Contextual Representations:
ELMo & BERT

Antoine Bosselut

EPFL

Announcements

- **Course Feedback:** Indicative Feedback this week! Please fill it out!
- **Assignment 1:** Due Friday, 03/24/2023 at 11:59 PM
  - Don’t wait until the last moment!
  - Additional Office Hours (today @ 11 AM; tomorrow)
- **Assignment 2:** Released Saturday, 03/25/2023 at 12:00 AM
- **Course Project:** Description to be released next week!

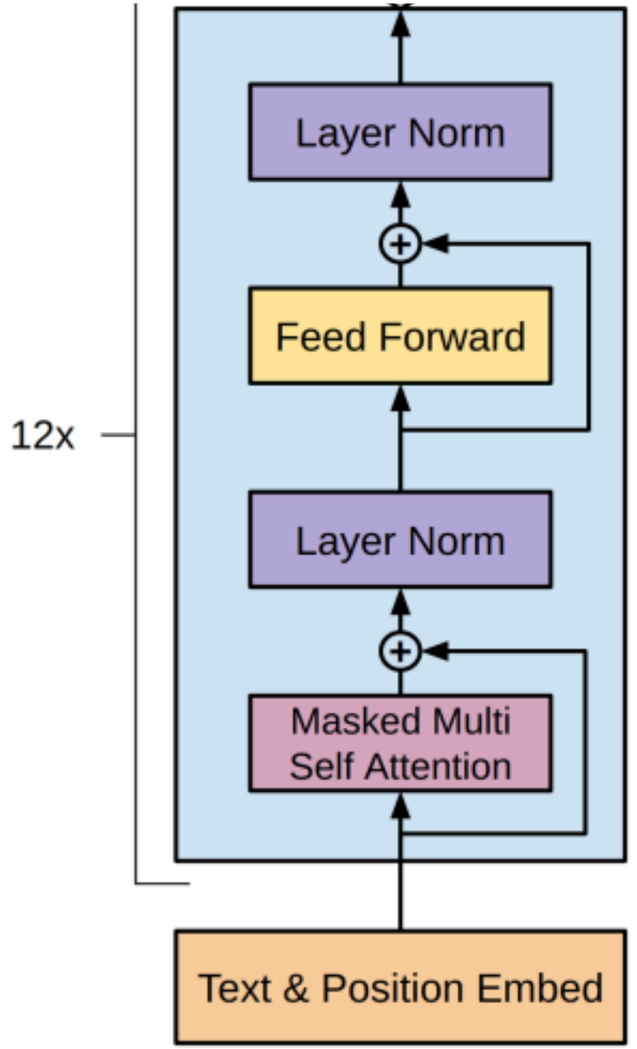
Today's Outline

- **Lecture**
  - **Quick Recap:** GPT
  - **Going Bidirectional:** ELMo + BERT
  - A1 Office Hours

Some slides adapted from Danqi Chen, Greg Durrett

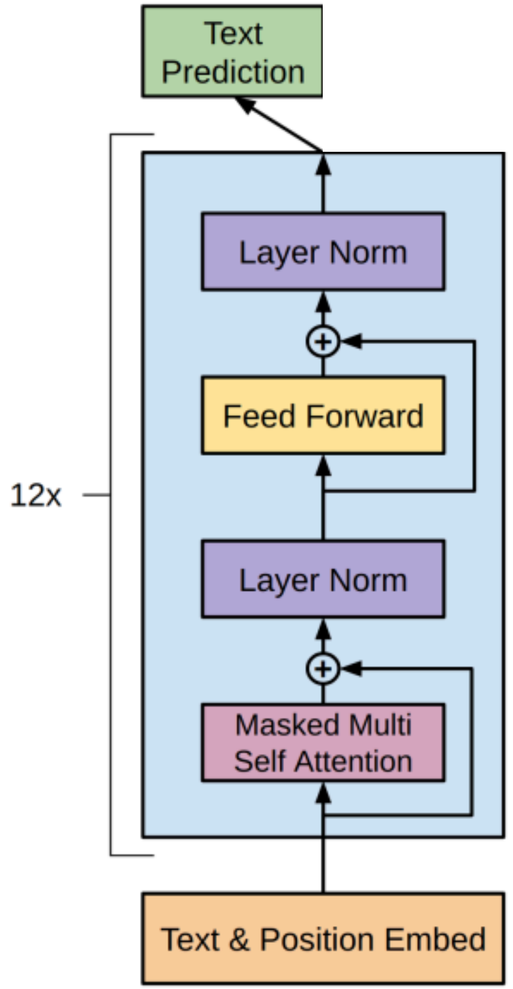
GPT: Generative Pretrained Transformer

- Called a *decoder* transformer
- **But**, actual GPT block mixes design of encoder and decoder from original transformer
- Uses masked multi-headed self-attention (decoder)
  - Can’t see future
- No cross-attention; only computes a self-attention over its history in each block (encoder)



Training GPT

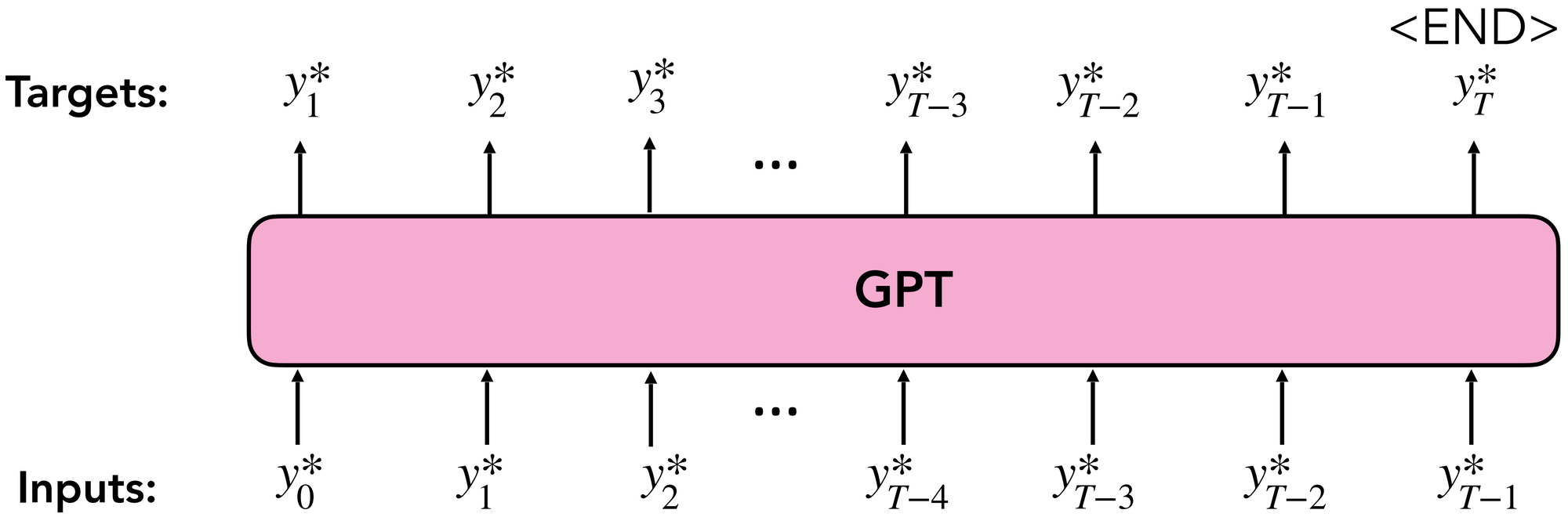
- Pretrained on TorontoBooks corpus: 7000 unpublished books (~13 GB)
- Corpus segmented broken up into windows of 512 tokens
  - can model long-range context during pretraining
- **Pretraining task:** next word prediction (i.e., language modelling)



