Boulder

Natural Language Processing; Language Modeling

YouTube Playlist

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu

Deep contextualized word representations

YouTube Video

ELMo: Embeddings from Language Models

$(t_1, t_2, \ldots, t_N) \rightarrow$ sequence of N tokens

Forward Language Model

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_1, t_2, \ldots, t_{k-1})$$

$x_k^{LM} \rightarrow$ context-independent token representation

$x_k^{LM} \triangleright$ L layers of forward LSTMs

$\overrightarrow{h}_{k,j}^{LM}, j = 1, \ldots, L$

$\overrightarrow{h}_{k,L}^{LM} \triangleright$ softmax $\rightarrow$ predict next token t_{k+1}

Backward LM

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_{k+1}, t_{k+2}, \ldots, t_N)$$

$\overleftarrow{h}_{k,j}^{LM}, j = 1, \ldots, L$

Bidirectional LM

token softmax

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

ELMo

$$\begin{aligned} R_k &= \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L\} \\ &= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L\}, \end{aligned}$$

$$h_{k,0}^{Lm} = x_k^{LM}$$

$$h_{k,j}^{LM} = [\overrightarrow{h}_{k,j}^{LM}; \overleftarrow{h}_{k,j}^{LM}]$$

$$\text{ELMo}_k = E(R_k; \Theta_e)$$

$$E(R_k) = h_{k,L}^{LM}$$

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

$s_j^{\text{task}} \rightarrow$ softmax-normalized weights

$[x_k; \text{ELMo}_k^{\text{task}}] \rightarrow$ input to task RNN

$[h_k; \text{ELMo}_k^{\text{task}}] \rightarrow$ output of task RNN

Downstream Tasks

1. Question Answering
2. Textual Entailment
   Determine whether a "hypothesis" is true given a "premise"
3. Semantic Role Labeling
   who, did what, to whom, at where
4. Coreference Resolution
   clustering mentions in the text that refer to the same underlying real world entities
5. Named Entity Extraction
   (e.g., entity types PER, LOC, ORG, MISC
6. Sentiment Analysis

Task	Previous SOTA			Our baseline	ELMo + baseline	Increase (absolute/ relative)
Stanford Question Answering Dataset ← SQuAD	Liu et al. (2017)	F_1	84.4	81.1	85.8	4.7 / 24.9%
Stanford Natural Language Inference ← SNLI	Chen et al. (2017)	accuracy	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
Semantic Role Labeling ← SRL	He et al. (2017)	F_1	81.7	81.4	84.6	3.2 / 17.2%
Coreference Resolution ← Coref	Lee et al. (2017)	average F_1	67.2	67.2	70.4	3.2 / 9.8%
Named Entity Recognition ← NER	Peters et al. (2017)	F_1	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
Stanford Sentiment Treebank ← SST-5	McCann et al. (2017)	accuracy	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {…}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {…}	{…} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

Peters, Matthew E., et al. "Deep contextualized word representations." *arXiv preprint arXiv:1802.05365* (2018).

An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling

Sequence Modeling

$x_0, \ldots, x_T \rightarrow$ input sequence

$y_0, \ldots, y_T \rightarrow$ output sequence

$f : \mathcal{X}^{T+1} \rightarrow \mathcal{Y}^{T+1}$

$\hookrightarrow$ sequence modeling network

$\hat{y}_0, \ldots, \hat{y}_T = f(x_0, \ldots, x_T)$

Causal Constraint

y_t depends only on $x_0, \ldots, x_t$ and not on any "future" inputs $x_{t+1}, \ldots, x_T$

$L(y_0, \ldots, y_T, f(x_0, \ldots, x_T)) \rightarrow$ loss

1D fully-convolutional network (FCN)

zero padding

causal convolutions

TCN = 1D FCN + causal convolutions

$\hookrightarrow$ temporal convolutional network

Dilated Convolutions

longer history

$x \in \mathbb{R}^n \rightarrow$ 1D sequence input

$f : \{0, \ldots, k-1\} \rightarrow \mathbb{R}$

$\hookrightarrow$ filter

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) x_{s-di}$$

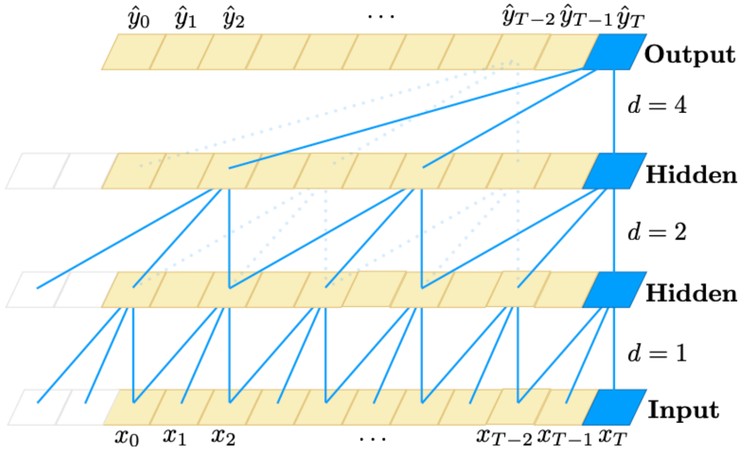
$d \rightarrow$ dilation factor

$k \rightarrow$ filter size

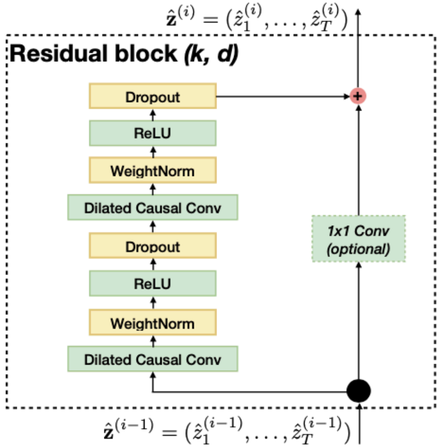
$s - di \rightarrow$ direction of the past

$(k-1)d \rightarrow$ effective history of such a layer

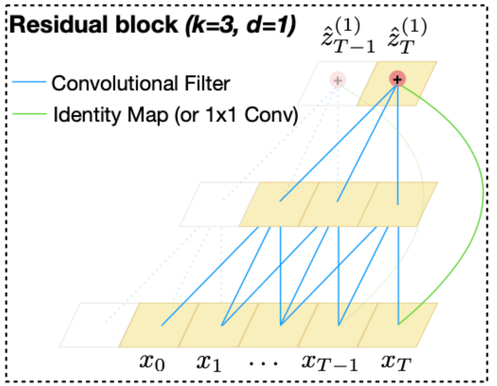
$d = O(2^i)$ at level i of the network



(a)



(b)



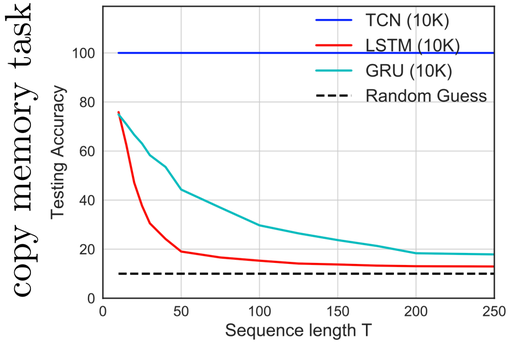
(c)

Sequence Modeling Task	Model Size (≈)	Models			
		LSTM	GRU	RNN	TCN
Seq. MNIST (accuracy^h)	70K	87.2	96.2	21.5	**99.0**
Permuted MNIST (accuracy)	70K	85.7	87.3	25.3	**97.2**
Adding problem T=600 (loss$^\ell$)	70K	0.164	**5.3e-5**	0.177	**5.8e-5**
Copy memory T=1000 (loss)	16K	0.0204	0.0197	0.0202	**3.5e-5**
Music JSB Chorales (loss)	300K	8.45	8.43	8.91	**8.10**
Music Nottingham (loss)	1M	3.29	3.46	4.05	**3.07**
Word-level PTB (perplexity$^\ell$)	13M	**78.93**	92.48	114.50	88.68
Word-level Wiki-103 (perplexity)	-	48.4	-	-	**45.19**
Word-level LAMBADA (perplexity)	-	4186	-	14725	**1279**
Char-level PTB (bpc$^\ell$)	3M	1.36	1.37	1.48	**1.31**
Char-level text8 (bpc)	5M	1.50	1.53	1.69	**1.45**

Residual Connections

$o = \text{Activation}(x + \mathcal{F}(x))$

JSB Chorales: Each input is a sequence of elements. Each element is an 88-bit binary code that corresponds to the 88 keys on a piano, with 1 indicating a key that is pressed at a given time.



Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." *arXiv preprint arXiv:1803.01271* (2018).

Improving Language Understanding by Generative Pre-Training

YouTube Video

Unsupervised pre-training

$\mathcal{U} = \{u_1, u_2, \ldots, u_n\} \rightarrow$ unsupervised corpus of tokens

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

$k \rightarrow$ size of the context window

Transformer decoder

$U = (u_{-k}, \ldots, u_{-1}) \rightarrow$ context vector of tokens

$$h_0 = UW_e + W_p$$

$W_p \rightarrow$ position embedding matrix

$W_e \rightarrow$ token embedding matrix

$$h_l = \texttt{transformer\_block}(h_{l-1}), l = 1, \ldots, L$$

$L \rightarrow$ number of layers

$$P(u) = \texttt{softmax}(h_L W_e^T)$$

Supervised fine-tuning

$\mathcal{C} \rightarrow$ labeled dataset

$x^1, \ldots, x^m \rightarrow$ input tokens

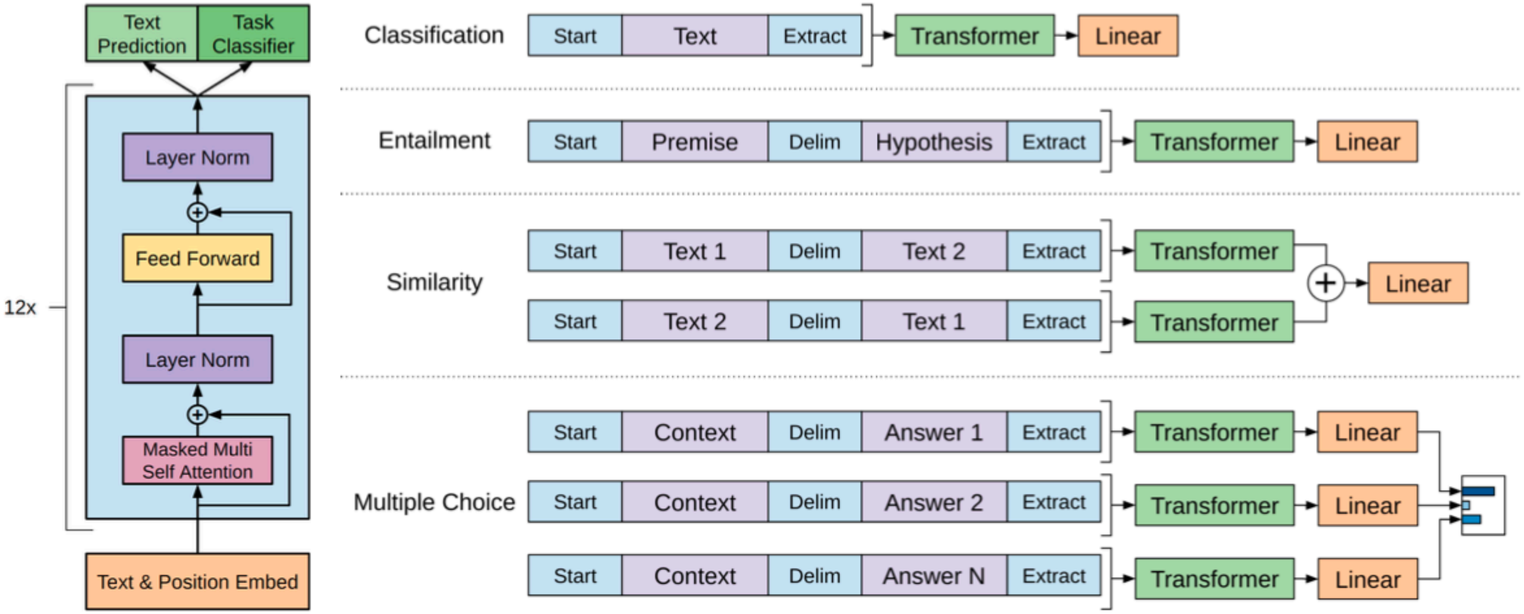
$y \rightarrow$ label

$h_L^m \rightarrow$ final transformer block's activation

$$p(y | x^1, \ldots, x^m) = \text{softmax}(h_L^m W_y)$$

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y | x^1, \ldots, x^m)$$

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$



Task	Datasets
Natural language inference	SNLI [5], MultiNLI [66], Question NLI [64], RTE [4], SciTail [25]
Question Answering	RACE [30], Story Cloze [40]
Sentence similarity	MSR Paraphrase Corpus [14], Quora Question Pairs [9], STS Benchmark [6]
Classification	Stanford Sentiment Treebank-2 [54], CoLA [65]

