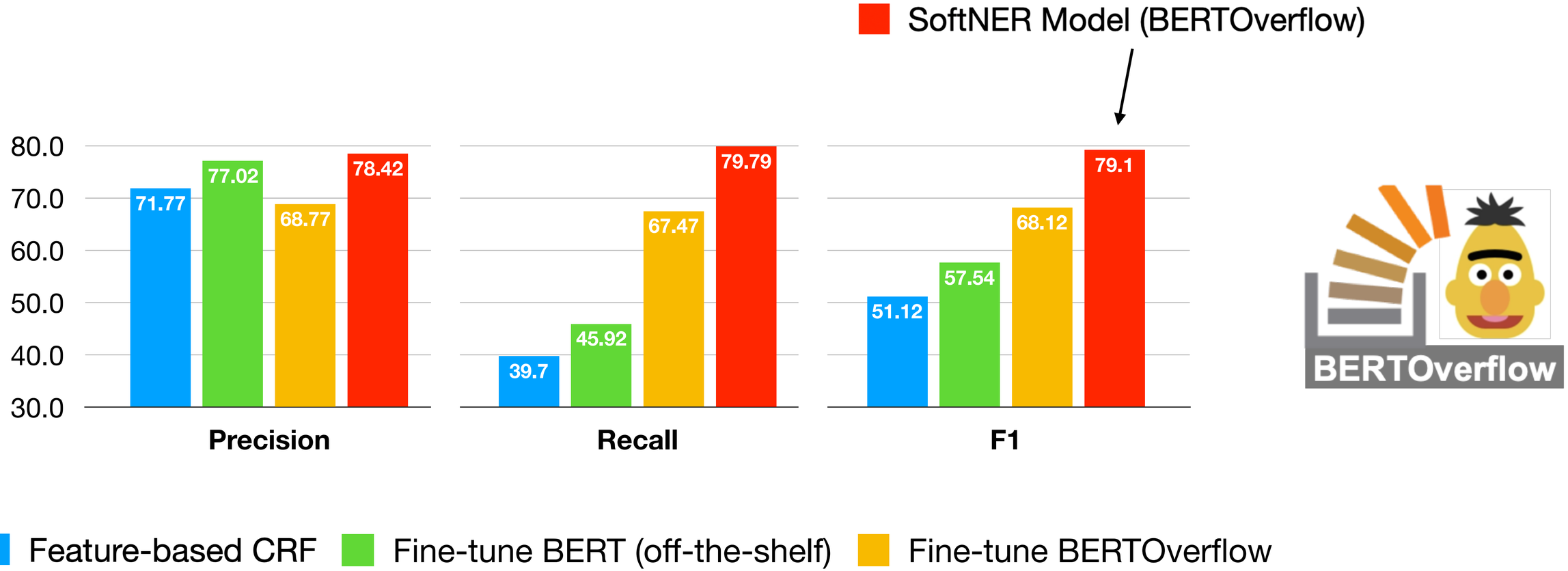
Combines BERTOverflow with domain-specific embeddings (Code Recognizer & Entity Segmenter) via attention.



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)

BERT/MLMs

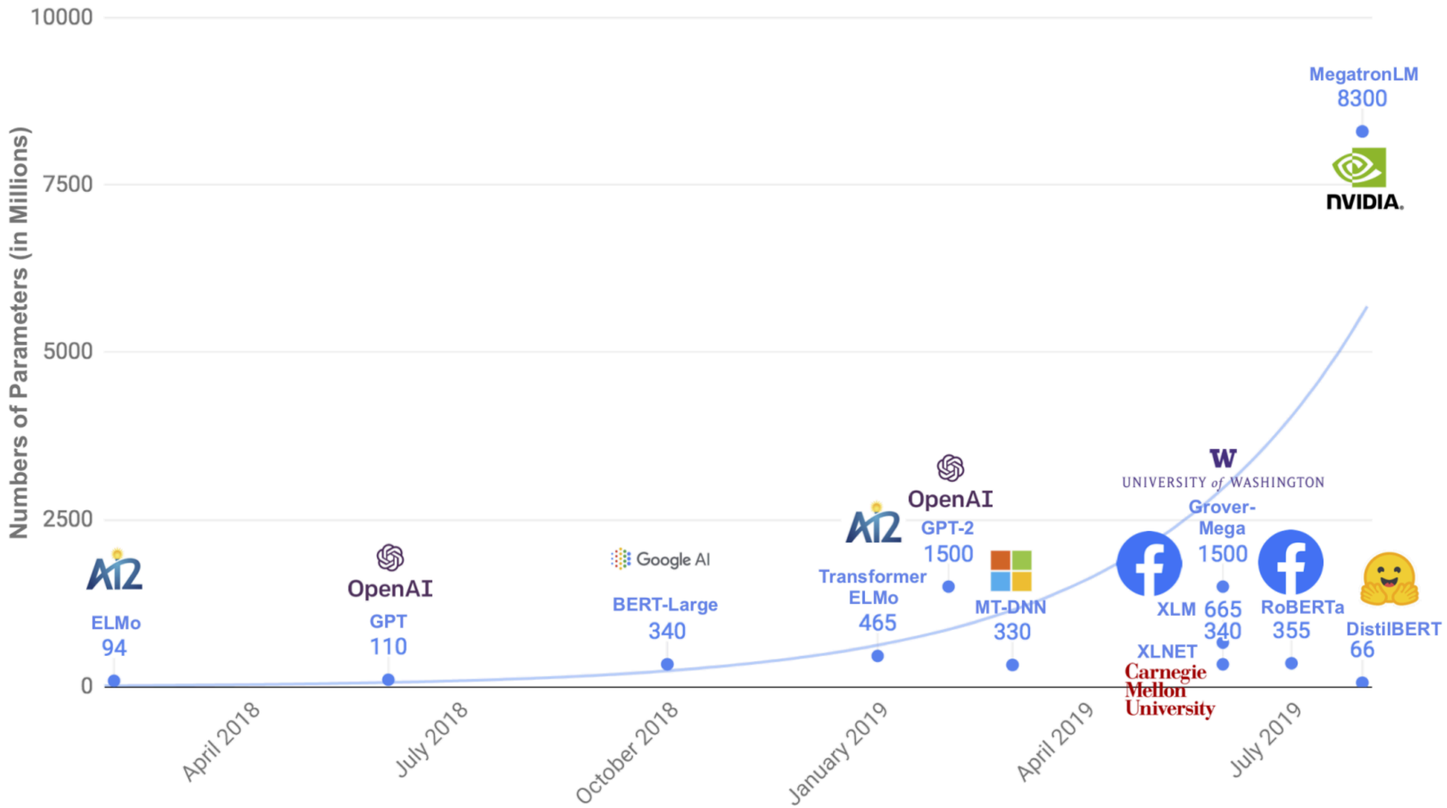
- There are lots of ways to train these models!
- Key factors:
  - Big enough model
  - Big enough data
  - Well-designed “self-supervised” objective (something like language modeling). Needs to be a hard enough problem!