Pretraining

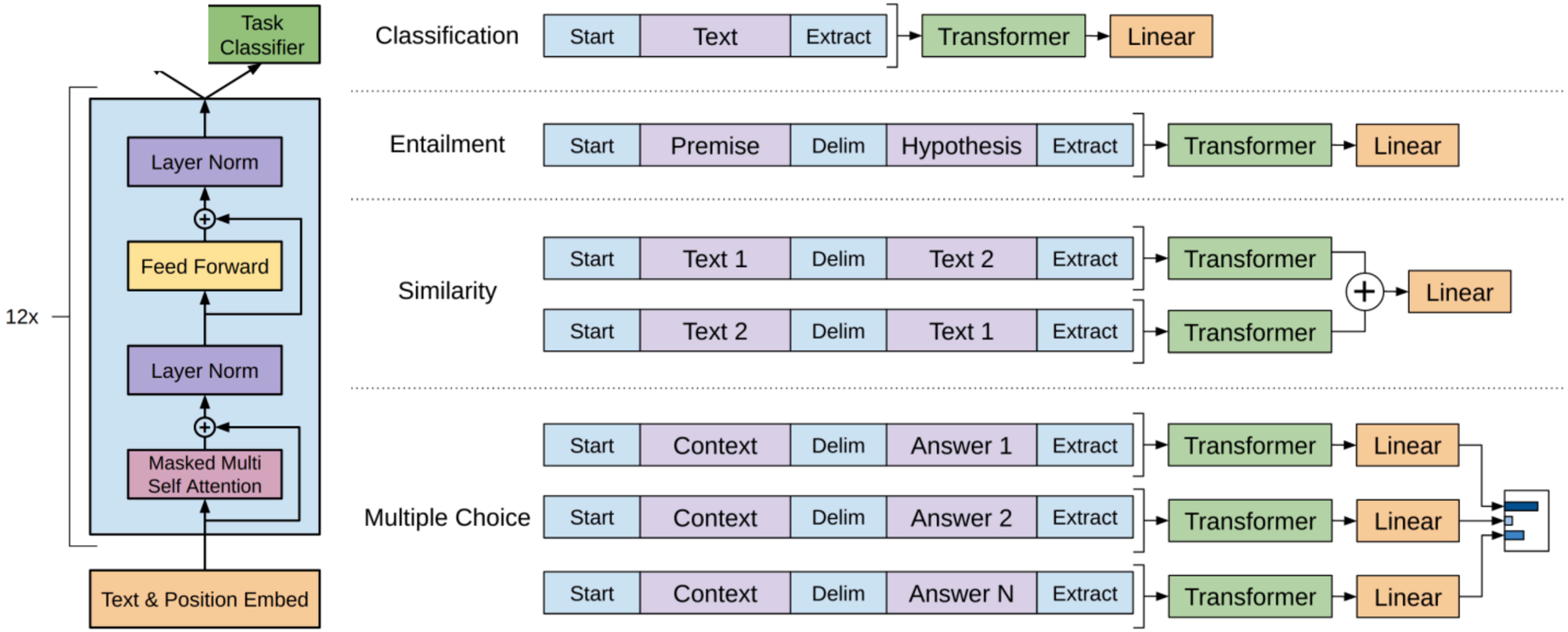
- Minimize the negative log probability of the gold* sequences in your dataset

$$\mathscr{L} = -\sum_{t=1}^{T} \log P\left(y_t^* \mid \{y_s^*\}_{s<t}\right)$$



Fine-tuning

- After pre-training, model can be fine-tuned by training on individual datasets
- Pretrained model used as initialisation for training on individual tasks



Massive Improvements (back then)

Dataset	Task	SOTA	Ours
SNLI	Textual entailment	89.3	89.9
MNLI matched	Textual entailment	80.6	82.1
MNLI mismatched	Textual entailment	80.1	81.4
SciTail	Textual entailment	83.3	88.3
QNLI	Textual entailment	82.3	88.1
RTE	Textual entailment	61.7	56.0
STS-B	Semantic similarity	81.0	82.0
QQP	Semantic similarity	66.1	70.3
MRPC	Semantic similarity	86.0	82.3
RACE	Reading comprehension	53.3	59.0
ROCStories	Commonsense reasoning	77.6	86.5
COPA	Commonsense reasoning	71.2	78.6
SST-2	Sentiment analysis	93.2	91.3
CoLA	Linguistic acceptability	35.0	45.4
GLUE	Multi task benchmark	68.9	72.8

https://openai.com/research/language-unsupervised

Question

Was GPT the first large-scale pretrained neural representation?

Word Embeddings: word2vec!

Question

What's an issue with word embeddings?

1) Chico Ruiz made a spectacular **play** on Alusik's grounder {. . . }

2) Olivia De Havilland signed to do a Broadway **play** for Garson {. . . }

3) Kieffer was commended for his ability to hit in the clutch , as well as his all-round excellent **play** {. . . }

4) {. . . } they were actors who had been handed fat roles in a successful **play** {. . . }

5) Concepts **play** an important role in all aspects of cognition {. . . }

Question

Words have different meanings in different contexts!

1) Chico Ruiz made a spectacular **play** on Alusik's grounder {. . . }

2) Olivia De Havilland signed to do a Broadway **play** for Garson {. . . }

3) Kieffer was commended for his ability to hit in the clutch , as well as his all-round excellent **play** {. . . }

4) {. . . } they were actors who had been handed fat roles in a successful **play** {. . . }

5) Concepts **play** an important role in all aspects of cognition {. . . }

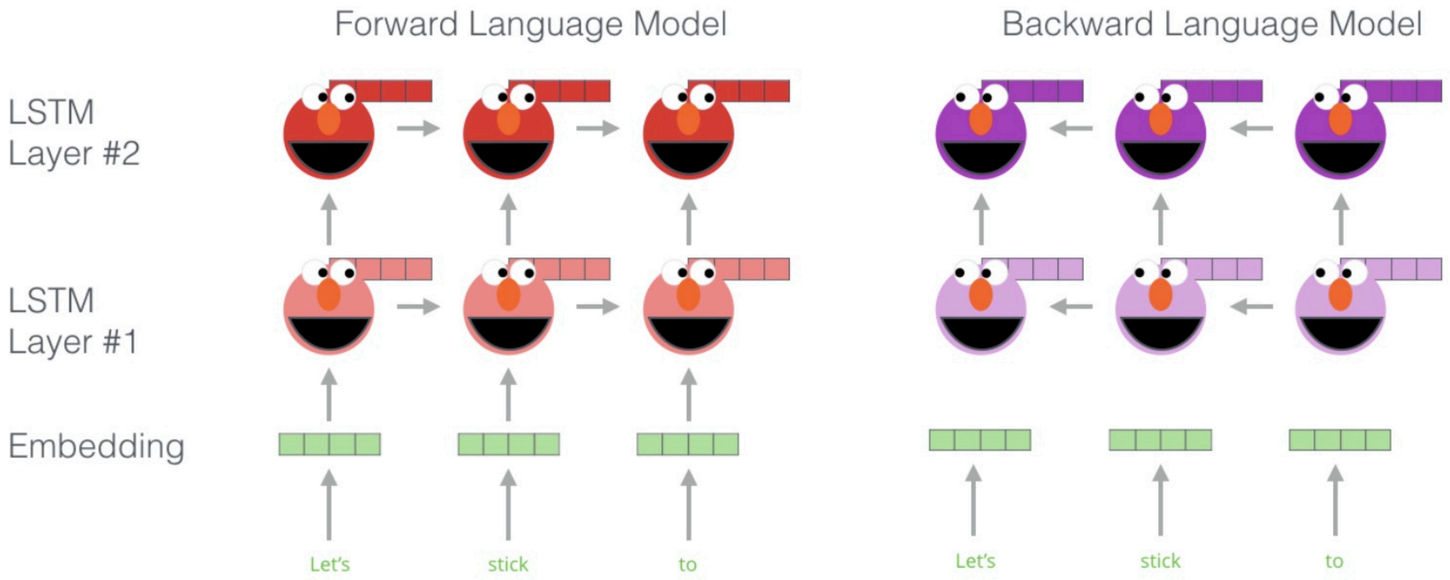
Question

How might we integrate contextual information into word representations?