Natural Language Inference `entailment`, `contradiction` or `neutral`

image captions (SNLI)

transcribed speech, popular fiction, and government reports (MNLI)

Wikipedia articles (QNLI)

science exams (SciTail)

news articles (RTE)

Corpus of Linguistic Acceptability (CoLA)

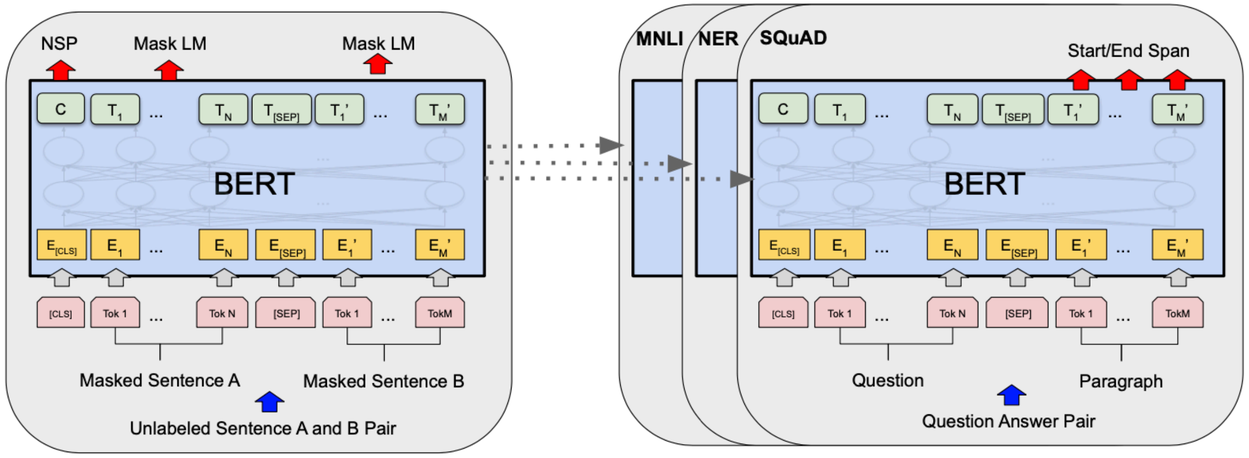
Semantic Textual Similarity (STS-B)

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018): 12.

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

YouTube Playlist

Bidirectional **E**ncoder **R**epresentations from **T**ransformers

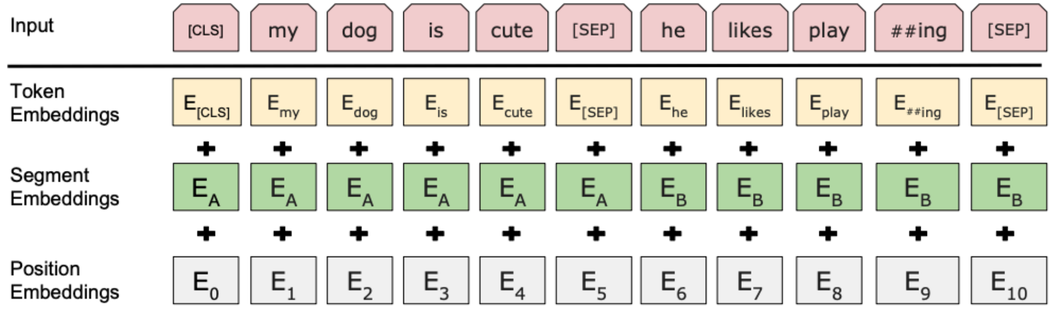


Task 1: Masked LM

mask 15% of all WordPiece tokens in each sequence at random

Task 2: Next Sentence Prediction (NSP)

IsNext vs NotNext



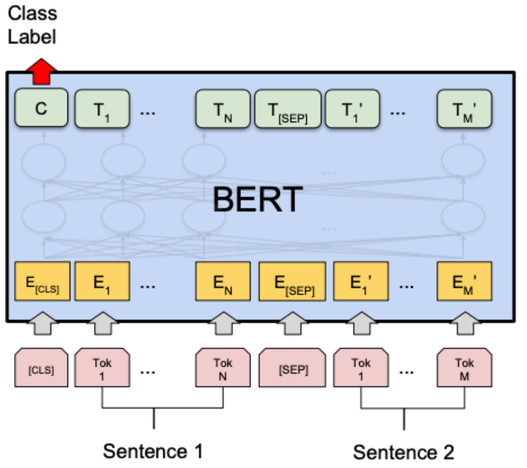
$S \in \mathbb{R}^H \rightarrow$ start

$E \in \mathbb{R}^H \rightarrow$ end

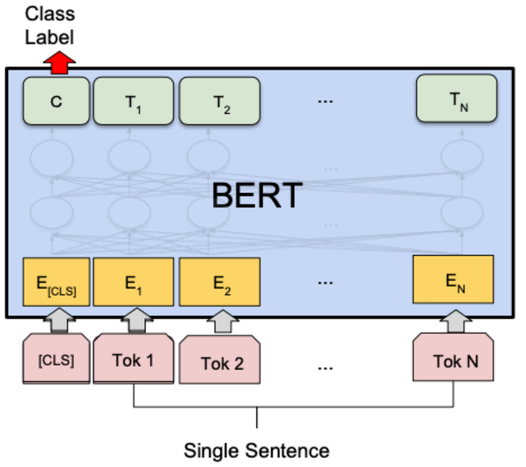
$$P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$$

$S \cdot T_i + E \cdot T_j$

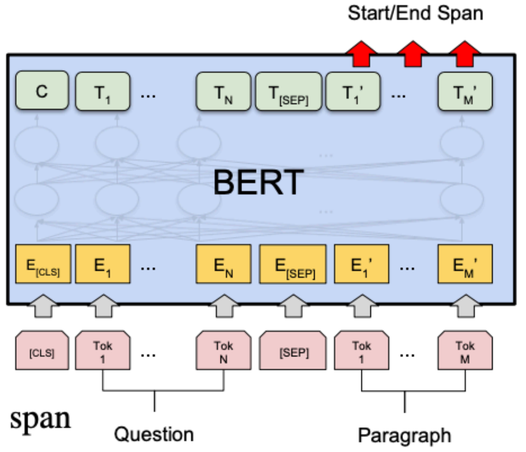
maximum scoring span



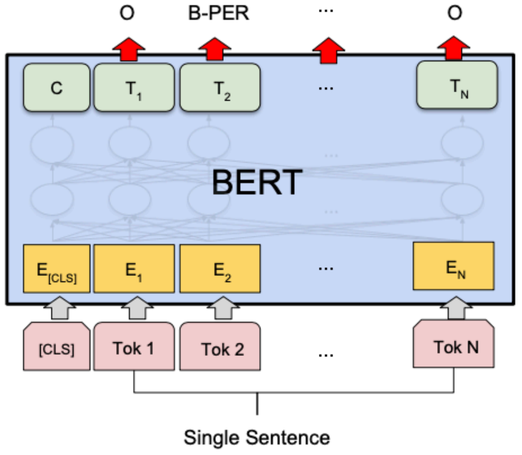
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



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



(c) Question Answering Tasks: SQuAD v1.1



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

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

Language Models are Unsupervised Multitask Learners

YouTube Video

– Question Answering
– Machine Translation
– Reading Comprehension
– Summarization

GPT-2 → a 1.5B parameter Transformer

$(x_1, x_2, \ldots, x_n)$ → set of examples

$x = (s_1, s_2, \ldots, s_n)$ → variable-length sequence of symbols

$$p(x) = \prod_{i=1}^{n} p(s_n|s_1, \ldots, s_{n-1})$$

$$p(s_{n-k}, \ldots, s_n|s_1, \ldots, s_{n-k-1})$$

$$p(output|input)$$

$$p(output|input, task)$$

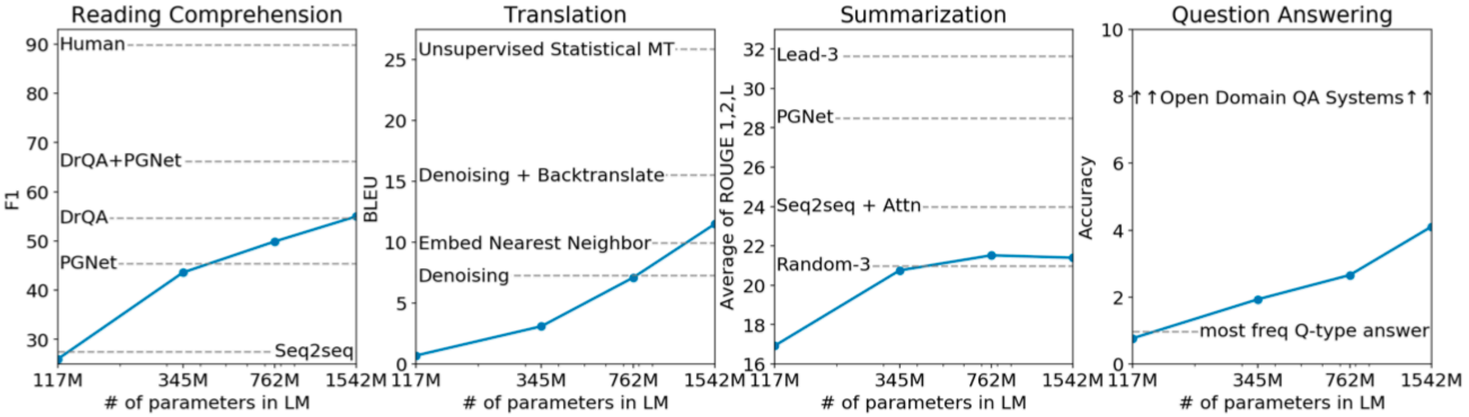
Task conditioning: 1. architectural level
2. algorithmic level
(e.g., Model-Agnostic Meta-Learning)

translation training example
`(translate to french, english text, french text)`

reading comprehension training example
`(answer the question, document, question, answer)`

pre-processing steps: lower-casing,
tokenization, and out-of-vocabulary tokens

Byte Pair Encoding (BPE): a practical middle ground
between character and word level language modeling



"I'm not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I'm not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: "**Mentez mentez, il en restera toujours quelque chose**," which translates as, "**Lie lie and something will always remain.**"

"I hate the word '**perfume**,"' Burr says. 'It's somewhat better in French: '**parfum**.'

If listened carefully at 29:55, a conversation can be heard between two guys in French: "**-Comment on fait pour aller de l'autre coté? -Quel autre coté?**", which means "**- How do you get to the other side? - What side?**".

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

"**Brevet Sans Garantie Du Gouvernement**", translated to English: "**Patented without government warranty**".

Table 1. Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.

Context (passage and previous question/answer pairs)

Tom goes everywhere with Catherine Green, a 54-year-old secretary. He moves around her office at work and goes shopping with her. "Most people don't seem to mind Tom," says Catherine, who thinks he is wonderful. "He's my fourth child," she says. She may think of him and treat him that way as her son. He moves around buying his food, paying his health bills and his taxes, but in fact Tom is a dog.

Catherine and Tom live in Sweden, a country where everyone is expected to lead an orderly life according to rules laid down by the government, which also provides a high level of care for its people. This level of care costs money.

People in Sweden pay taxes on everything, so aren't surprised to find that owning a dog means more taxes. Some people are paying as much as 500 Swedish kronor in taxes a year for the right to keep their dog, which is spent by the government on dog hospitals and sometimes medical treatment for a dog that falls ill. However, most such treatment is expensive, so owners often decide to offer health and even life for their dog.

In Sweden dog owners must pay for any damage their dog does. A Swedish Kennel Club official explains what this means: if your dog runs out on the road and gets hit by a passing car, you, as the owner, have to pay for any damage done to the car, even if your dog has been killed in the accident.