Compressing BERT

DistilBERT

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

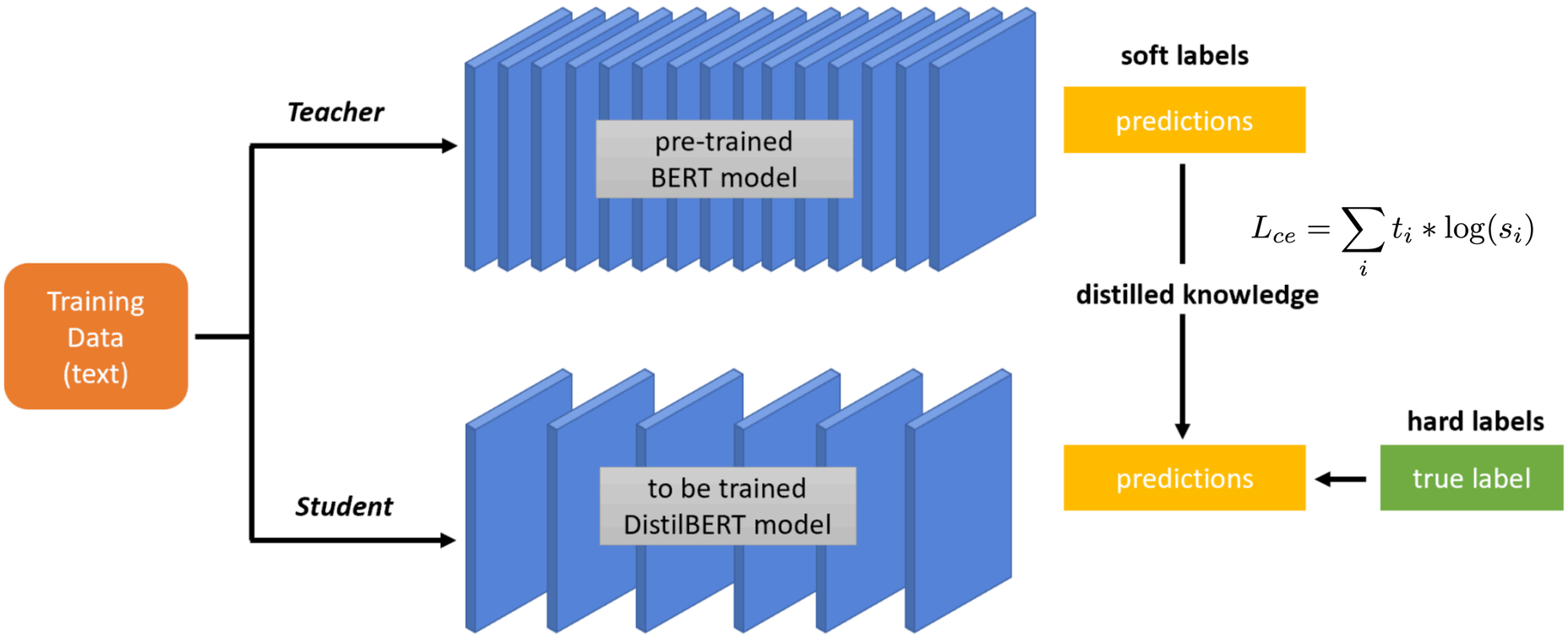
- DistilBERT (Sanh et al., 2019): nearly as good with half the parameters of BERT (via knowledge distillation)



Michel et al. (2019)

DistilBERT

- Knowledge Distillation

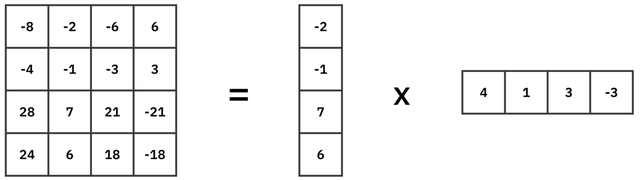


Sanh et al. (2019)