2018: Use bidirectional LSTMs!

ELMo

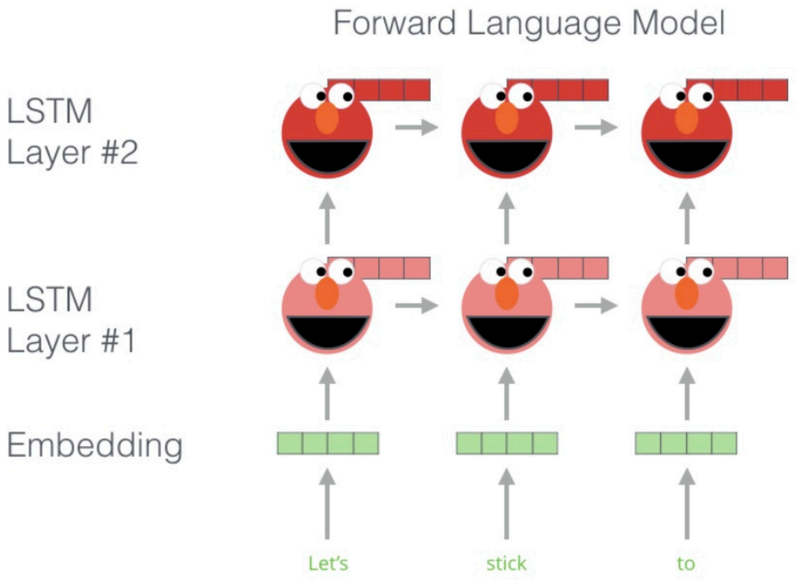
- Train two-layer LSTM-based language model on a **large** corpus
- Use **hidden states** of the LSTMs for each token to compute an embedding of each word
- LSTM should be bidirectional



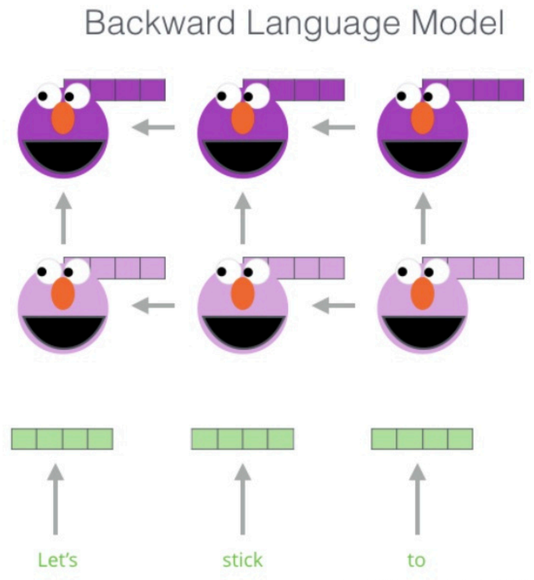
Peters et al. (2018)

ELMo

- Train two-layer LSTM-based language model on a **large** corpus
- Use **hidden states** of the LSTMs for each token to compute an embedding of each word
- LSTM should be bidirectional



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



Peters et al. (2018)

ELMo

- Train two-layer LSTM-based language model on a **large** corpus
- Use **hidden states** of the LSTMs for each token to compute an embedding of each word
- LSTM should be bidirectional
- Use 1B word benchmark (**single sentences** — why might this be a problem?)

Input Embeddings Forward LSTM

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

Output Embeddings

Backward LSTM

Peters et al. (2018)

Using ELMo Embeddings

At layer 0, this is just word embeddings

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

At higher layers, concatenation of biLSTM hidden states

- γ^{task}: allows the task model to scale the entire ELMo vector
- s_j^{task}: softmax-normalized weights across layers

Learn both of these parameters

Why average the representation at each layer as opposed to the final one?

Peters et al. (2018)

ELMo Improvements

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%

Across the board improvements over SOTA when introduced

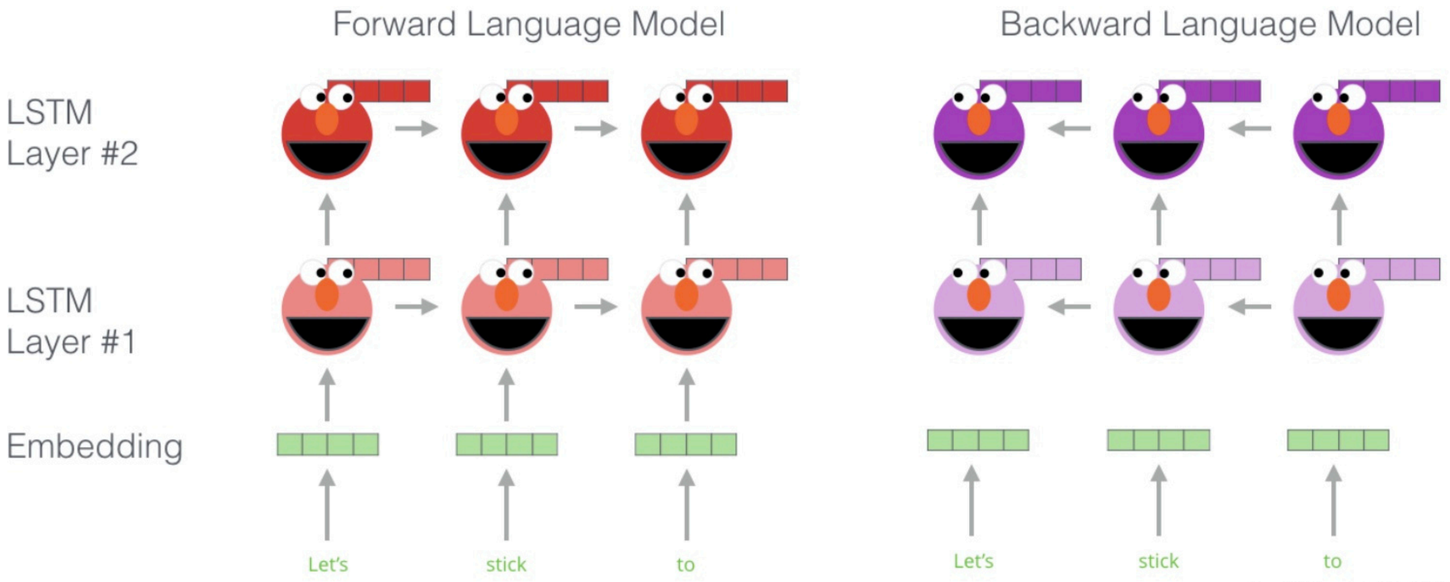
Peters et al. (2018)

Question

What's a problem with ELMo's notion of bidirectionality?