Q: How old is Catherine?
A: 54

Q: where does she live?
A:

Model answer: Stockholm
Turker answers: Sweden, Sweden, in Sweden, Sweden

Radford, Alec, et al. "Language models are unsupervised multitask learners." *OpenAI Blog* 1.8 (2019): 9.

ALBERT: A Lite BERT for Self-supervised Learning of Language Representations

Boulder

YouTube Video

$V \rightarrow$ vocabulary size ($\approx$ 30,000)
$E \rightarrow$ vocabulary embedding size
$L \rightarrow$ number of encoder layers
$H \rightarrow$ hidden size
$4H \rightarrow$ feedforward/filter size
$H/64 \rightarrow$ number of attention heads

Factorized embedding parameterization

Reduce the embedding parameters from $\mathcal{O}(V \times H)$ to $\mathcal{O}(V \times E + E \times H)$

This parameter reduction is significant when $H \gg E$.

Cross-layer parameter sharing

Inter-sentence coherence loss

In addition to the masked language modeling (MLM) loss, BERT uses an additional loss called next-sentence prediction (NSP).

Positive examples:
consecutive segments from the corpus

Negative examples:
pairing segments from different documents

sentence-order prediction (SOP) loss

Positive examples:
consecutive segments from the corpus

Negative examples:
swapped consecutive segments

Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	0.3x

The effect of vocabulary embedding size on the performance of ALBERT-base.

Model	E	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base not-shared	64	87M	89.9/82.9	80.1/77.8	82.9	91.5	66.7	81.3
	128	89M	89.9/82.8	80.3/77.3	83.7	91.5	67.9	81.7
	256	93M	90.2/83.2	80.3/77.4	84.1	91.9	67.3	81.8
	768	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base all-shared	64	10M	88.7/81.4	77.5/74.8	80.8	89.4	63.5	79.0
	128	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1
	256	16M	88.8/81.5	79.1/76.3	81.5	90.3	63.4	79.6
	768	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8

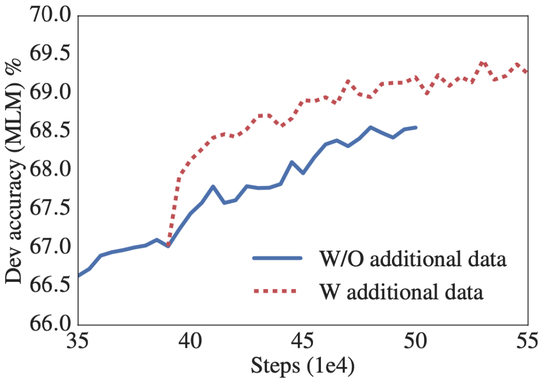
The effect of cross-layer parameter-sharing strategies, ALBERT-base configuration.

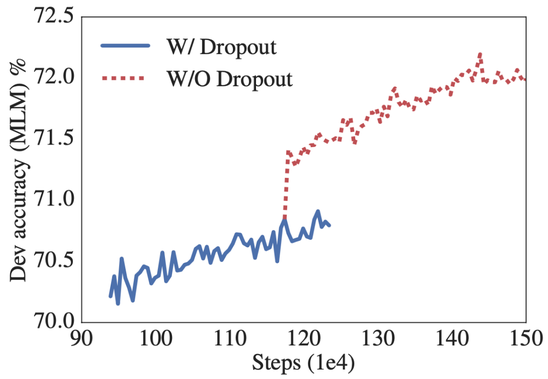
Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base E=768	all-shared	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8
	shared-attention	83M	89.9/82.7	80.0/77.2	84.0	91.4	67.7	81.6
	shared-FFN	57M	89.2/82.1	78.2/75.4	81.5	90.8	62.6	79.5
	not-shared	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base E=128	all-shared	12M	89.3/82.3	80.0/77.1	82.0	90.3	64.0	80.1
	shared-attention	64M	89.9/82.8	80.7/77.9	83.4	91.9	67.6	81.7
	shared-FFN	38M	88.9/81.6	78.6/75.6	82.3	91.7	64.4	80.2
	not-shared	89M	89.9/82.8	80.3/77.3	83.2	91.5	67.9	81.6

NSP vs. SOP

	Intrinsic Tasks		
SP tasks	MLM	NSP	SOP
None	54.9	52.4	53.3
NSP	54.5	90.5	52.0
SOP	54.0	78.9	86.5

Downstream Tasks					
SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
88.6/81.5	78.1/75.3	81.5	89.9	61.7	79.0
88.4/81.5	77.2/74.6	81.6	**91.1**	62.3	79.2
89.3/82.3	**80.0/77.1**	**82.0**	90.3	**64.0**	**80.1**





Lan, Zhenzhong, et al. "Albert: A lite bert for self-supervised learning of language representations." *arXiv preprint arXiv:1909.11942* (2019).

RoBERTa: A Robustly Optimized BERT Pretraining Approach

Boulder

Robustly **o**ptimized **BERT** **a**pproach
replication study of BERT pre-training
BERT
$x_1, x_2, \ldots, x_N \rightarrow$ segment 1 (sequence of tokens)
$y_1, y_2, \ldots, y_M \rightarrow$ segment 2 (sequence of tokens)
$[CLS], x_1, x_2, \ldots, x_N, [SEP], y_1, y_2, \ldots, y_M, [EOS] \rightarrow$ single input sequence
$M + N < T \rightarrow$ maximum sequence length during training
$L \rightarrow$ number of layers
$A \rightarrow$ number of self-attention heads
$H \rightarrow$ hidden dimension
– masked language modeling (MLM)
15% of input tokens are selected for possible replacement. Of the selected tokens, 80% are replaced with [MASK], 10% are left unchanged, and 10% are replaced by a randomly selected vocabulary token.
– next sentence prediction (NSP)
positive examples (consecutive sentences from the text corpus)
negative examples (pairing segments from different documents)
The NSP objective was designed to improve performance on downstream tasks, such as Natural Language Inference
$T = 512 \rightarrow$ maximum length
$S = 1,000,000 \rightarrow$ updates
$B = 256 \rightarrow$ minibatch
16GB of uncompressed text (BOOKCORPUS + English WIKIPEDIA)
RoBERTa's modifications are simple, they include: (1) training the model longer, with bigger batches, over more data; (2) removing the next sentence prediction objective; (3) training on longer sequences; (4) dynamically changing the masking pattern applied to the training data; and (5) collecting a large new dataset (CC-NEWS) of comparable size to other privately used datasets, to better control for training set size effects.

160GB of uncompressed text:
– BOOKCORPUS + English WIKIPEDIA (16GB)
– CC-NEWS (76GB)
– OPENWEBTEXT (38GB)
– STORIES (31GB)

masking once during data preprocessing ←

Masking	SQuAD 2.0	MNLI-m	SST-2
reference	76.3	84.3	92.8
Our reimplementation:			
static	78.3	84.3	92.5
dynamic	78.7	84.0	92.9

Model	SQuAD 1.1/2.0	MNLI-m	SST-2	RACE
Our reimplementation (with NSP loss):				
SEGMENT-PAIR	90.4/78.7	84.0	92.9	64.2
SENTENCE-PAIR	88.7/76.2	82.9	92.1	63.0
Our reimplementation (without NSP loss):				
FULL-SENTENCES	90.4/79.1	84.7	92.5	64.8
DOC-SENTENCES	90.6/79.7	84.7	92.7	65.6
$\text{BERT}_{\text{BASE}}$	88.5/76.3	84.3	92.8	64.3
$\text{XLNet}_{\text{BASE}}$ (K = 7)	–/81.3	85.8	92.7	66.1
$\text{XLNet}_{\text{BASE}}$ (K = 6)	–/81.0	85.6	93.4	66.7

Model	SQuAD 1.1 EM	SQuAD 1.1 F1	SQuAD 2.0 EM	SQuAD 2.0 F1
Single models on dev, w/o data augmentation				
$\text{BERT}_{\text{LARGE}}$	84.1	90.9	79.0	81.8
$\text{XLNet}_{\text{LARGE}}$	**89.0**	94.5	86.1	88.8
RoBERTa	88.9	**94.6**	**86.5**	**89.4**
Single models on test (as of July 25, 2019)				
$\text{XLNet}_{\text{LARGE}}$			86.3†	89.1†
RoBERTa			86.8	89.8
XLNet + SG-Net Verifier			**87.0**†	**89.9**†

batch size ←

bsz	steps	lr	ppl	MNLI-m	SST-2
256	1M	1e-4	3.99	84.7	92.7
2K	125K	7e-4	**3.68**	**85.2**	**92.9**
8K	31K	1e-3	3.77	84.6	92.8

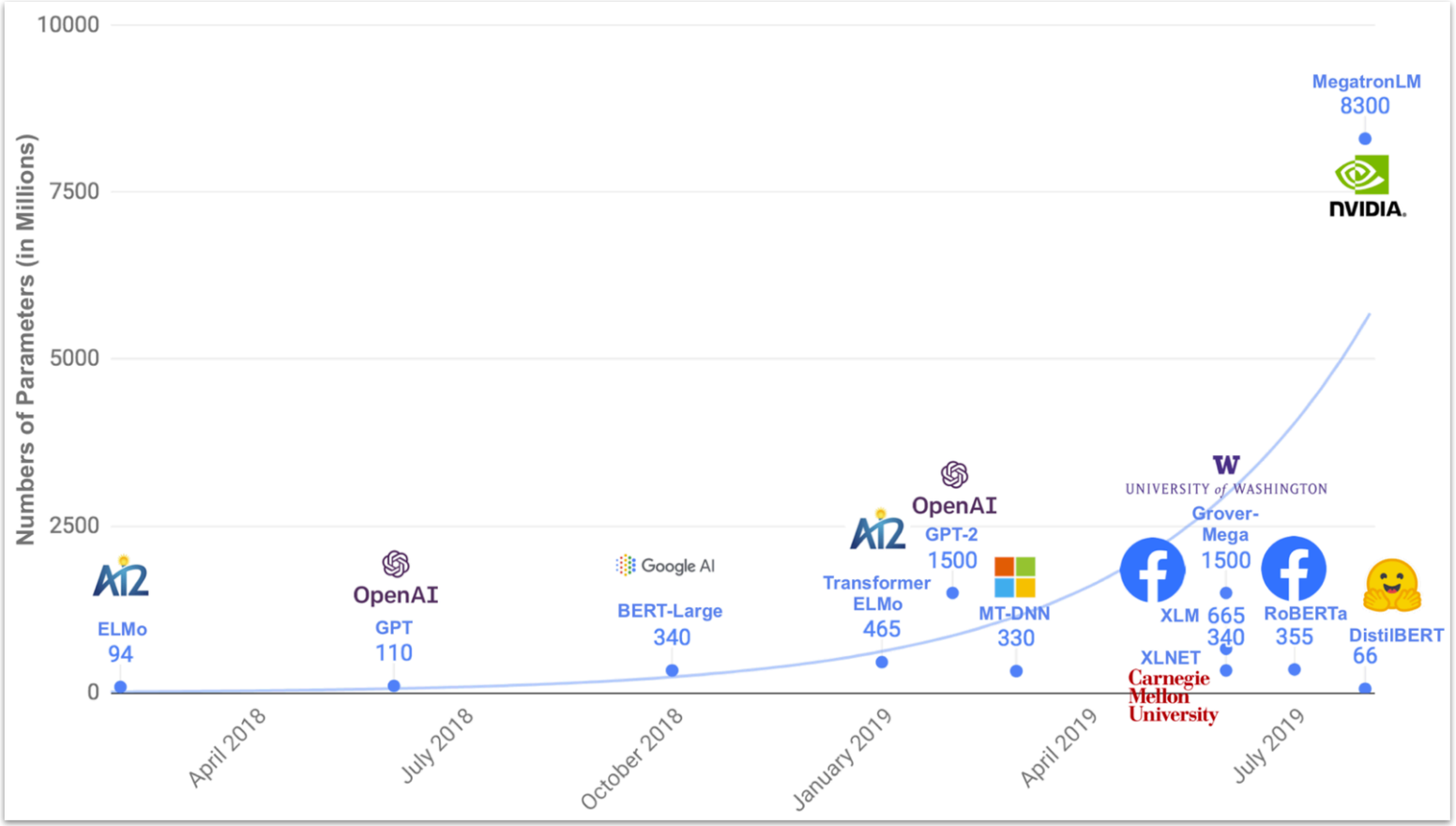
Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
$\text{XLNet}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

Results on the RACE test set.