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



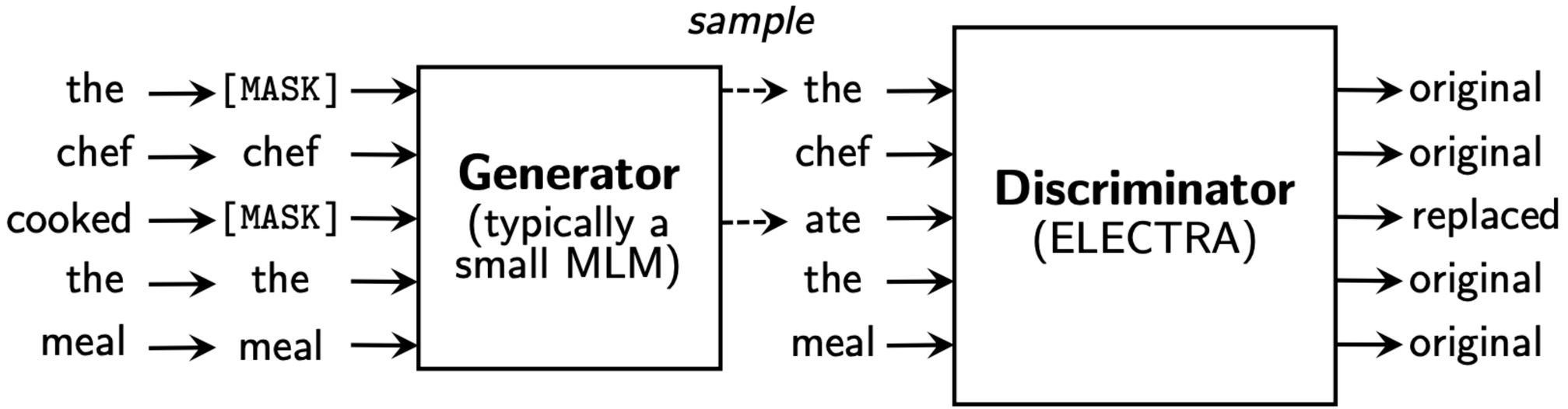
  Factor into two matrices with a low-rank approximation

  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)

Multilingual BERT

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

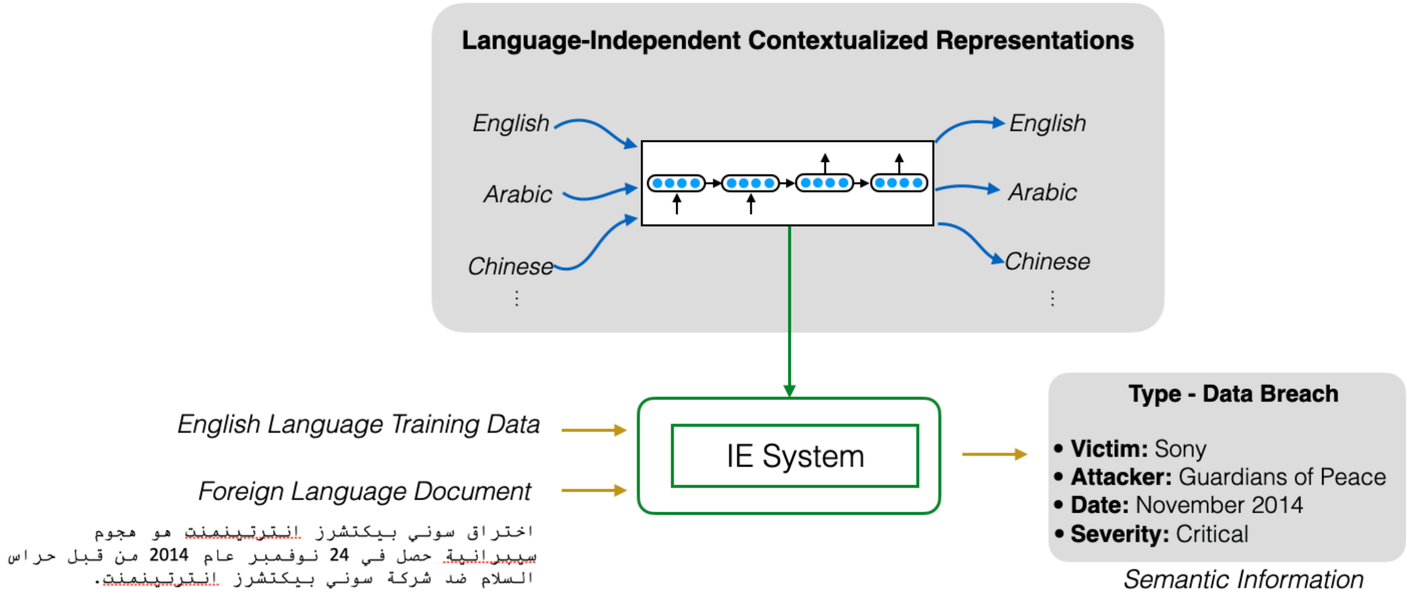
supervised learning

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.