ELMo Issue

- ELMo (and bidirectional LSTMs / RNNs, in general) are **unidirectional** models masquerading as **bidirectional**
- Separate language model encodes the forward and backward sequence (and the representations are concatenated)



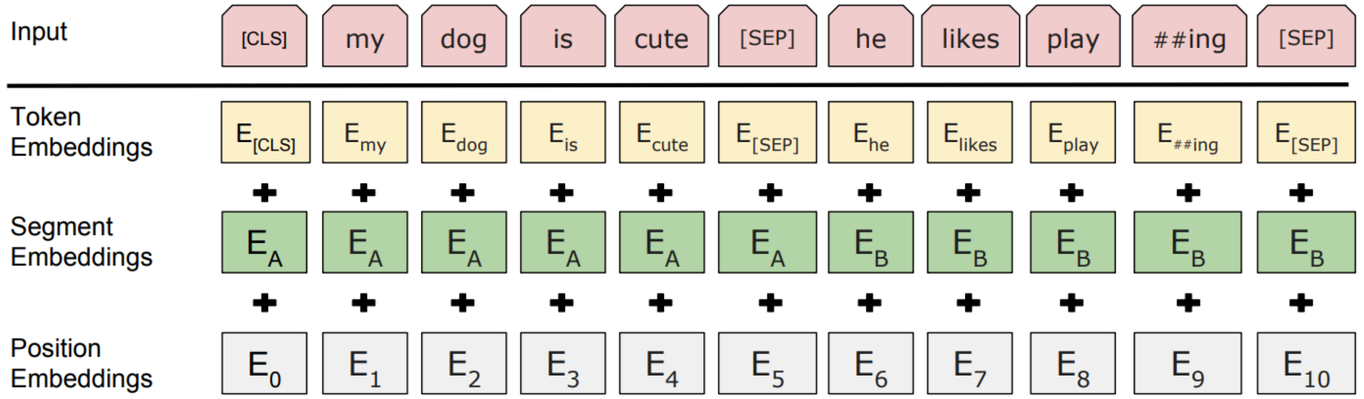
How can we fix this?

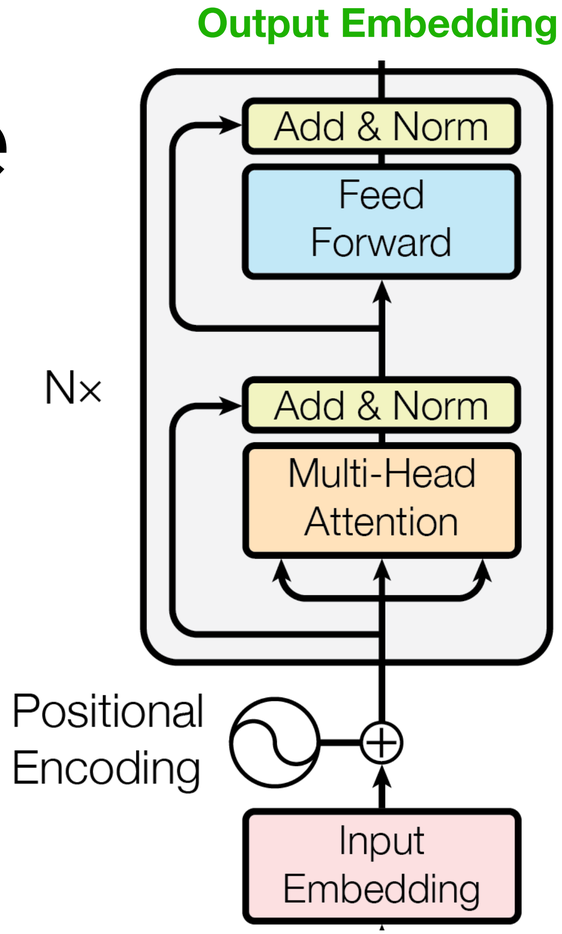
Peters et al. (2018)

What is BERT  ?

BERT Architecture

- **Transformer Encoder as we saw previously!**
- BERT-Base: 12 layers, *d*=768, 12 heads.
- Total params = **110M**
- BERT-Large: 24 layers, *d*=1024, 16 heads.
  Total params = **340M**





- Input embeddings for V = 30k word pieces
- Positional & segment embeddings

Devlin et al. (2019)

How is BERT trained?

Pretraining: Before

(Causal, Left-to-right) Language Modeling

I really enjoyed the movie we watched on ____

OpenAI

(Radford et al., 2018, 2019, many others)

Pretraining: Two Approaches

(Causal, Left-to-right) Language Modeling

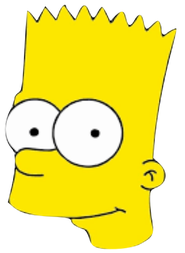
I really enjoyed the movie we watched on ____

OpenAI

(Radford et al., 2018, 2019, many others)

Masked Language Modeling

I really enjoyed the ____ we watched on Saturday!

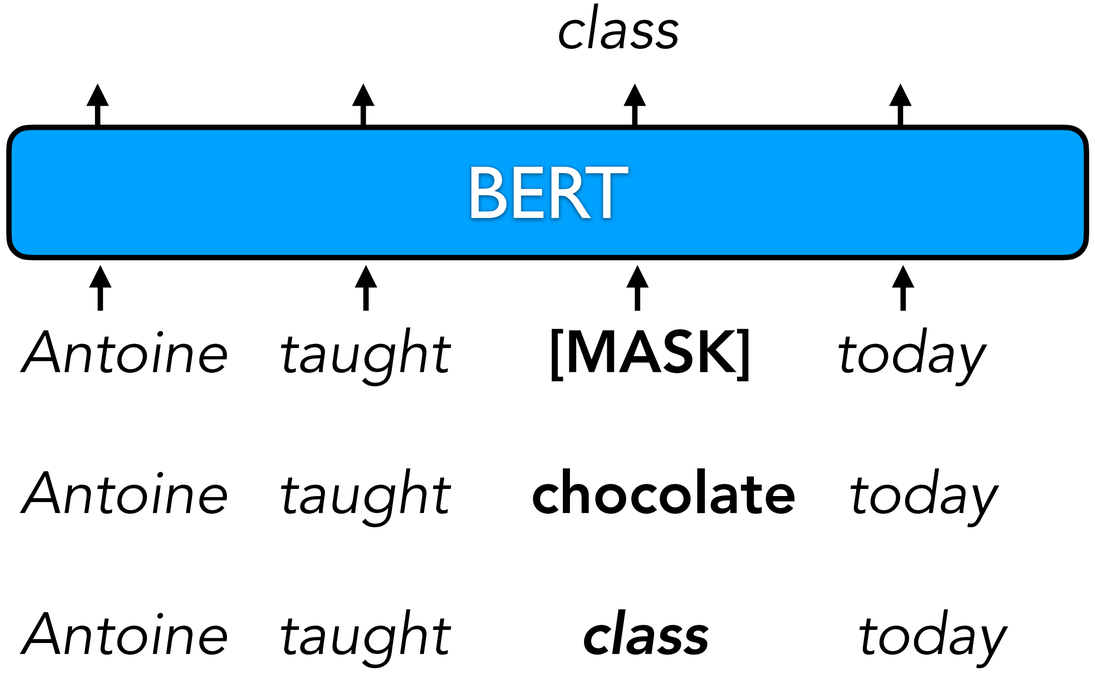


(Devlin et al., 2018; Liu et al., 2020)

Masked Language Modeling (BERT)