Model	Accuracy	Middle	High
Single models on test (as of July 25, 2019)			
$\text{BERT}_{\text{LARGE}}$	72.0	76.6	70.1
$\text{XLNet}_{\text{LARGE}}$	81.7	85.4	80.2
RoBERTa	**83.2**	**86.5**	**81.3**

Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).

DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter



DistilBERT retains 97% of BERT performance.

Model	Score	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Knowledge distillation

compact model → student

larger model → teacher

$L_{ce} = \sum_i t_i \log s_i$ → distillation loss

t_i → probability estimated by a teacher

s_i → probability estimated by a student

$p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$ → soft-temperature

Same temperature T for student and teacher at training

$T = 1$ at inference

Triple Loss

Combining language modeling, distillation and cosine-distance losses!

L_{mlm} → masked language modeling loss

L_{cos} → cosine embedding loss (align the directions of the student and teacher hidden states vectors)

Student Architecture & Initialization

Number of layers is reduced by a factor of 2!

Initialize the student from the teacher by taking one layer out of two!

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv preprint arXiv:1910.01108* (2019).

Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context

YouTube Playlist

$x = (x_1, x_2, \ldots, x_T) \rightarrow$ corpus of tokens

$p(x) = \prod_t p(x_t|x_{<t}) \rightarrow$ Language Modeling

encode the context $x_{<t}$
into a fixed size hidden state

$$\begin{cases} s_\tau = [x_{\tau,1}, x_{\tau,2}, \ldots, x_{\tau,L}] \\ s_{\tau+1} = [x_{\tau+1,1}, x_{\tau+1,2}, \ldots, x_{\tau+1,L}] \end{cases}$$

consecutive segments of length L

$h_\tau^n \in \mathbb{R}^{L \times d}$

n-th layer hidden state sequence produced for the τ-th segment s_τ

$\tilde{h}_{\tau+1}^{n-1} = [\text{SG}(h_\tau^{n-1}), h_{\tau+1}^{n-1}] \in \mathbb{R}^{2L \times d}$

stop gradient

$q_{\tau+1}^n = h_{\tau+1}^{n-1} W_q^T$

$k_{\tau+1}^n = \tilde{h}_{\tau+1}^{n-1} W_k^T$

$v_{\tau+1}^n = \tilde{h}_{\tau+1}^{n-1} W_v^T$

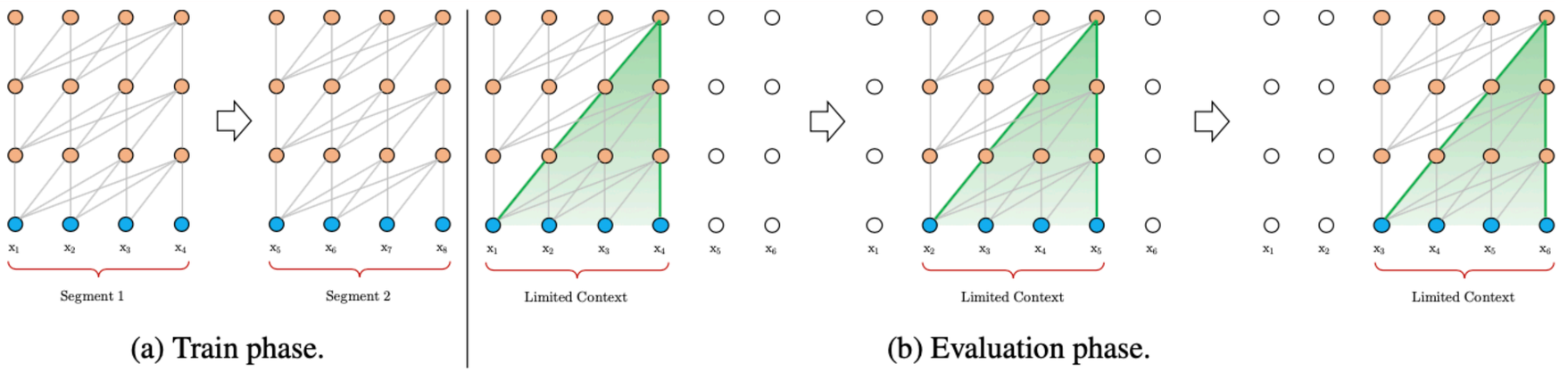
$h_{\tau+1}^n = \text{transformer-layer}(q_{\tau+1}^n, k_{\tau+1}^n, v_{\tau+1}^n)$

$m_\tau^n \in \mathbb{R}^{M \times d}$ (e.g., $m_\tau^n = h_\tau^n$) $\rightarrow$ memory

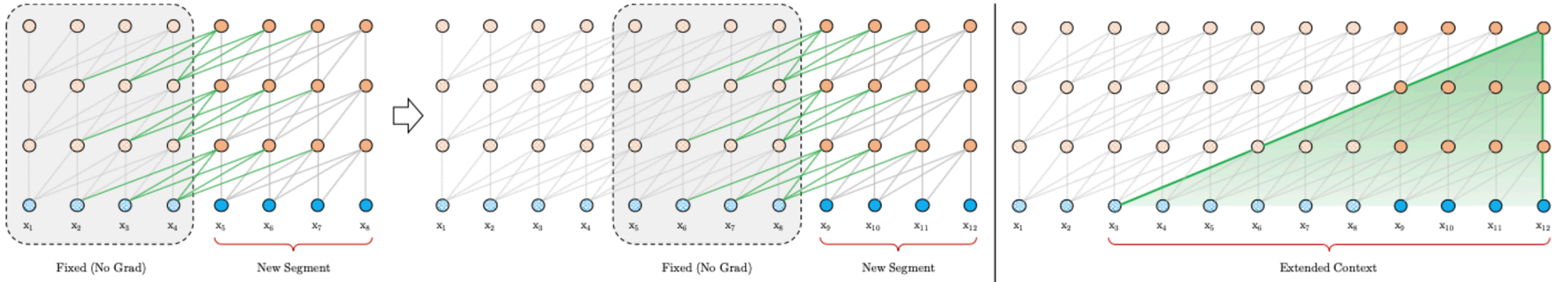
Relative Positional Encodings

$U \in \mathbb{R}^{L_{\max} \times d}$

"absolute" positional encodings



(a) Train phase. (b) Evaluation phase.



(a) Training phase. (b) Evaluation phase.

The model has no information to distinguish the positional difference between $x_{\tau,j}$ and $x_{\tau+1,j}$ for any $j = 1, \ldots, L$

$R \in \mathbb{R}^{L_{\max} \times d}$ i-th row: relative distance of i between two positions

"relative" positional encoding

$$\mathbf{A}_{i,j}^{\text{abs}} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(b)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(c)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(d)}.$$

attention score between query q_i and key vector k_j

$$\mathbf{A}_{i,j}^{\text{rel}} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(b)} + \underbrace{u^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(c)} + \underbrace{v^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(d)}.$$

Dai, Zihang, et al. "Transformer-xl: Attentive language models beyond a fixed-length context." *arXiv preprint arXiv:1901.02860* (2019).

XLNet: Generalized Autoregressive Pretraining for Language Understanding

Boulder

YouTube Video

Pre-training objectives:
- Autoencoding (AE) Language Modeling
- Autoregressive (AR) Language Modeling

$x = (x_1, x_2, \ldots, x_T)$

$$p(x) = \prod_{t=1}^{T} p(x_t|x_{<t}) \rightarrow \text{forward product}$$

$$p(x) = \prod_{t=T}^{1} p(x_t|x_{>t}) \rightarrow \text{backward product} \Big\} \text{AR}$$

[MASK] → AE (pretrain-finetune discrepency)

all possible permutations of the factorization order

$$\max_\theta \log p_\theta(\mathbf{x}) = \sum_{t=1}^{T} \log p_\theta(x_t \mid \mathbf{x}_{<t})$$

$$= \sum_{t=1}^{T} \log \frac{\exp\left(h_\theta(\mathbf{x}_{1:t-1})^\top e(x_t)\right)}{\sum_{x'} \exp\left(h_\theta(\mathbf{x}_{1:t-1})^\top e(x')\right)}$$

$x \rightarrow \hat{x} \rightarrow \overline{x}$ masked tokens

corrupted ([MASK] 15%)

$$\max_\theta \log p_\theta(\bar{\mathbf{x}} \mid \hat{\mathbf{x}}) \approx \sum_{t=1}^{T} m_t \log p_\theta(x_t \mid \hat{\mathbf{x}})$$

$m_t = 1$ indicates x_t is masked.

$$= \sum_{t=1}^{T} m_t \log \frac{\exp\left(H_\theta(\hat{\mathbf{x}})_t^\top e(x_t)\right)}{\sum_{x'} \exp\left(H_\theta(\hat{\mathbf{x}})_t^\top e(x')\right)}$$

$\mathcal{Z}_T \rightarrow$ set of all possible permutations of $\{1, 2, \ldots, T\}$

$z_t \rightarrow t$-th element of a permutation $z \in \mathcal{Z}_T$

$z_{<t} \rightarrow$ the first $t-1$ elements of a permutation $z \in \mathcal{Z}_T$

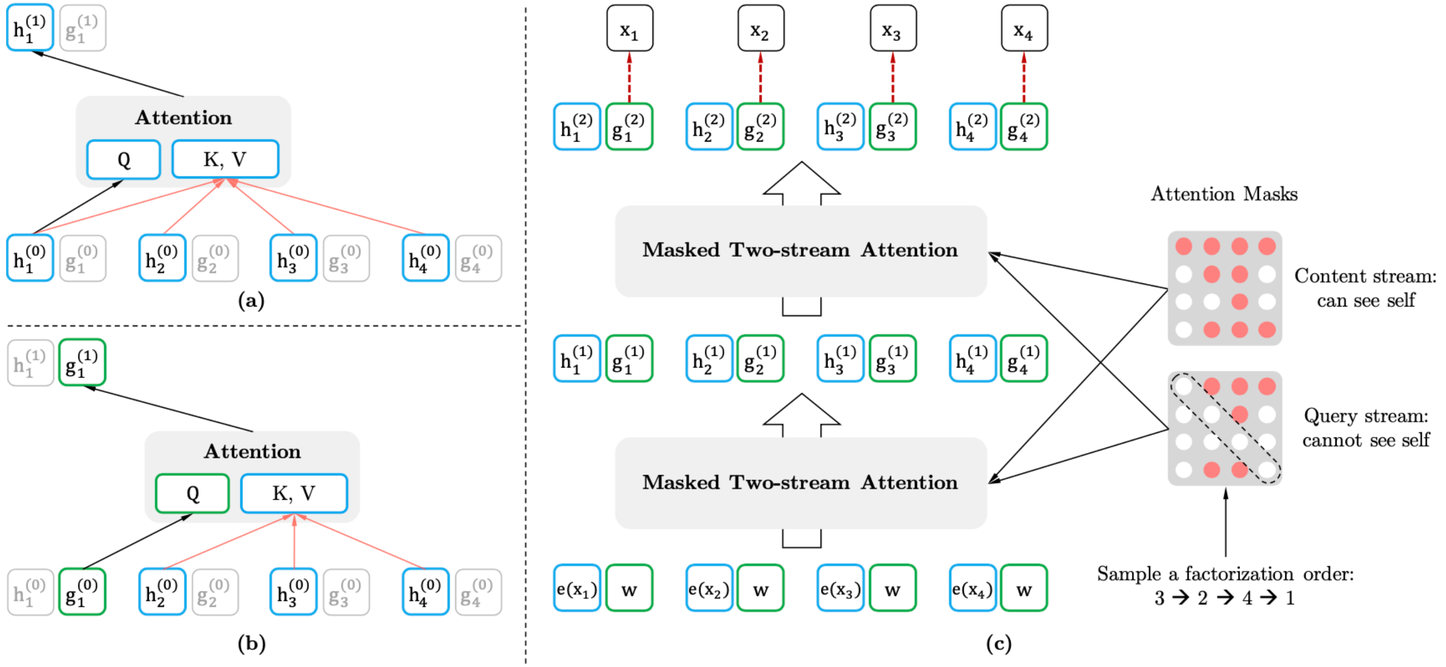
$$\mathcal{J}_{\text{BERT}} = \log p(\text{New} \mid \text{is a city}) + \log p(\text{York} \mid \text{is a city}),$$

$$\mathcal{J}_{\text{XLNet}} = \log p(\text{New} \mid \text{is a city}) + \log p(\text{York} \mid \text{New}, \text{is a city}).$$

Partial Prediction

$$\max_\theta \quad \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{t=1}^{T} \log p_\theta(x_{z_t} \mid \mathbf{x}_{\mathbf{z}_{<t}})\right] \qquad \max_\theta \quad \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\log p_\theta(\mathbf{x}_{\mathbf{z}_{>c}} \mid \mathbf{x}_{\mathbf{z}_{\le c}})\right] = \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{t=c+1}^{|\mathbf{z}|} \log p_\theta(x_{z_t} \mid \mathbf{x}_{\mathbf{z}_{<t}})\right]$$

Make the objective function depend on the position (i.e., the value of z_t) it will predict.