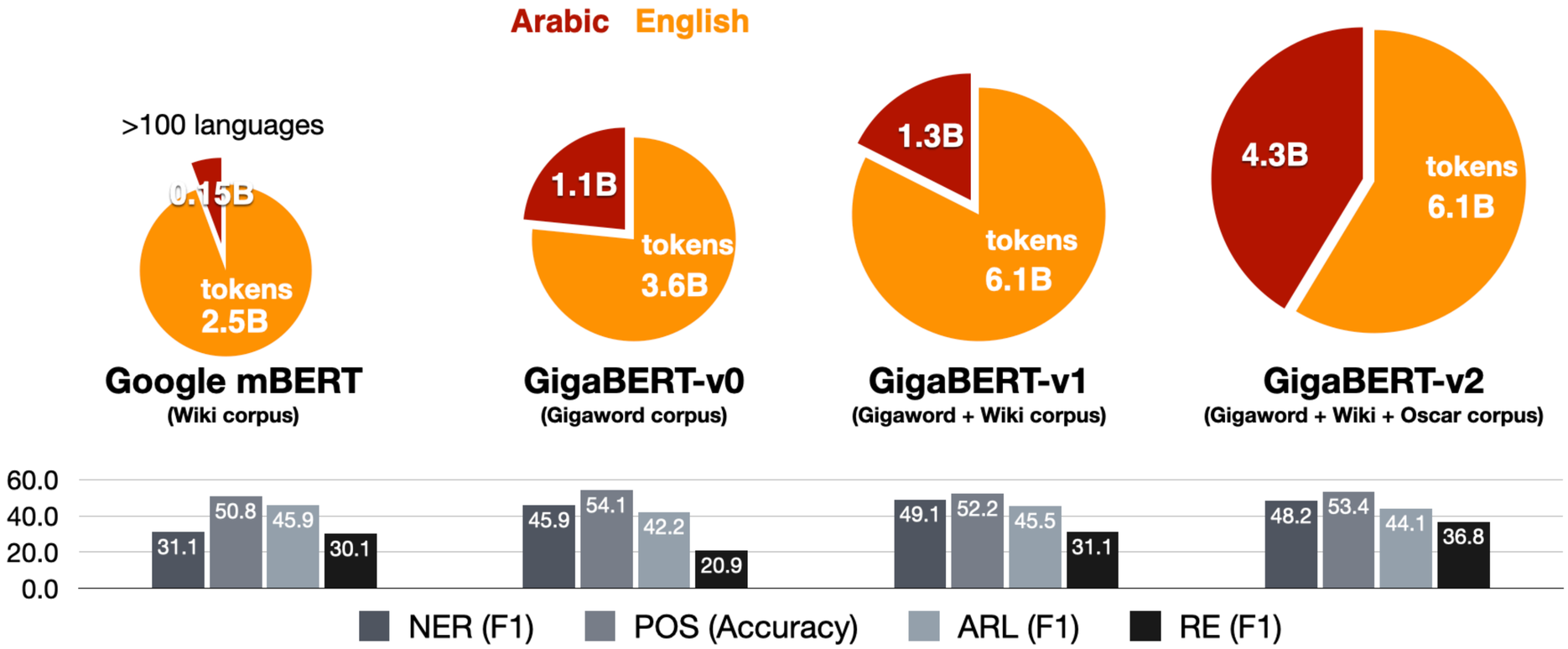


IARPA

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)

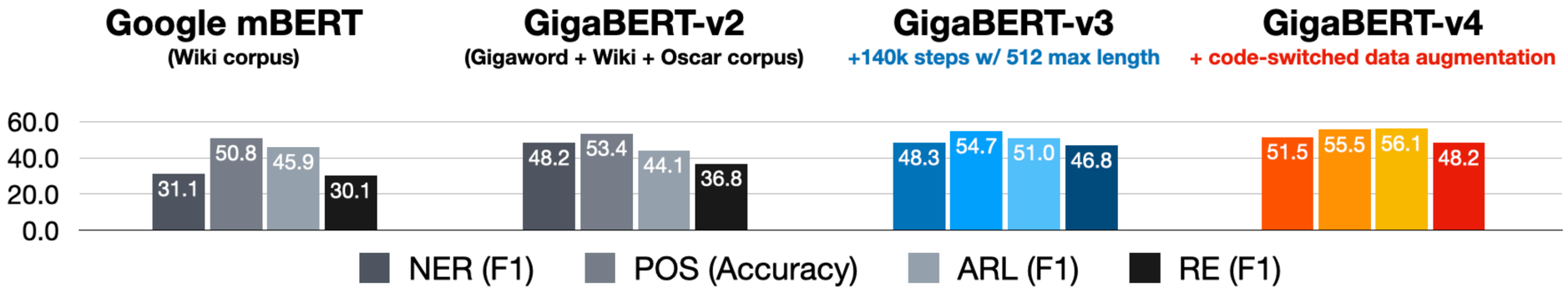
Bilingual BERT

- 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)

mmBERT

- Multi-stage training with different number of languages per stage

		Pre-training		Mid-training		Decay Phase	
Category	**Dataset**	**Tokens (B)**	**%**	**Tokens (B)**	**%**	**Tokens (B)**	**%**
Code	Code (ProLong)	–	–	–	–	2.8	2.7
Code	Starcoder	100.6	5.1	17.2	2.9	0.5	0.5
Crawl	DCLM	600.0	30.2	10.0	1.7	–	–
Crawl	DCLM (Dolmino)	–	–	40.0	6.7	2.0	2.0
Crawl	FineWeb2	1196.6	60.2	506.7	84.3	78.5	76.0
Instruction	Tulu Flan	15.3	0.8	3.1	0.5	1.0	1.0
Math	Dolmino Math	11.2	0.6	4.3	0.7	0.5	0.5
Reference	Books	4.3	0.2	3.9	0.7	2.2	2.1
Reference	Textbooks (ProLong)	–	–	–	–	3.1	3.0
Reference	Wikipedia (MegaWika)	4.7	0.2	1.2	0.2	9.5	9.2
Scientific	Arxiv	27.8	1.4	5.4	0.9	3.3	3.2
Scientific	PeS2o	8.4	0.4	3.2	0.5	–	–
Social	StackExchange	18.6	0.9	3.0	0.5	–	–
Social	StackExchange (Dolmino)	1.4	0.1	2.8	0.5	–	–
Total		1989.0	100.0	600.8	100.0	103.3	100.0

Marone et al. (2025)