- **Training**: take a sequence of text, and predict 15% of the tokens

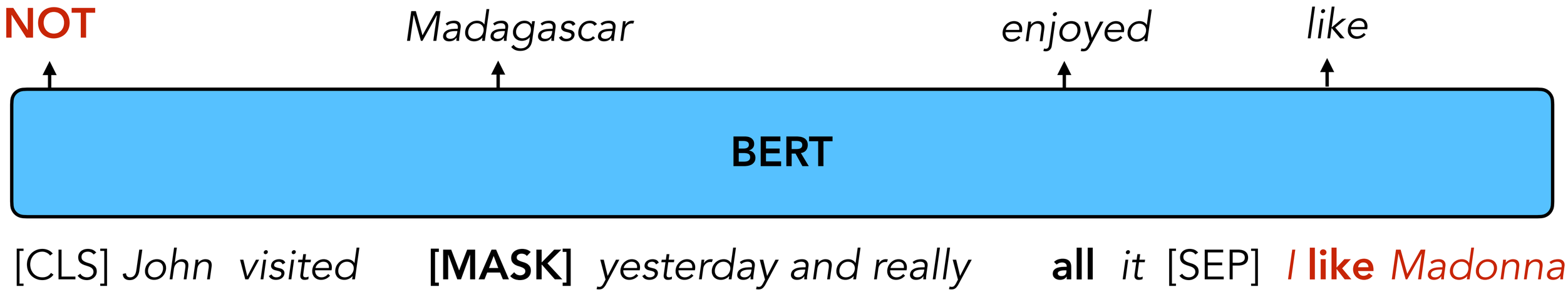
- **When predicting:**
  - Replace input token with [MASK] (80% of predictions)
  - Replace input token with a random token (10% of predictions)
  - Keep the same input token (10% of predictions)



Devlin et al. (2019)

Next "Sentence" Prediction (NSP)

- Input: **[CLS] Text Segment 1 [SEP]** **Text Segment 2**
- **Text Segment 2** = true text continuation (50%) or random text sequence (50%)
- Predict whether **Text Segment 2** is the actual continuation of **Text Segment 1**

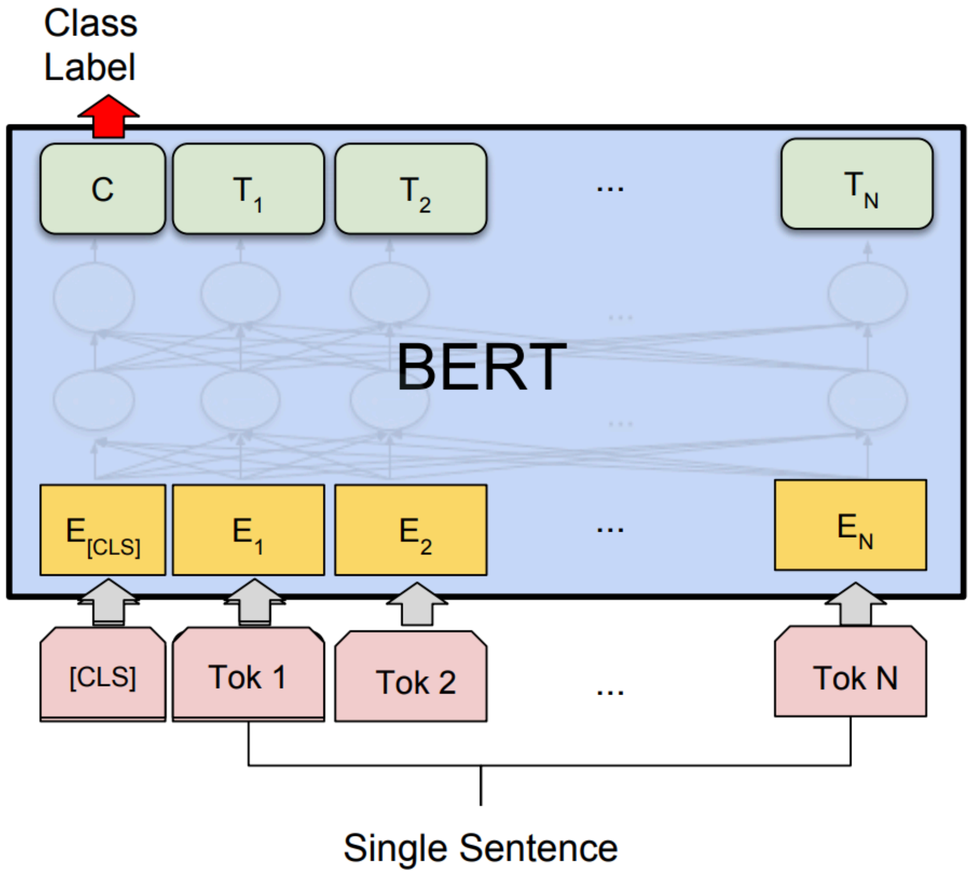


Follow-up work showed this objective didn't really matter

Devlin et al. (2019)

Fine-tuning BERT

Fine-tuning BERT for classification

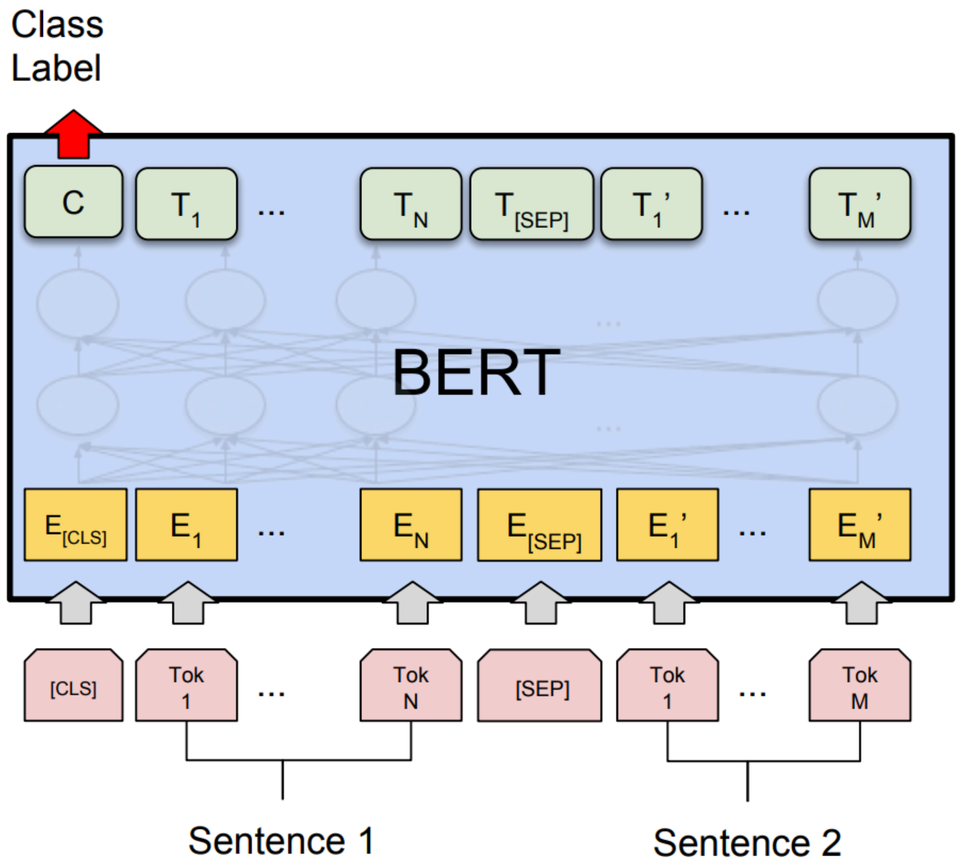


- A contextual embedding is outputted for each token in the input
- Prepend a special token **[CLS]** to the front of the sequence
- Classify the output embedding for this token

How do we classify the output embedding for this token?