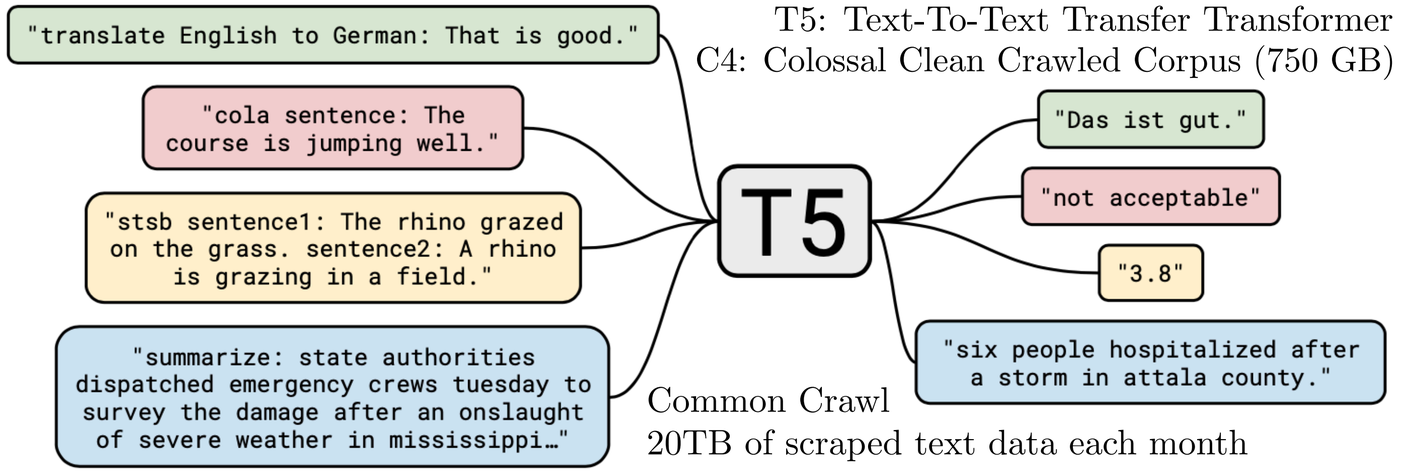
$$g_{z_t}^{(m)} \leftarrow \text{Attention}(\text{Q} = g_{z_t}^{(m-1)}, \text{KV} = \mathbf{h}_{\mathbf{z}_{<t}}^{(m-1)}; \theta), \quad \text{(query stream: use } z_t \text{ but cannot see } x_{z_t})$$

$$h_{z_t}^{(m)} \leftarrow \text{Attention}(\text{Q} = h_{z_t}^{(m-1)}, \text{KV} = \mathbf{h}_{\mathbf{z}_{\le t}}^{(m-1)}; \theta), \quad \text{(content stream: use both } z_t \text{ and } x_{z_t}).$$

Yang, Zhilin, et al. "Xlnet: Generalized autoregressive pretraining for language understanding." *Advances in neural information processing systems*. 2019.

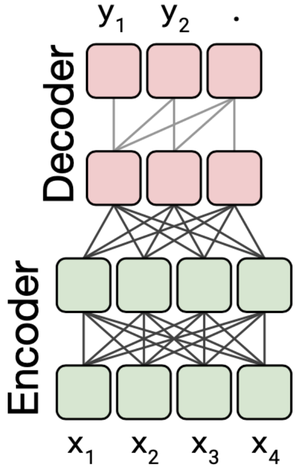
Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

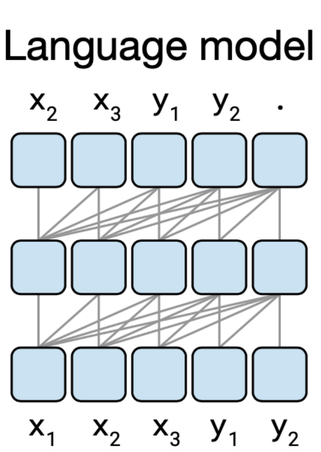
Boulder

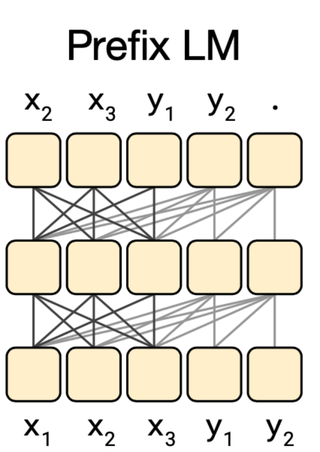


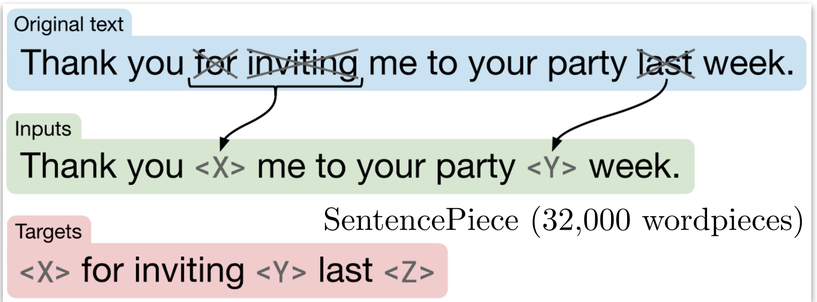
T5: Text-To-Text Transfer Transformer
C4: Colossal Clean Crawled Corpus (750 GB)

Common Crawl
20TB of scraped text data each month

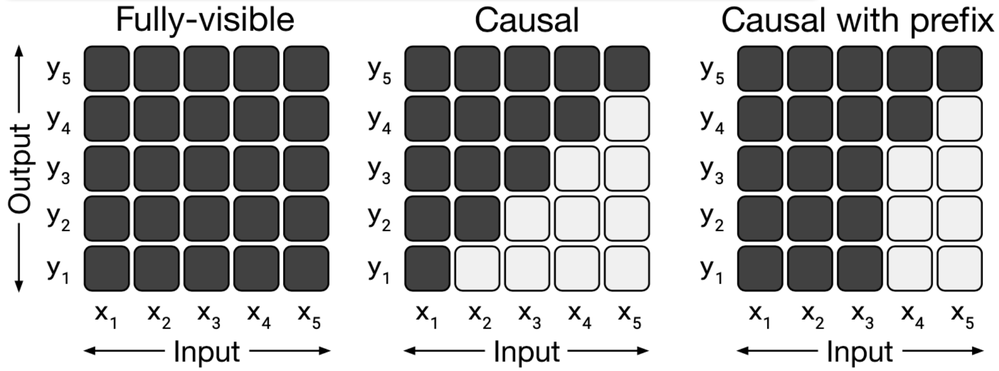








Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you <M> <M> me to your party apple week .	*(original text)*
Deshuffling	party me for your to . last fun you inviting week Thank	*(original text)*
MASS-style Song 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>



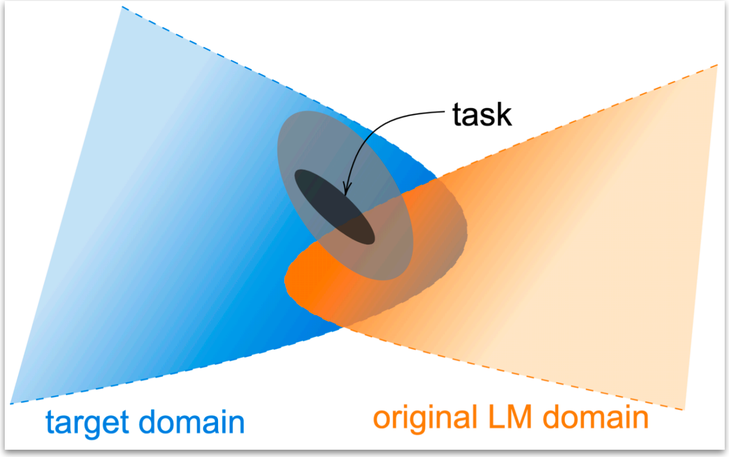
$1/\sqrt{\max(n, 10^4)}$ → "inverse square root" learning rate schedule

AdaFactor Optimiser

Data set	Size	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ C4	745GB	83.28	**19.24**	80.88	71.36	**26.98**	**39.82**	**27.65**
C4, unfiltered	6.1TB	81.46	19.14	78.78	68.04	26.55	39.34	27.21
RealNews-like	35GB	**83.83**	**19.23**	80.39	72.38	**26.75**	**39.90**	**27.48**
WebText-like	17GB	**84.03**	**19.31**	**81.42**	71.40	**26.80**	**39.74**	**27.59**
Wikipedia	16GB	81.85	**19.31**	81.29	68.01	**26.94**	39.69	**27.67**
Wikipedia + TBC	20GB	83.65	**19.28**	**82.08**	**73.24**	**26.77**	39.63	**27.57**

Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *arXiv preprint arXiv:1910.10683* (2019).

Don't Stop Pretraining: Adapt Language Models to Domains and Tasks



A spectra of domains defined around tasks at various levels of granularity (e.g., Amazon reviews for a specific product, all Amazon reviews, all reviews on the web, the web).

List of the domain-specific unlabeled datasets.

Domain	Pretraining Corpus	# Tokens	Size	$\mathcal{L}_{\text{ROB.}}$	$\mathcal{L}_{\text{DAPT}}$
BIOMED	2.68M full-text papers from S2ORC (Lo et al., 2020)	7.55B	47GB	1.32	0.99
CS	2.22M full-text papers from S2ORC (Lo et al., 2020)	8.10B	48GB	1.63	1.34
NEWS	11.90M articles from REALNEWS (Zellers et al., 2019)	6.66B	39GB	1.08	1.16
REVIEWS	24.75M AMAZON reviews (He and McAuley, 2016)	2.11B	11GB	2.10	1.93
ROBERTA (baseline)	see Appendix §A.1	N/A	160GB	‡1.19	-

Domain-Adaptive Pretraining (DAPT)

Continue pretraining RoBERTa on a large corpus of unlabeled domain-specific text.

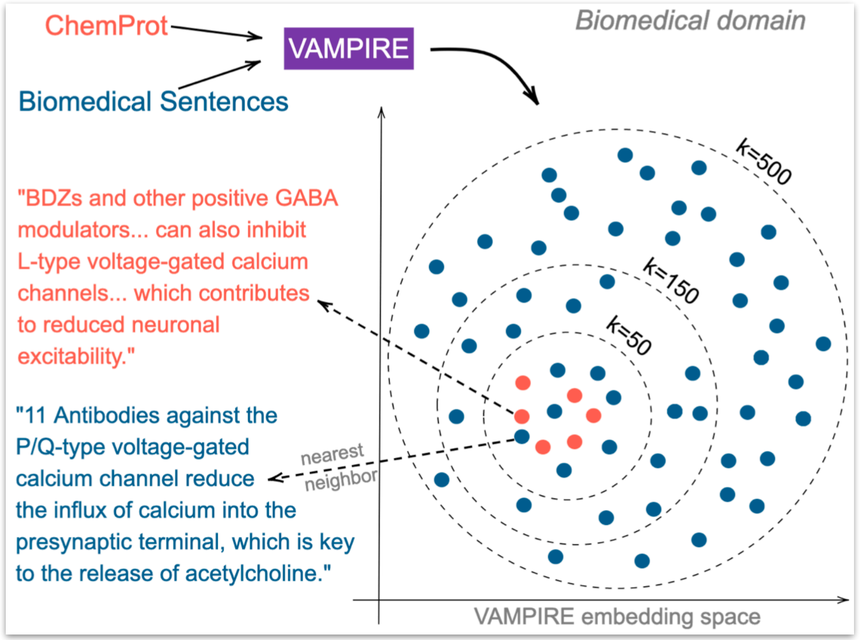
Task-Adaptive Pretraining (TAPT)

Pretraining on the unlabeled training set for a given task.