mmBERT

- Pretrained on 3T tokens of multilingual text in over 1800 languages

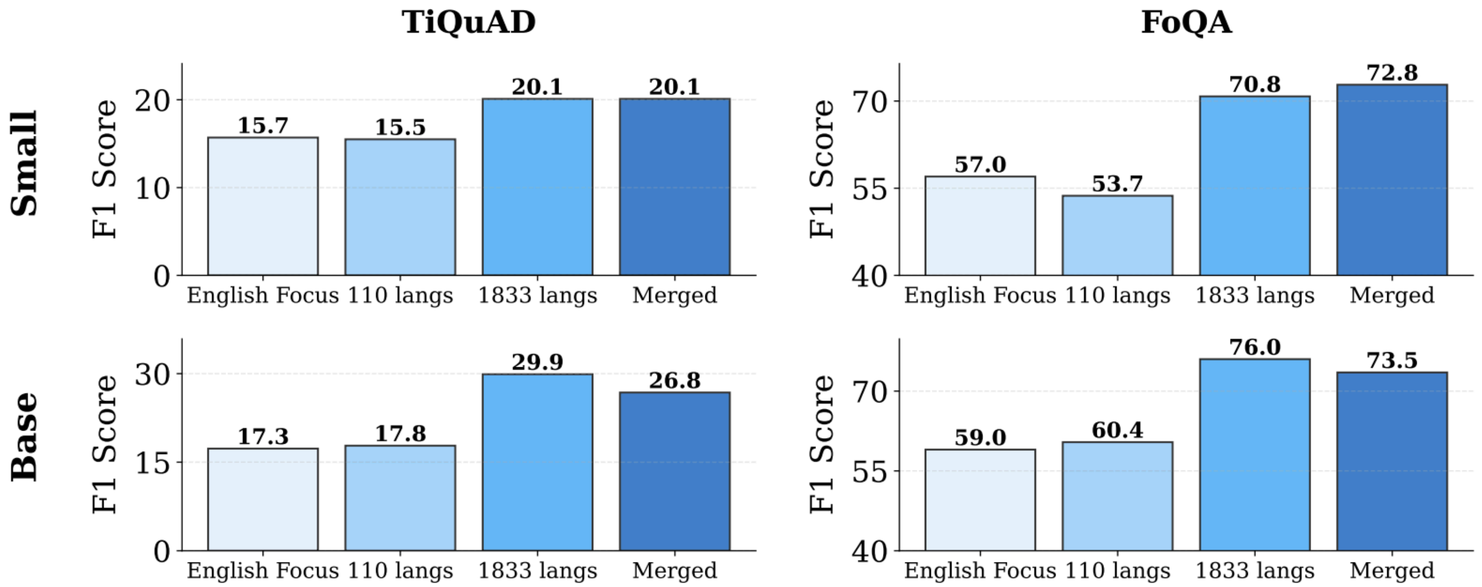


Figure 2: Performance of models using different decay phases on two languages (Tigray and Faroese) only added during the decay phase. We see that MMBERT with the 1833 language decay phase shows rapid performance improvements despite only having the models in the last 100B tokens of training. The final MMBERT models shows improvements by merging together checkpoints.

Marone et al. (2025)

mmBERT

- Pre-trained on 3T tokens of multilingual text in over 1800 languages

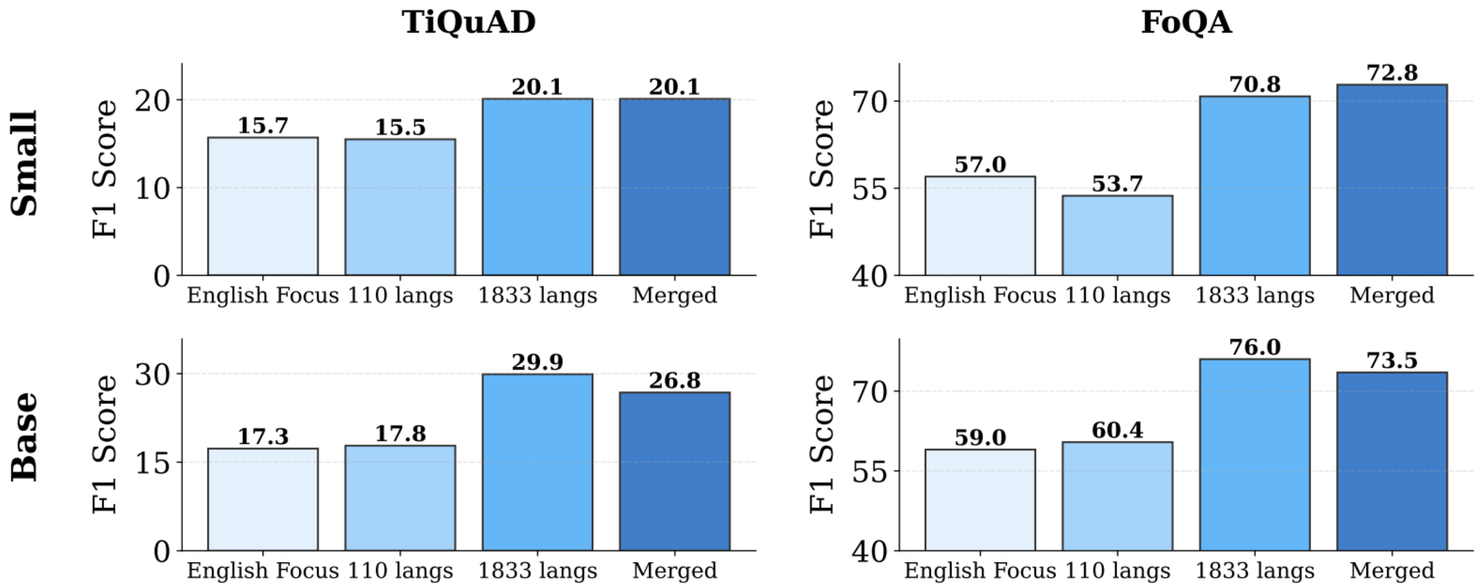


Figure 2: Performance of models using different decay phases on two languages (Tigray and Faroese) only added during the decay phase. We see that MMBERT with the 1833 language decay phase shows rapid performance improvements despite only having the models in the last 100B tokens of training. The final MMBERT models shows improvements by merging together checkpoints.