Devlin et al. (2019)

Fine-tuning BERT for classification

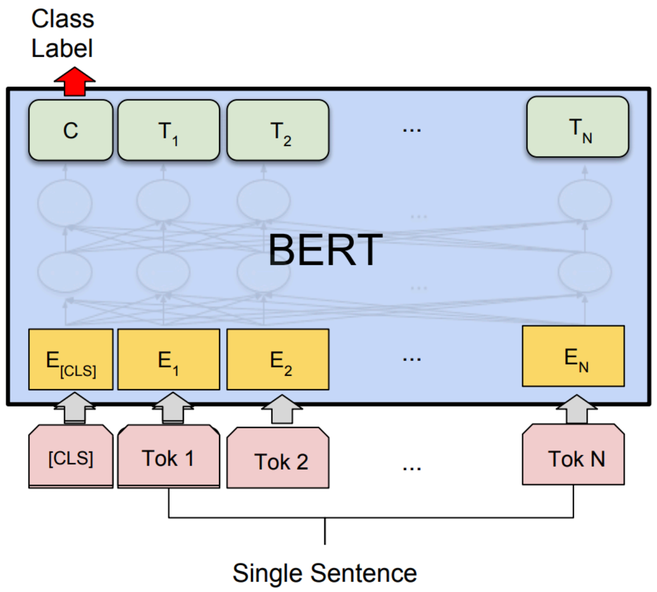


- A contextual embedding is outputted for each token in the input
- Prepend a special token **[CLS]** to the front of the sequence
- Classify the output embedding for this token
- Separate sequences with special **[SEP]** token

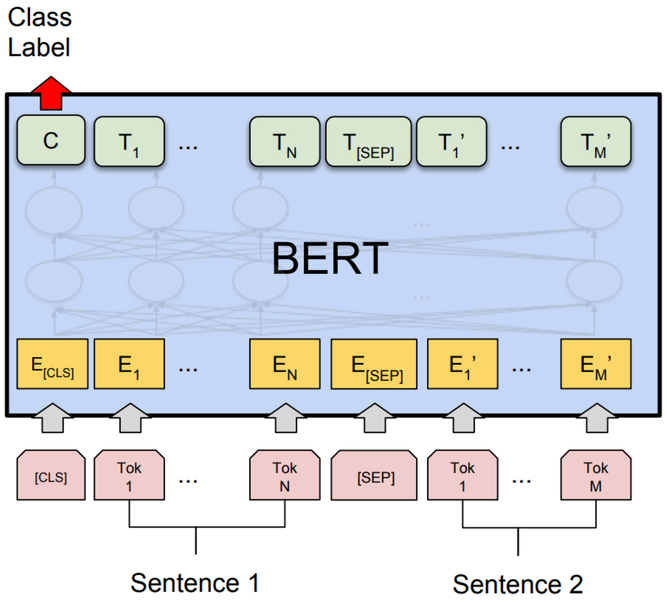
Can add special meta-tokens your vocabulary when they're needed!

Devlin et al. (2019)

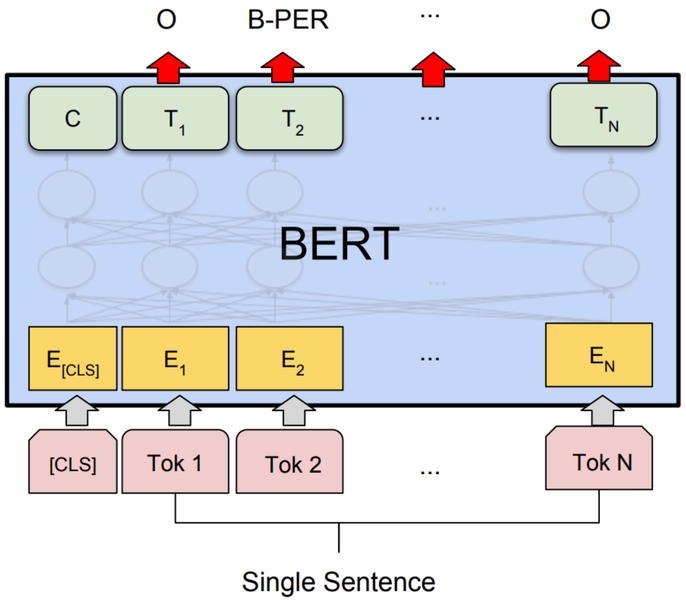
Fine-tuning a single model



(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

- Can use same model for classification tasks, sentence pair tasks, sequence labelling tasks, and many more!

Devlin et al. (2019)

Question

Why do we put the [CLS] token at the front of the sequence?

Easiest place to put it. Bidirectionality ensures it attends to representations of all other tokens

GLUE: Prototypical NLU Benchmark

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

Wang et al. (2019)

BERT on GLUE

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**

Performance increases across the board

Devlin et al. (2019)

BERT on GLUE

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**

Performance increases across the board

Big increase over OpenAI GPT highlights importance of bidirectionality

Devlin et al. (2019)

Ablation Studies

Tasks	Dev Set MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

Table 5: Ablation over the pre-training tasks using the $\text{BERT}_{\text{BASE}}$ architecture. "No NSP" is trained without the next sentence prediction task. "LTR & No NSP" is trained as a left-to-right LM without the next sentence prediction, like OpenAI GPT. "+ BiLSTM" adds a randomly initialized BiLSTM on top of the "LTR + No NSP" model during fine-tuning.

Importance of bidirectionality

Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

Table 6: Ablation over BERT model size. #L = the number of layers; #H = hidden size; #A = number of attention heads. "LM (ppl)" is the masked LM perplexity of held-out training data.

More parameters = better!

Devlin et al. (2019)

Question

Should we use BERT to generate embeddings or fine-tune the full model?

Fine-tuning vs. Embeddings

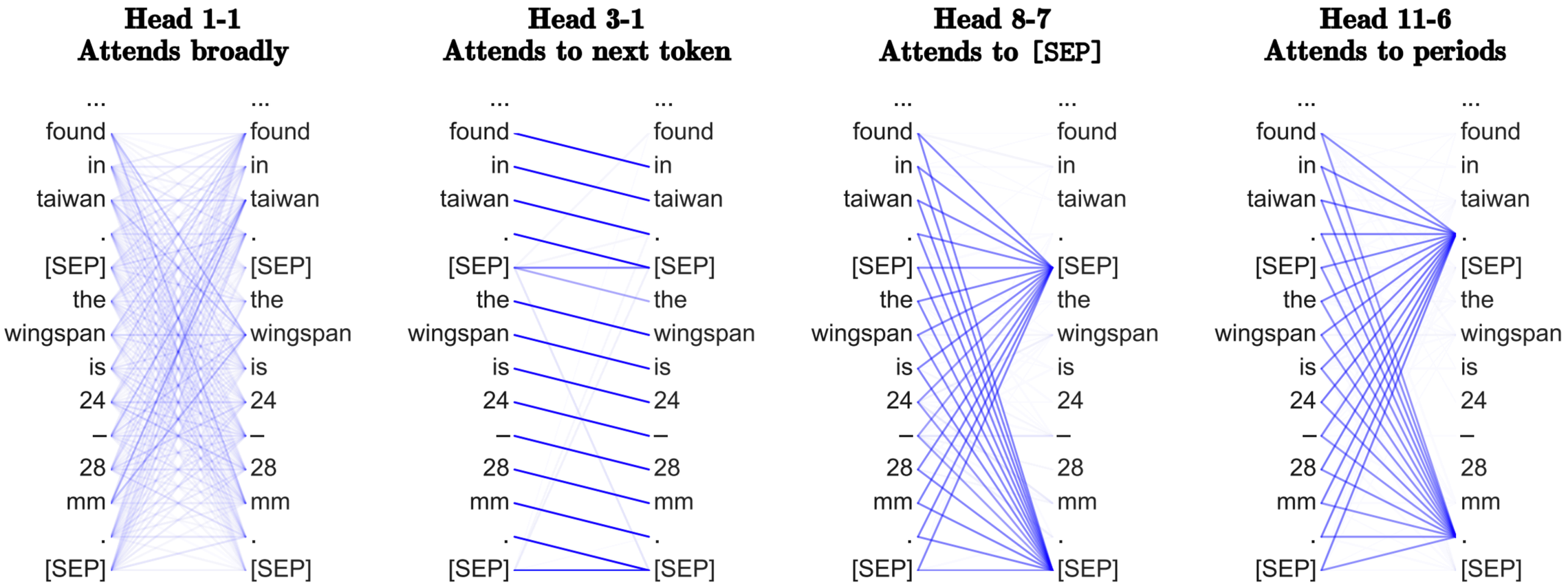
Pretraining	Adaptation	NER CoNLL 2003	SA SST-2	Nat. lang. inference MNLI	Nat. lang. inference SICK-E	Semantic textual similarity SICK-R	Semantic textual similarity MRPC	Semantic textual similarity STS-B
Skip-thoughts	❄️	-	81.8	62.9	-	86.6	75.8	71.8
ELMo	❄️	91.7	**91.8**	**79.6**	**86.3**	**86.1**	**76.0**	**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 outputs embeddings that can be frozen and provided to a different model, but BERT performs better when fully fine-tuned