Domain	Task	RoBERTa	Additional Pretraining Phases		
			DAPT	TAPT	DAPT + TAPT
BIOMED	CHEMPROT	$81.9_{1.0}$	$84.2_{0.2}$	$82.6_{0.4}$	$\mathbf{84.4}_{0.4}$
	†RCT	$87.2_{0.1}$	$87.6_{0.1}$	$87.7_{0.1}$	$\mathbf{87.8}_{0.1}$
CS	ACL-ARC	$63.0_{5.8}$	$75.4_{2.5}$	$67.4_{1.8}$	$\mathbf{75.6}_{3.8}$
	SCIERC	$77.3_{1.9}$	$80.8_{1.5}$	$79.3_{1.5}$	$\mathbf{81.3}_{1.8}$
NEWS	HYPERPARTISAN	$86.6_{0.9}$	$88.2_{5.9}$	$\mathbf{90.4}_{5.2}$	$90.0_{6.6}$
	†AGNEWS	$93.9_{0.2}$	$93.9_{0.2}$	$94.5_{0.1}$	$\mathbf{94.6}_{0.1}$
REVIEWS	†HELPFULNESS	$65.1_{3.4}$	$66.5_{1.4}$	$68.5_{1.9}$	$\mathbf{68.7}_{1.8}$
	†IMDB	$95.0_{0.2}$	$95.4_{0.1}$	$95.5_{0.1}$	$\mathbf{95.6}_{0.1}$

eight classification tasks

Automated Data Selection for TAPT



Gururangan, Suchin, et al. "Don't stop pretraining: adapt language models to domains and tasks." *arXiv preprint arXiv:2004.10964* (2020).

Cross-lingual Language Model Pretraining

YouTube Video

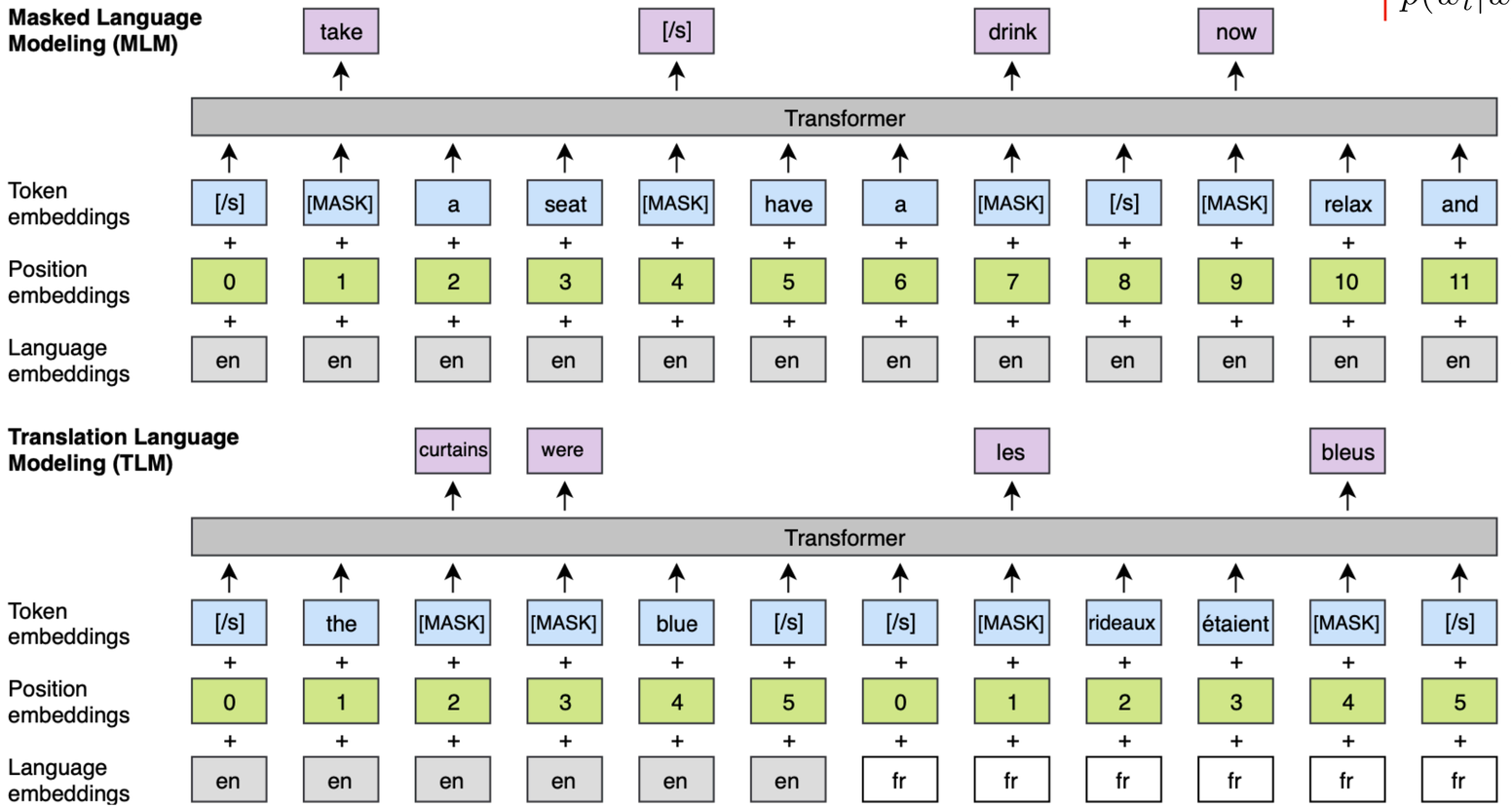
– multiple languages
– cross-lingual pretraining
cross-lingual language models (XLMs)
– monolingual data (unsupervised)
– parallel data (supervised)
$\{C_i\}_{i=1}^N \rightarrow N$ monolingual corpora
$n_i \rightarrow$ number of sentences in C_i

Shared sub-word vocabulary
Shared vocabulary created using Byte Pair Encoding (BPE)
Learn the BPE splits on the concatenation of sentences sampled randomly from the monolingual corpora

Sentences are sampled according to a multinomial distribution with probabilities $\{q_i\}_{i=1}^N$

$$q_i = \frac{p_i^\alpha}{\sum_{j=1}^N p_j^\alpha} \quad \text{with} \quad p_i = \frac{n_i}{\sum_{k=1}^N n_k}, \alpha = 0.5$$

Alleviates the bias towards high-resource languages!
Causal Language Modeling (CLM)
$p(w_t|w_1, \ldots, w_{t-1}; \theta) \rightarrow$ a Transformer language model



Results on cross-lingual classification accuracy. Test accuracy on the 15 XNLI languages.

	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Δ
Machine translation baselines (TRANSLATE-TRAIN)																
Devlin et al. (2018)	81.9	-	77.8	75.9	-	-	-	-	70.7	-	-	76.6	-	-	61.6	-
XLM (MLM+TLM)	85.0	80.2	80.8	80.3	78.1	79.3	78.1	74.7	76.5	76.6	75.5	78.6	72.3	70.9	63.2	76.7
Machine translation baselines (TRANSLATE-TEST)																
Devlin et al. (2018)	81.4	-	74.9	74.4	-	-	-	-	70.4	-	-	70.1	-	-	62.1	-
XLM (MLM+TLM)	85.0	79.0	79.5	78.1	77.8	77.6	75.5	73.7	73.7	70.8	70.4	73.6	69.0	64.7	65.1	74.2
Evaluation of cross-lingual sentence encoders																
Conneau et al. (2018b)	73.7	67.7	68.7	67.7	68.9	67.9	65.4	64.2	64.8	66.4	64.1	65.8	64.1	55.7	58.4	65.6
Devlin et al. (2018)	81.4	-	74.3	70.5	-	-	-	-	62.1	-	-	63.8	-	-	58.3	-
Artetxe and Schwenk (2018)	73.9	71.9	72.9	72.6	73.1	74.2	71.5	69.7	71.4	72.0	69.2	71.4	65.5	62.2	61.0	70.2
XLM (MLM)	83.2	76.5	76.3	74.2	73.1	74.0	73.1	67.8	68.5	71.2	69.2	71.9	65.7	64.6	63.4	71.5
XLM (MLM+TLM)	**85.0**	**78.7**	**78.9**	**77.8**	**76.6**	**77.4**	**75.3**	**72.5**	**73.1**	**76.1**	**73.2**	**76.5**	**69.6**	**68.4**	**67.3**	**75.1**

Results on unsupervised MT. BLEU

		en-fr	fr-en	en-de	de-en	en-ro	ro-en
Previous state-of-the-art - Lample et al. (2018b)							
NMT		25.1	24.2	17.2	21.0	21.2	19.4
PBSMT		28.1	27.2	17.8	22.7	21.3	23.0
PBSMT + NMT		27.6	27.7	20.2	25.2	25.1	23.9
Our results for different encoder and decoder initializations							
EMB	EMB	29.4	29.4	21.3	27.3	27.5	26.6
-	-	13.0	15.8	6.7	15.3	18.9	18.3
-	CLM	25.3	26.4	19.2	26.0	25.7	24.6
-	MLM	29.2	29.1	21.6	28.6	28.2	27.3
CLM	-	28.7	28.2	24.4	30.3	29.2	28.0
CLM	CLM	30.4	30.0	22.7	30.5	29.0	27.8
CLM	MLM	32.3	31.6	24.3	32.5	31.6	29.8
MLM	-	31.6	32.1	**27.0**	33.2	31.8	30.5
MLM	CLM	**33.4**	32.3	24.9	32.9	31.7	30.4
MLM	MLM	**33.4**	**33.3**	26.4	**34.3**	**33.3**	**31.8**

Results on supervised MT. BLEU scores

Pretraining	-	CLM	MLM
Sennrich et al. (2016)	33.9	-	-
ro → en	28.4	31.5	35.3
ro ↔ en	28.5	31.5	35.6
ro ↔ en + BT	34.4	37.0	**38.5**

Low-resource language model

Training languages	Nepali perplexity
Nepali	157.2
Nepali + English	140.1
Nepali + Hindi	115.6
Nepali + English + Hindi	**109.3**

Lample, Guillaume, and Alexis Conneau. "Cross-lingual language model pretraining." *arXiv preprint arXiv:1901.07291* (2019).

Unsupervised Cross-lingual Representation Learning at Scale

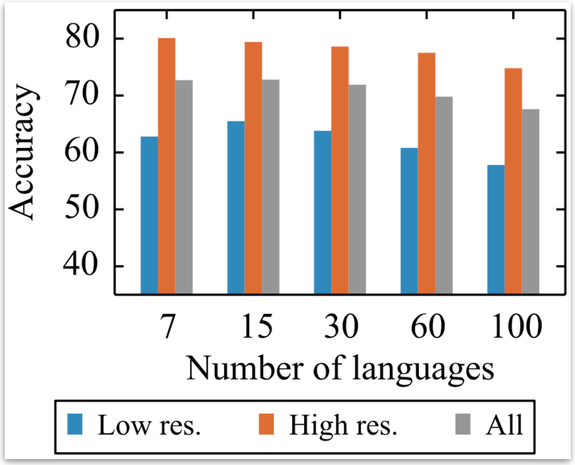
Boulder

XLM-RoBERTa (XLM-R)

XLM-R is a transformer-based multilingual masked language model pre-trained on text in 100 languages, which obtains state-of-the-art performance on cross-lingual classification, sequence labeling and question answering

Curse of Multilinguality

For a fixed model capacity, more languages leads to better cross-lingual performance on low-resource languages up until a point, after which the overall performance on monolingual and cross-lingual benchmarks degrades.



Positive Transfer v.s.
Capacity Dilution Tradeoff

The transfer-interference trade-off: Low-resource languages benefit from scaling to more languages, until dilution (interference) kicks in and degrades overall performance.

Masked Language Models → Same as Lample and Conneau (2019)

Subword tokenization directly on raw text data using Sentence Piece with a unigram language model.

$\alpha = 0.3$

Unlike Lample and Conneau (2019), do not use language embeddings, which allows the model to better deal with code-switching.

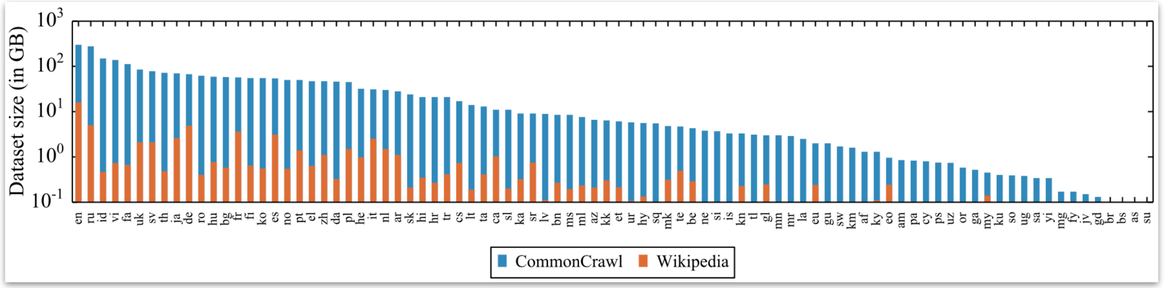
Use a large vocabulary size of 250K.