Marone et al. (2025)

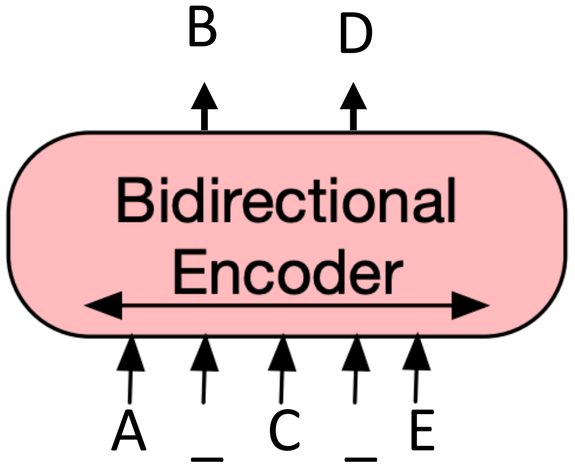
Other Encoder-only LMs

- XLM-RoBERTa (Conneau et al., 2020) for 100 languages
- mDeBERTa (He et al., 2021) for 100 languages
- MosiacBERT (Portes et al., 2023; Nussbaum et al., 2024)
- ModernBERT (Warner et al., 2024)
- EuroBERT (Boizard et al., 2025) for 15 languages
- NeoBERT (Le Breton et al., 2025)

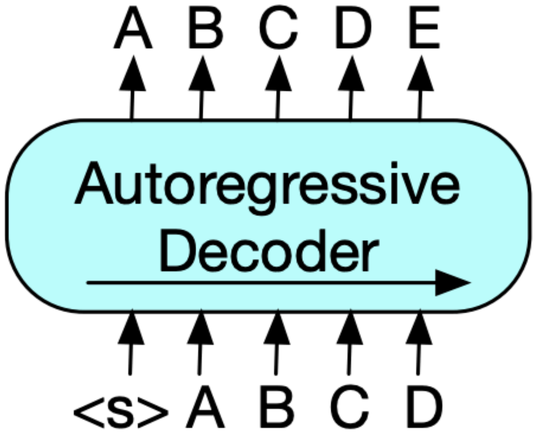
BART/T5
(encoder-decoder type of LMs)

BERT (encoder) vs. GPT (decoder)

- BERT: only parameters are an encoder, trained with masked language modeling objective
  - No way to do translation or left-to-right language modeling tasks



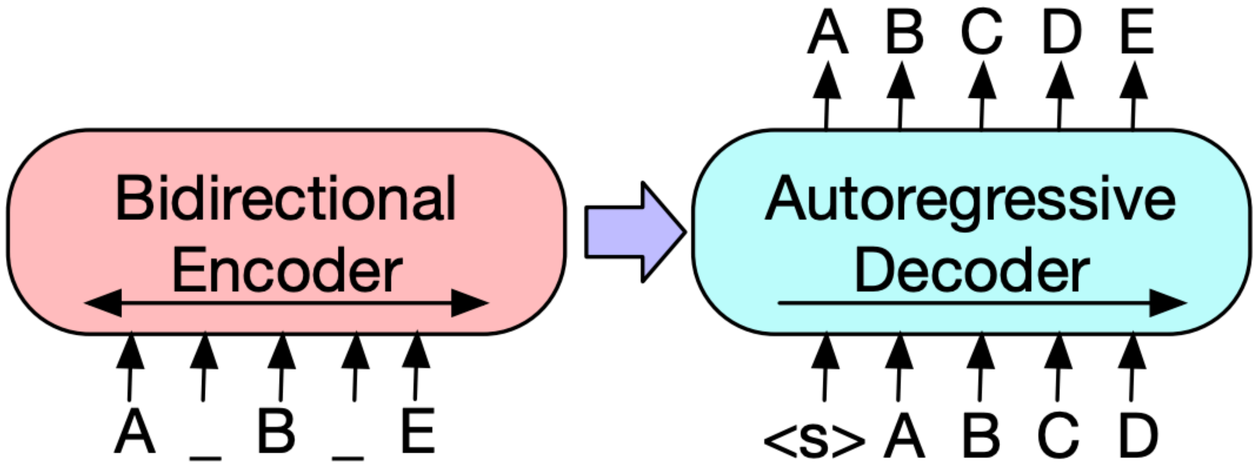
- GPT: only the decoder, autoregressive LM
  - (Small-size versions) Typically used for unconditioned generation tasks, e.g. story or dialog generation



Lewis et al. (2019)

BART (encoder-decoder)

- What to do for seq2seq tasks?
- Sequence-to-sequence BERT variant: permute/make/delete tokens, then predict full sequence autoregressively
- For downstream tasks: feed document into both encoder + decoder, use decoder hidden state as output

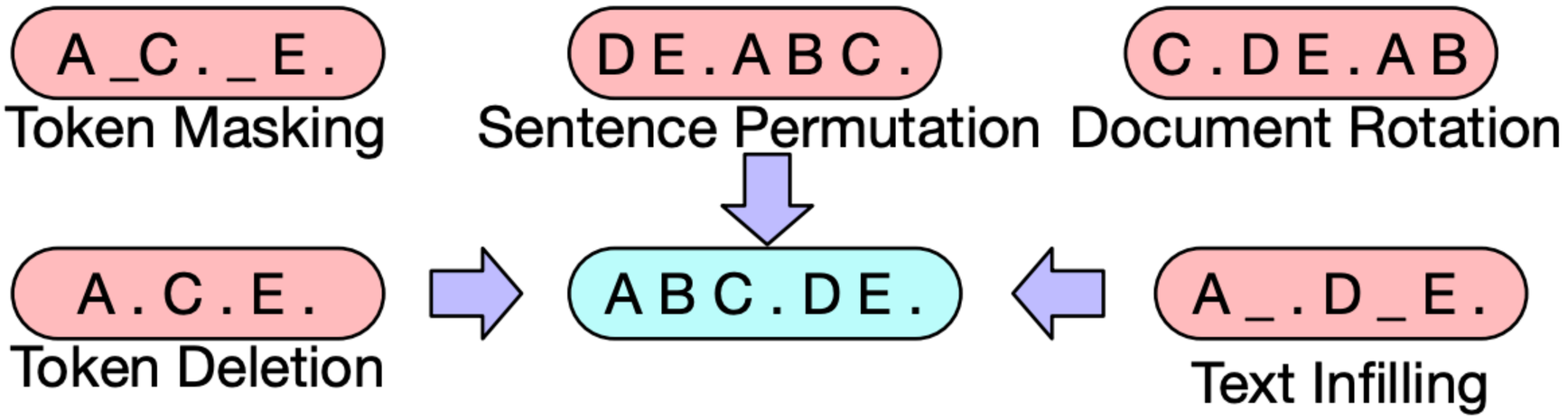


- Good results on translation, summarization tasks

Lewis et al. (October 30, 2019)