Peters et al., 2019

What does BERT Learn?

What does BERT learn?



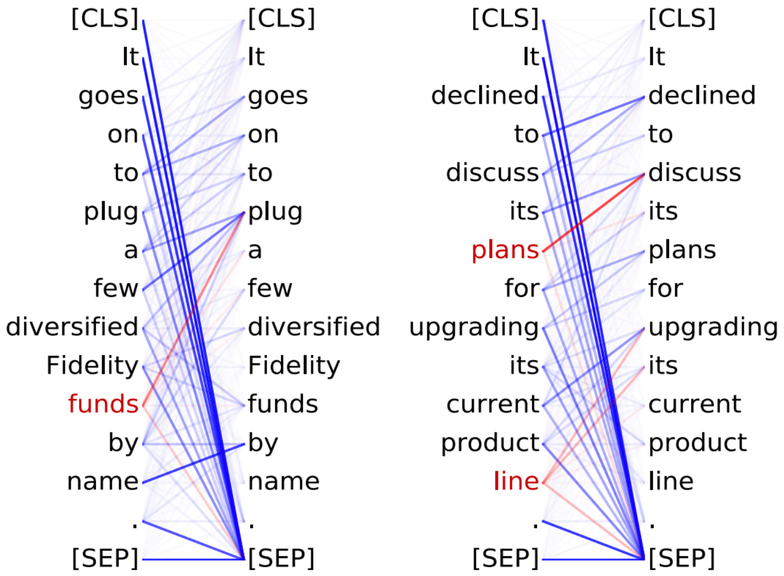
Transformer heads learn diverse concepts that map to positional, semantic, and syntactic relationships

Clark et al. (2019)

What does BERT learn?

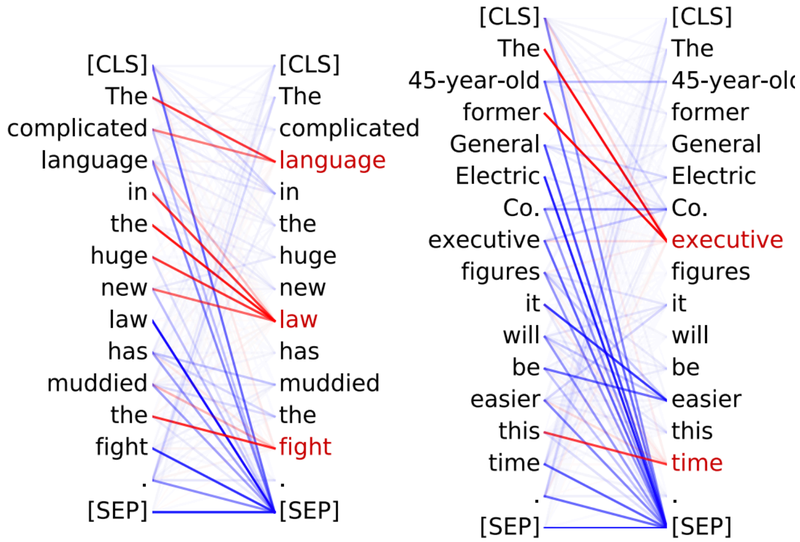
Head 8-10

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



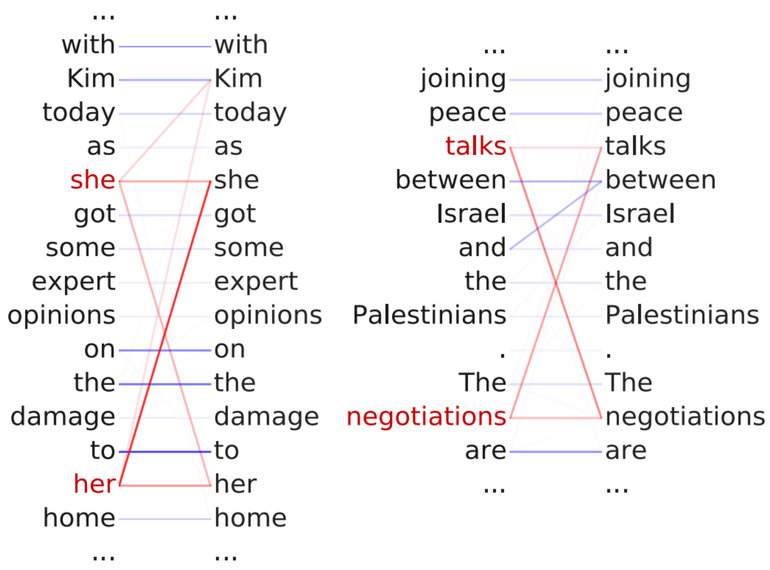
Head 8-11

- **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



Transformer heads learn diverse concepts that map to positional, semantic, and syntactic relationships

Clark et al. (2019)

Improvements to BERT

Whole word masking

[MASK] _**bama** _was _the _president _of _the _United _States _in _2010

vs.