Scaling to a hundred languages

Scaling the Amount of Training Data

Wikipedia → limited scale!

More than two terabytes of filtered CommonCrawl data!



Five-hundred 32GB Nvidia V100 GPUs!

Model	D	#M	#lg	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Avg
Fine-tune multilingual model on English training set (Cross-lingual Transfer)																			
Lample and Conneau (2019)	Wiki+MT	N	15	85.0	78.7	78.9	77.8	76.6	77.4	75.3	72.5	73.1	76.1	73.2	76.5	69.6	68.4	67.3	75.1
Huang et al. (2019)	Wiki+MT	N	15	85.1	79.0	79.4	77.8	77.2	77.2	76.3	72.8	73.5	76.4	73.6	76.2	69.4	69.7	66.7	75.4
Devlin et al. (2018)	Wiki	N	102	82.1	73.8	74.3	71.1	66.4	68.9	69.0	61.6	64.9	69.5	55.8	69.3	60.0	50.4	58.0	66.3
Lample and Conneau (2019)	Wiki	N	100	83.7	76.2	76.6	73.7	72.4	73.0	72.1	68.1	68.4	72.0	68.2	71.5	64.5	58.0	62.4	71.3
Lample and Conneau (2019)	Wiki	1	100	83.2	76.7	77.7	74.0	72.7	74.1	72.7	68.7	68.6	72.9	68.9	72.5	65.6	58.2	62.4	70.7
XLM-R$_{\text{Base}}$	CC	1	100	85.8	79.7	80.7	78.7	77.5	79.6	78.1	74.2	73.8	76.5	74.6	76.7	72.4	66.5	68.3	76.2
XLM-R	CC	1	100	**89.1**	**84.1**	**85.1**	**83.9**	**82.9**	**84.0**	**81.2**	**79.6**	**79.8**	**80.8**	**78.1**	**80.2**	**76.9**	**73.9**	**73.8**	**80.9**
Translate everything to English and use English-only model (TRANSLATE-TEST)																			
BERT-en	Wiki	1	1	88.8	81.4	82.3	80.1	80.3	80.9	76.2	76.0	75.4	72.0	71.9	75.6	70.0	65.8	65.8	76.2
RoBERTa	Wiki+CC	1	1	**91.3**	82.9	84.3	81.2	81.7	83.1	78.3	76.8	76.6	74.2	74.1	77.5	70.9	66.7	66.8	77.8
Fine-tune multilingual model on each training set (TRANSLATE-TRAIN)																			
Lample and Conneau (2019)	Wiki	N	100	82.9	77.6	77.9	77.9	77.1	75.7	75.5	72.6	71.2	75.8	73.1	76.2	70.4	66.5	62.4	74.2
Fine-tune multilingual model on all training sets (TRANSLATE-TRAIN-ALL)																			
Lample and Conneau (2019)†	Wiki+MT	1	15	85.0	80.8	81.3	80.3	79.1	80.9	78.3	75.6	77.6	78.5	76.0	79.5	72.9	72.8	68.5	77.8
Huang et al. (2019)	Wiki+MT	1	15	85.6	81.1	82.3	80.9	79.5	81.4	79.7	76.8	78.2	77.9	77.1	80.5	73.4	73.8	69.6	78.5
Lample and Conneau (2019)	Wiki	1	100	84.5	80.1	81.3	79.3	78.6	79.4	77.5	75.2	75.6	78.3	75.7	78.3	72.1	69.2	67.7	76.9
XLM-R$_{\text{Base}}$	CC	1	100	85.4	81.4	82.2	80.3	80.4	81.3	79.7	78.6	77.3	79.7	77.9	80.2	76.1	73.1	73.0	79.1
XLM-R	CC	1	100	**89.1**	**85.1**	**86.6**	**85.7**	**85.3**	**85.9**	**83.5**	**83.2**	**83.1**	**83.7**	**81.5**	**83.7**	**81.6**	**78.0**	**78.1**	**83.6**

Table 1: **Results on cross-lingual classification.** We report the accuracy on each of the 15 XNLI languages and the average accuracy. We specify the dataset D used for pretraining, the number of models #M the approach requires and the number of languages #lg the model handles. Our *XLM-R* results are averaged over five different seeds. We show that using the translate-train-all approach which leverages training sets from multiple languages, *XLM-R* obtains a new state of the art on XNLI of 83.6% average accuracy. Results with † are from Huang et al. (2019).

Conneau, Alexis, et al. "Unsupervised cross-lingual representation learning at scale." *arXiv preprint arXiv:1911.02116* (2019).

SpanBERT: Improving Pre-training by Representing and Predicting Spans

Extractive Question Answering

"Which NFL team won Super Bowl 50?"

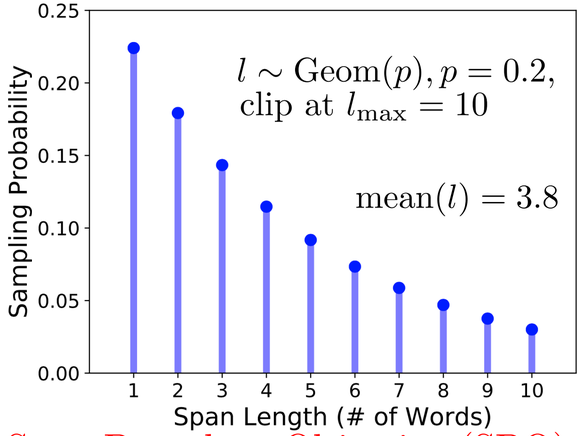
determining that "Denver Broncos" is a type of "NFL team" is critical for answering this question

Span Masking

$X = (x_1, \ldots, x_n) \rightarrow$ sequence of tokens

select a subset of tokens $Y \subset X$ by iteratively sampling spans of text until the masking budget (e.g., 15% of X) has been spent

at each iteration sample a span length (number of words) from a geometric distribution (skewed towards shorter spans)



randomly (uniformly) select the starting point for the span to be masked

As in BERT, also mask 15% of the tokens in total: replacing 80% of the masked tokens with [MASK], 10% with random tokens, and 10% with the original tokens. However, perform this replacement at the span level and not for each token individually; that is, all the tokens in a span are replaced with [MASK] or sampled tokens.

Span Boundary Objective (SBO)

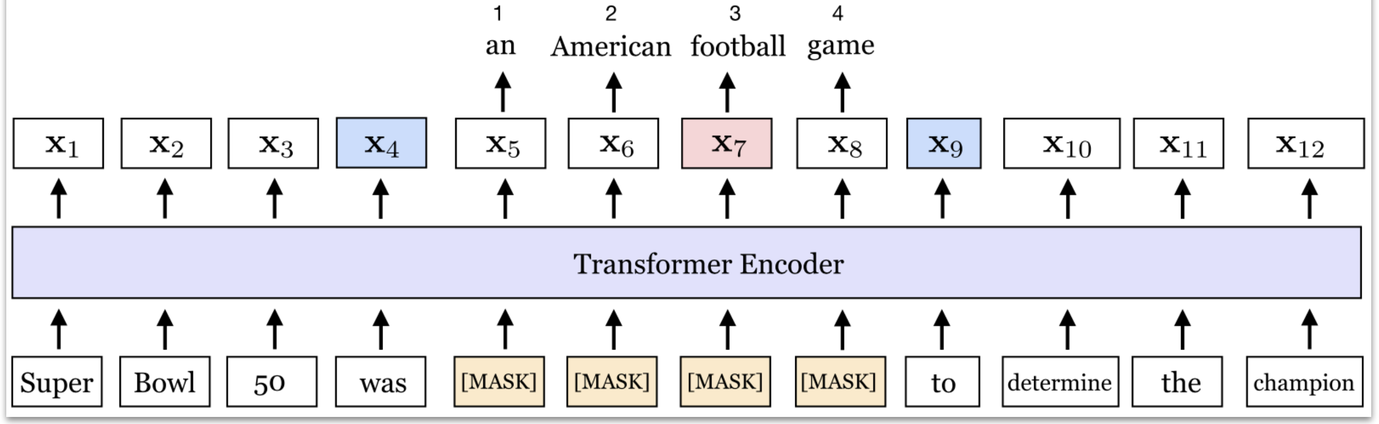
$\mathbf{x}_1, \ldots, \mathbf{x}_n \rightarrow$ output of the transformer encoder for each token in the sequance

$(x_s, \ldots, x_e) \in Y \rightarrow$ masked span of tokens

$(s, e) \rightarrow$ start and end positions

$\mathbf{x}_{s-1}, \mathbf{x}_{e+1} \rightarrow$ output encodings of the external boundary tokens

$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$
$$= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)$$



$\mathbf{p}_{i-s+1} \rightarrow$ position embedding of the target token

$\mathbf{y}_i = f(\mathbf{x}_{s-1}, \mathbf{x}_{e+1}, \mathbf{p}_{i-s+1}) \rightarrow$ representation of token x_i in the span

$$\begin{aligned}\mathbf{h}_0 &= [\mathbf{x}_{s-1}; \mathbf{x}_{e+1}; \mathbf{p}_{i-s+1}] \\ \mathbf{h}_1 &= \text{LayerNorm}\left(\text{GeLU}(\mathbf{W}_1\mathbf{h}_0)\right) \\ \mathbf{y}_i &= \text{LayerNorm}\left(\text{GeLU}(\mathbf{W}_2\mathbf{h}_1)\right)\end{aligned}$$

$\rightarrow$ two layer feed-forward network

Gaussian error Linear Unit (GeLU)

$$\begin{aligned}\mathcal{L}(x_i) &= \mathcal{L}_{\text{MLM}}(x_i) + \mathcal{L}_{\text{SBO}}(x_i) \\ &= -\log P\left(x_i \mid \mathbf{x}_i\right) - \log P\left(x_i \mid \mathbf{y}_i\right)\end{aligned}$$

reuse the input embedding for the target tokens in both MLM and SBO

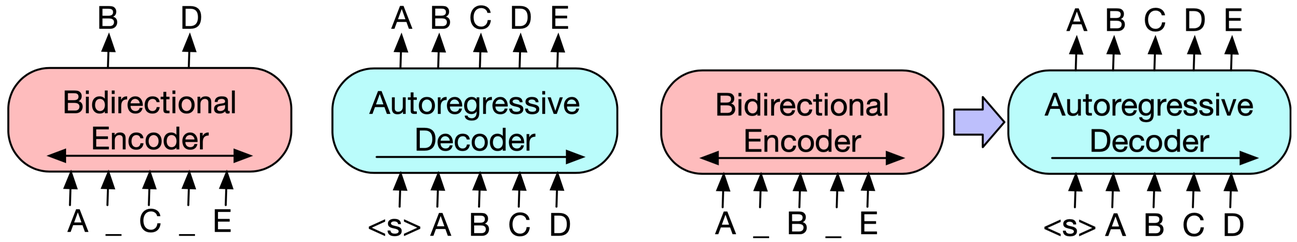
	SQuAD 1.1		SQuAD 2.0	
	EM	F1	EM	F1
Human Perf.	82.3	91.2	86.8	89.4
Google BERT	84.3	91.3	80.0	83.3
Our BERT	86.5	92.6	82.8	85.9
Our BERT-1seq	87.5	93.3	83.8	86.6
SpanBERT	**88.8**	**94.6**	**85.7**	**88.7**

Joshi, Mandar, et al. "Spanbert: Improving pre-training by representing and predicting spans." *Transactions of the Association for Computational Linguistics* 8 (2020): 64-77.