BART

- BART uses multiple de-noising LM objective:



Infilling is longer spans than masking

Lewis et al. (2019)

BART

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	**90.8**	84.0	24.26	**6.61**	**11.05**	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	**90.8**	83.8	24.17	6.62	11.12	**5.41**

- Infilling is all-around a bit better than masking or deletion
- Final system: combination of infilling and sentence permutation

Lewis et al. (2019)

SQuAD 2.0 (span-based QA)

- SQuAD 1.1 contains 100k+ QA pairs from 500+ Wikipedia articles.
- SQuAD 2.0 includes additional 50k questions that cannot be answered.
- These questions were crowdsourced.

Passage

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.

Question: Which NFL team won Super Bowl 50?
Answer: Denver Broncos

Question: What does AFC stand for?
Answer: American Football Conference

Question: What year was Super Bowl 50?
Answer: 2016

Rajpurkar et al. (2016)

BART

	SQuAD 1.1 EM/F1	SQuAD 2.0 EM/F1	MNLI m/mm	SST Acc	QQP Acc	QNLI Acc	STS-B Acc	RTE Acc	MRPC Acc	CoLA Mcc
BERT	84.1/90.9	79.0/81.8	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
UniLM	-/-	80.5/83.4	87.0/85.9	94.5	-	92.7	-	70.9	-	61.1
XLNet	**89.0**/94.5	86.1/88.8	89.8/-	95.6	91.8	93.9	91.8	83.8	89.2	63.6
RoBERTa	88.9/**94.6**	**86.5/89.4**	**90.2/90.2**	96.4	92.2	94.7	**92.4**	86.6	**90.9**	**68.0**
BART	88.8/**94.6**	86.1/89.2	89.9/90.1	**96.6**	**92.5**	**94.9**	91.2	**87.0**	90.4	62.8

- Results on GLUE benchmark are not better than RoBERTa

Lewis et al. (2019)

BART

	SQuAD 1.1 EM/F1	SQuAD 2.0 EM/F1	MNLI m/mm	SST Acc	QQP Acc	QNLI Acc	STS-B Acc	RTE Acc	MRPC Acc	CoLA Mcc
BERT	84.1/90.9	79.0/81.8	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
UniLM	-/-	80.5/83.4	87.0/85.9	94.5	-	92.7	-	70.9	-	61.1
XLNet	**89.0**/94.5	86.1/88.8	89.8/-	95.6	91.8	93.9	91.8	83.8	89.2	63.6
RoBERTa	88.9/**94.6**	**86.5/89.4**	**90.2/90.2**	96.4	92.2	94.7	**92.4**	86.6	**90.9**	**68.0**
BART	88.8/**94.6**	86.1/89.2	89.9/90.1	**96.6**	**92.5**	**94.9**	91.2	**87.0**	90.4	62.8

- Results on GLUE benchmark are not better than RoBERTa

Lewis et al. (2019)

CoLA

- Corpus of Linguistic Acceptability (CoLA); to test whether a model can recognize (a) morphological anomalies, (b) syntactic anomalies, and (c) semantic anomalies.

Label	Sentence	Source
*	The more books I ask to whom he will give, the more he reads.	Culicover and Jackendoff (1999)
✓	I said that my father, he was tight as a hoot-owl.	Ross (1967)
✓	The jeweller inscribed the ring with the name.	Levin (1993)
*	many evidence was provided.	Kim and Sells (2008)
✓	They can sing.	Kim and Sells (2008)
✓	The men would have been all working.	Baltin (1982)
*	Who do you think that will question Seamus first?	Carnie (2013)
*	Usually, any lion is majestic.	Dayal (1998)
✓	The gardener planted roses in the garden.	Miller (2002)
✓	I wrote Blair a letter, but I tore it up before I sent it.	Rappaport Hovav and Levin (2008)

(✓= acceptable, *=unacceptable)

Warstadt et al. (2020)