[MASK] [MASK] _was _the _president _of _the _United _States _in _2010

RoBERTa

- "Robustly Optimised BERT" — a collection of improvements to BERT
- Same architecture as BERT
- 160 GB of training data, rather than only 16 GB in BERT

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

Liu et al. (2019)

DistilBERT

- Do we need all parameters of BERT, which require lots of storage?
- What if BERT was a much smaller?
- Train with distillation over soft target probabilities of BERT (and MLM)

$$\mathscr{L}_{distil} = -P_{BERT}\left(y_t^* \mid [\mathsf{M}], \{y_s^*\}_{s \neq t}\right) \log P\left(y_t^* \mid [\mathsf{M}], \{y_s^*\}_{s \neq t}\right)$$

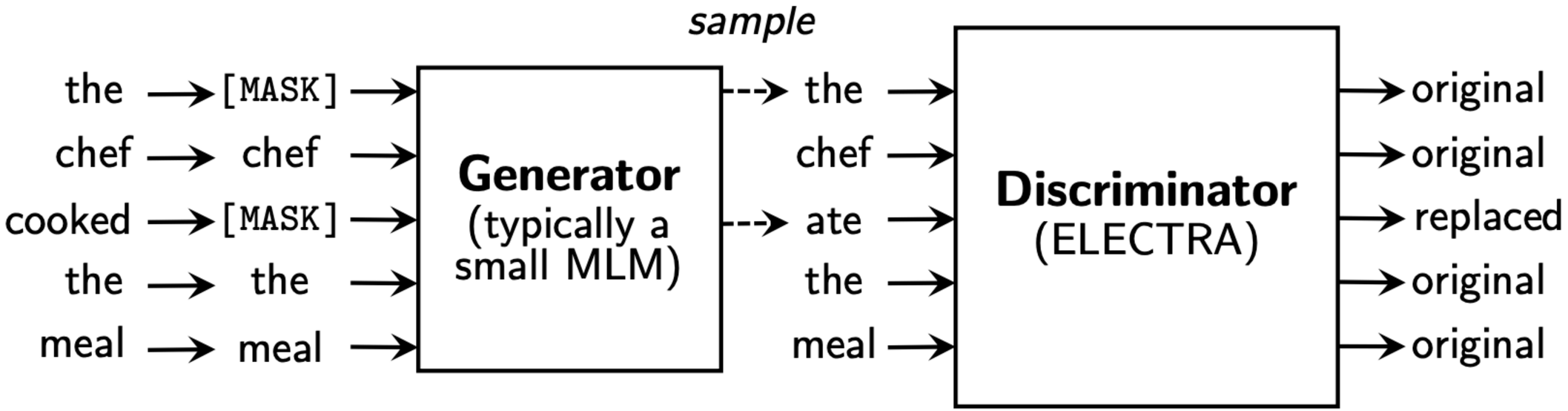
$$\mathscr{L}_{mlm} = -\log P\left(y_t^* \mid [\mathsf{M}], \{y_s^*\}_{s \neq t}\right)$$

$$\mathscr{L}_{tot} = \gamma_1 \mathscr{L}_{distil} + \gamma_1 \mathscr{L}_{mlm}$$

- Allows you to train much smaller DistilBERT with ~97% performance of BERT

Sanh et al. (2019)

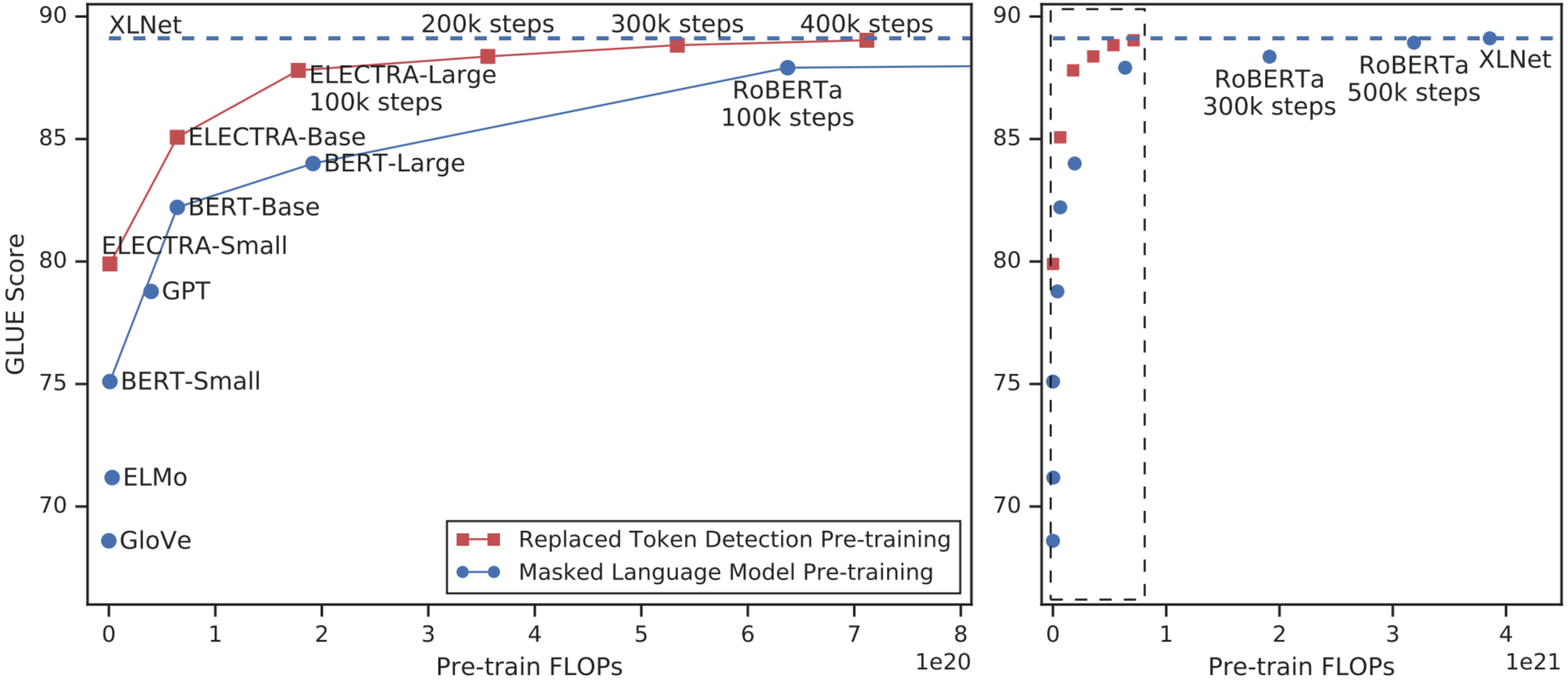
ELECTRA



- **Recall:** BERT only learns from 15% of masked tokens **(quite inefficient!)**
- Instead, predict whether a token is corrupted on the discriminator
- Learning from all tokens drastically speeds up training

Clark et al. (2019)

ELECTRA


90
85
80
75
70
GLUE Score
XLNet
200k steps
300k steps
400k steps
ELECTRA-Large
100k steps
ELECTRA-Base
BERT-Large
BERT-Base
ELECTRA-Small
GPT
BERT-Small
ELMo
GloVe
RoBERTa
100k steps
Replaced Token Detection Pre-training
Masked Language Model Pre-training
0
1
2
3
4
5
6
7
8
1e20
Pre-train FLOPs
RoBERTa
300k steps
RoBERTa
500k steps
XLNet
0
1
2
3
4
1e21
Pre-train FLOPs

Question

What was the main improvement of BERT-style models over GPT?

Bidirectionality allows for more expressive representations to be learned

Question

What can BERT NOT due as a result of its masked LM training objective?

Generate text!

Recap

- **Contextualised embeddings:** Let us model words and sequences conditioned on the context around them
- **ELMo:** One of the first models for contextualized embeddings based on bidirectional LSTMs
- **GPT**: First model for contextualised embeddings using transformer models
- **BERT**: Improving transformer-based contextual representations using masked language models and bidirectional encoding
- Many variants of BERT in recent years!
- Much work on analysing information learned in contextual representations (Week 11)

Looking Forward

- **Tomorrow**
  - **Lecture:** Encoder-Decoder Transformers - BART + T5
  - A1 Office Hours
  - **Review of Week 4 Exercise Session:** Transformers + Decoding
  - **Week 5 Exercise Session:** Transformers + Decoding

References

- Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. *North American Chapter of the Association for Computational Linguistics*.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.
- Clark, K., Khandelwal, U., Levy, O., & Manning, C. D. (2019). What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.