BART for Summarization

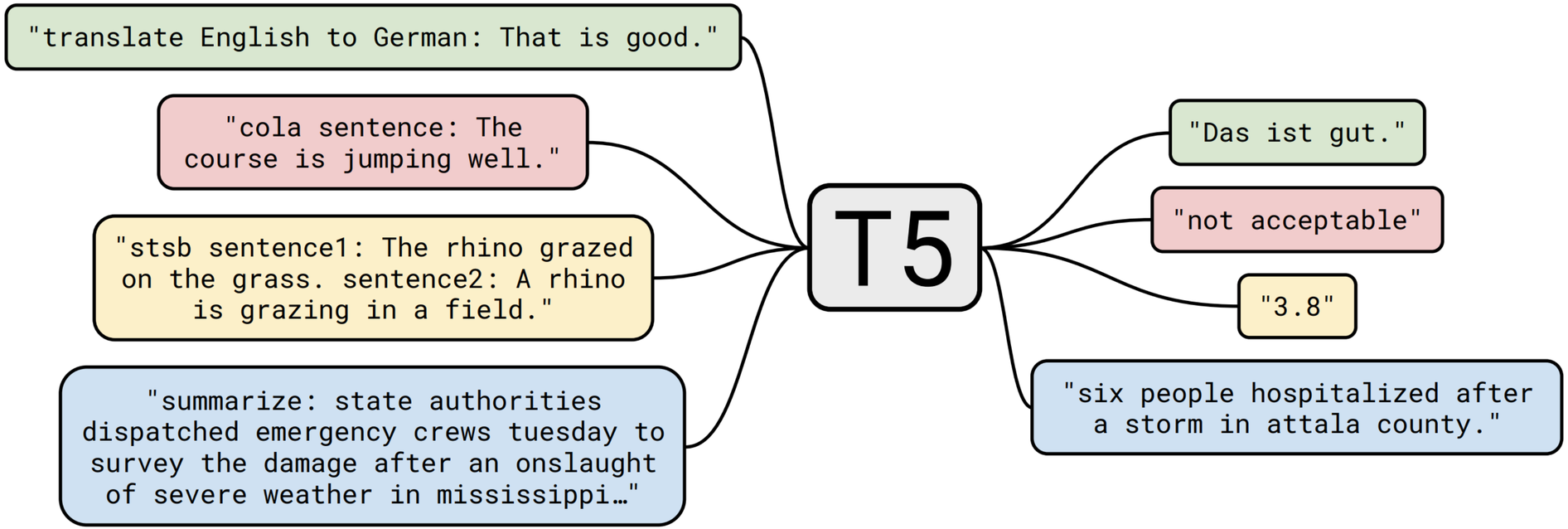
This is the first time anyone has been recorded to run a full marathon of 42.195 kilometers (approximately 26 miles) under this pursued landmark time. It was not, however, an officially sanctioned world record, as it was not an "open race" of the IAAF. His time was 1 hour 59 minutes 40.2 seconds. Kipchoge ran in Vienna, Austria. It was an event specifically designed to help Kipchoge break the two hour barrier.	Kenyan runner Eliud Kipchoge has run a marathon in less than two hours.
PG&E stated it scheduled the blackouts in response to forecasts for high winds amid dry conditions. The aim is to reduce the risk of wildfires. Nearly 800 thousand customers were scheduled to be affected by the shutoffs which were expected to last through at least midday tomorrow.	Power has been turned off to millions of customers in California as part of a power shutoff plan.

- But, strong results on dialogue, summarization, and other generation tasks.

Lewis et al. (2019)

T5

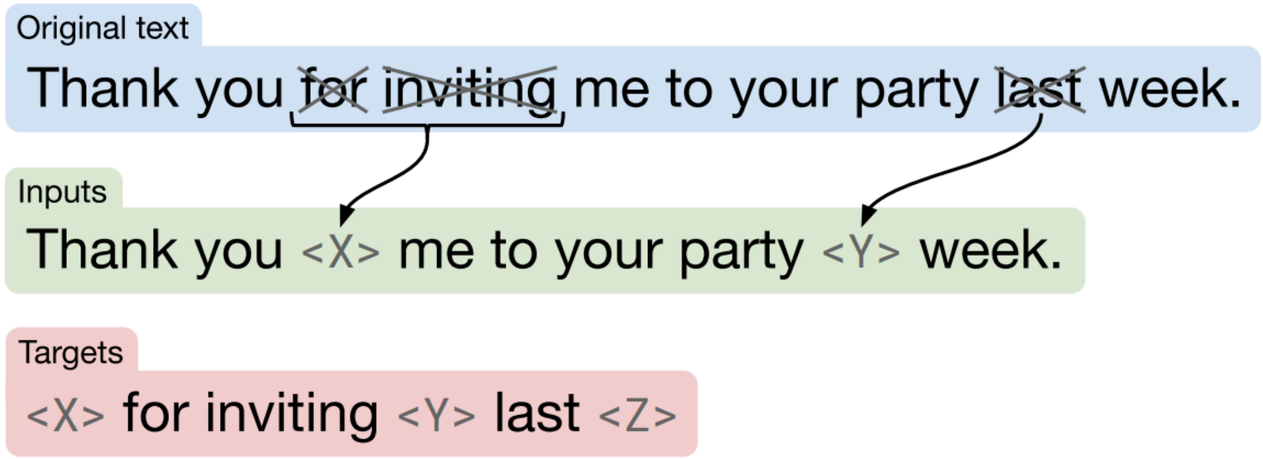
- Frame many problems as sequence-to-sequence ones:



Raffel et al. (2020)

T5

- Pre-training: similar denoising scheme to BART



- Different mask tokens for individual masked spans; also different format for targets

Raffel et al. (2020)

T5

- Compared several different unsupervised LM objectives:

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you <M> <M> me to your party apple week .	*(original text)*
Deshuffling	party me for your to . last fun you inviting week Thank	*(original text)*
MASS-style Song et al. (2019)	Thank you <M> <M> me to your party <M> week .	*(original text)*
I.i.d. noise, replace spans	Thank you <X> me to your party <Y> week .	<X> for inviting <Y> last <Z>
I.i.d. noise, drop tokens	Thank you me to your party week .	for inviting last
Random spans	Thank you <X> to <Y> week .	<X> for inviting me <Y> your party last <Z>

Raffel et al. (2020)

T5

Number of tokens	Repeats	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Full dataset	0	**83.28**	**19.24**	**80.88**	**71.36**	**26.98**	**39.82**	**27.65**
2^{29}	64	**82.87**	**19.19**	**80.97**	**72.03**	**26.83**	**39.74**	**27.63**
2^{27}	256	82.62	**19.20**	79.78	69.97	**27.02**	**39.71**	27.33
2^{25}	1,024	79.55	18.57	76.27	64.76	26.38	39.56	26.80
2^{23}	4,096	76.34	18.33	70.92	59.29	26.37	38.84	25.81

- Colossal Cleaned Common Crawl (C4): 750 GB of text
- We still haven't hit the limit of bigger data being useful for pre-training: here we see stronger MT results from the biggest data

Raffel et al. (2020)

Takeaways

- Transformers + lots of data + self-supervision seems to do very well
- Next time: GPT/GPT-2/GPT-3, etc.