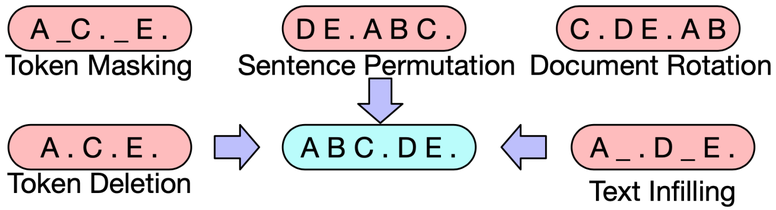
BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

Boulder

BERT GPT BART



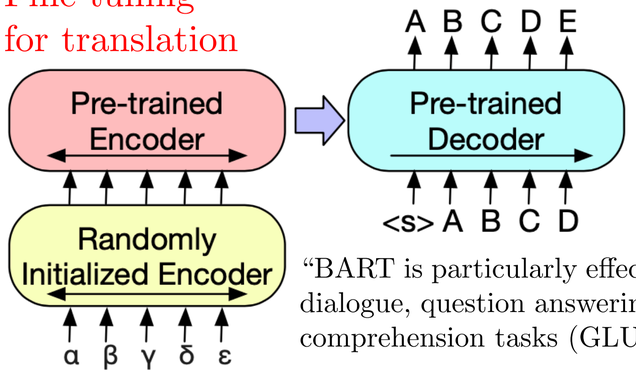
"A key advantage of this setup is the noising flexibility; arbitrary transformations can be applied to the original text, including changing its length."



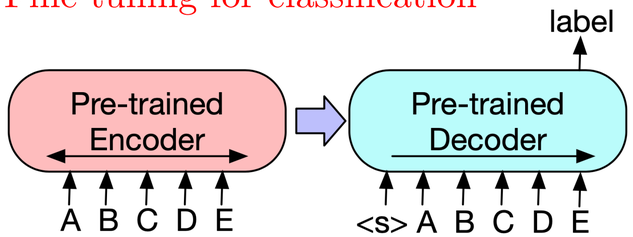
	RO-EN
Baseline	36.80
Fixed BART	36.29
Tuned BART	**37.96**

"Achieving the best performance by both randomly shuffling the order of the original sentences and using a novel in-filling scheme, where arbitrary length spans of text (including zero length) are replaced with a single mask token."

Fine tuning for translation



Fine tuning for classification



"BART is particularly effective when fine tuned for text generation (abstractive dialogue, question answering, and summarization tasks) but also works well for comprehension tasks (GLUE and SQuAD)."

	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 two standard summarization datasets.

	CNN/DailyMail R1	CNN/DailyMail R2	CNN/DailyMail RL	XSum R1	XSum R2	XSum RL
Lead-3	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	28.10	8.02	21.72
UniLM	43.33	20.21	40.51	-	-	-
BERTSUMABS (Liu & Lapata, 2019)	41.72	19.39	38.76	38.76	16.33	31.15
BERTSUMEXTABS (Liu & Lapata, 2019)	42.13	19.60	39.18	38.81	16.50	31.27
BART	**44.16**	**21.28**	**40.90**	**45.14**	**22.27**	**37.25**

Example summaries from the XSum-tuned BART model on WikiNews articles.

Source Document (abbreviated)	BART Summary
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.

abstractive question answering	ELI5 R1	ELI5 R2	ELI5 RL
Best Extractive	23.5	3.1	17.5
Language Model	27.8	4.7	23.1
Seq2Seq	28.3	5.1	22.8
Seq2Seq Multitask	28.9	5.4	23.1
BART	**30.6**	**6.2**	**24.3**

conversational response generation	ConvAI2 Valid F1	ConvAI2 Valid PPL
Seq2Seq + Attention	16.02	35.07
Best System	19.09	17.51
BART	**20.72**	**11.85**

Lewis, Mike, et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." *arXiv preprint arXiv:1910.13461* (2019).

Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

Sentence-pair tasks (e.g., semantic textual similarity)

STS

BERT: sentence 1, sentence 2 $\mapsto$ similarity score

Massive computational overhead!

Finding the most similar pair in a collection of $n = 10,000$ sentences requires about 50 million ($n(n-1)/2 = 49,995,000$) inference computations ($\approx$ 65 hours) with BERT.

BERT: sentence $\mapsto$ vector

average pooling or output of the first token (i.e., the [CLS] token)

The complexity reduces from 65 hours to the computation of 10,000 sentence embeddings ($\approx$ 5 seconds).

This common practice yields rather bad sentence embeddings, often worse than averaging GloVe embeddings.

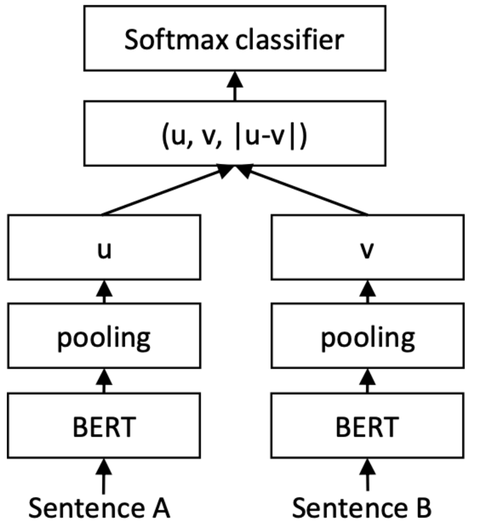
Spearman rank correlation between the cosine similarity of sentence representations and the gold labels for various Textual Similarity (STS) tasks.

Model	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg.
Avg. GloVe embeddings	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
Avg. BERT embeddings	38.78	57.98	57.98	63.15	61.06	46.35	58.40	54.81
BERT CLS-vector	20.16	30.01	20.09	36.88	38.08	16.50	42.63	29.19
InferSent - Glove	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder	64.49	67.80	64.61	76.83	73.18	74.92	**76.69**	71.22
SBERT-NLI-base	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SBERT-NLI-large	72.27	**78.46**	**74.90**	80.99	76.25	**79.23**	73.75	76.55
SRoBERTa-NLI-base	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
SRoBERTa-NLI-large	**74.53**	77.00	73.18	**81.85**	**76.82**	79.10	74.29	**76.68**

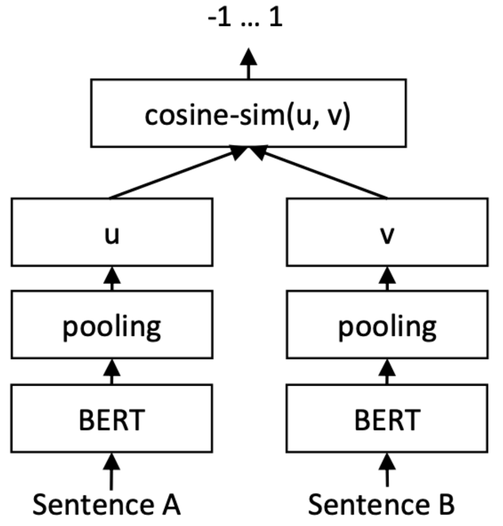
Unsupervised STS

BERT is also not suitable for unsupervised tasks like clustering.

Fine-tuning on SNLI dataset



SBERT architecture at inference



Model	Spearman
Not trained for STS	
Avg. GloVe embeddings	58.02
Avg. BERT embeddings	46.35
InferSent - GloVe	68.03
Universal Sentence Encoder	74.92
SBERT-NLI-base	77.03
SBERT-NLI-large	79.23
Trained on STS benchmark dataset	
BERT-STSb-base	84.30 $\pm$ 0.76
SBERT-STSb-base	84.67 $\pm$ 0.19
SRoBERTa-STSb-base	**84.92** $\pm$ 0.34
BERT-STSb-large	**85.64** $\pm$ 0.81
SBERT-STSb-large	84.45 $\pm$ 0.43
SRoBERTa-STSb-large	85.02 $\pm$ 0.76
Trained on NLI data + STS benchmark data	
BERT-NLI-STSb-base	**88.33** $\pm$ 0.19
SBERT-NLI-STSb-base	85.35 $\pm$ 0.17
SRoBERTa-NLI-STSb-base	84.79 $\pm$ 0.38
BERT-NLI-STSb-large	**88.77** $\pm$ 0.46
SBERT-NLI-STSb-large	86.10 $\pm$ 0.13
SRoBERTa-NLI-STSb-large	86.15 $\pm$ 0.35

Supervised STS

	NLI	STSb
Pooling Strategy		
MEAN	**80.78**	**87.44**
MAX	79.07	69.92
CLS	79.80	86.62
Concatenation		
(u, v)	66.04	-
$(\|u - v\|)$	69.78	-
$(u * v)$	70.54	-
$(\|u - v\|, u * v)$	78.37	-
$(u, v, u * v)$	77.44	-
$(u, v, \|u - v\|)$	**80.78**	-
$(u, v, \|u - v\|, u * v)$	80.44	-

Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." *arXiv preprint arXiv:1908.10084* (2019).

Language Models are Few-Shot Learners

YouTube Video

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.

```
sea otter => loutre de mer        <- example #1
```
↓
gradient update
↓
```
peppermint => menthe poivrée      <- example #2
```
↓
gradient update
↓
...
↓
```
plush giraffe => girafe peluche   <- example #N
```
gradient update
```
cheese =>                         <- prompt
```

In-context Learning

Zero-shot

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

```
Translate English to French:      <- task description
cheese =>                         <- prompt
```

One-shot

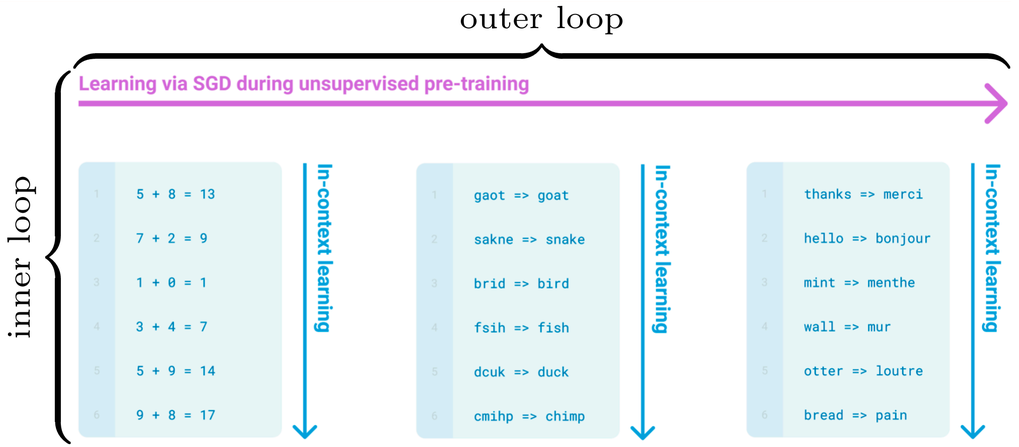
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:      <- task description
sea otter => loutre de mer        <- example
cheese =>                         <- prompt
```

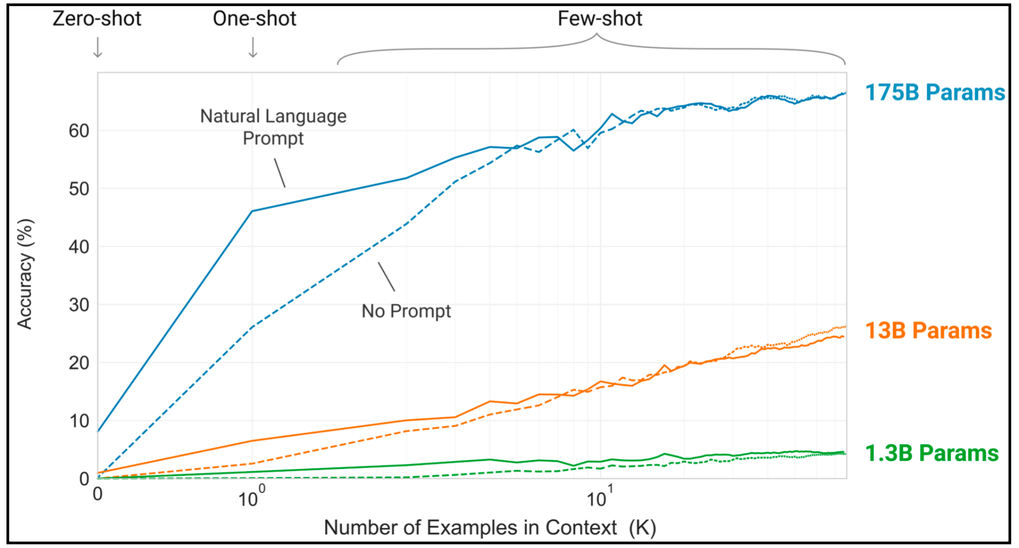
Few-shot

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

```
Translate English to French:      <- task description
sea otter => loutre de mer        <- examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>                         <- prompt
```



Larger models make increasingly efficient use of in-context information

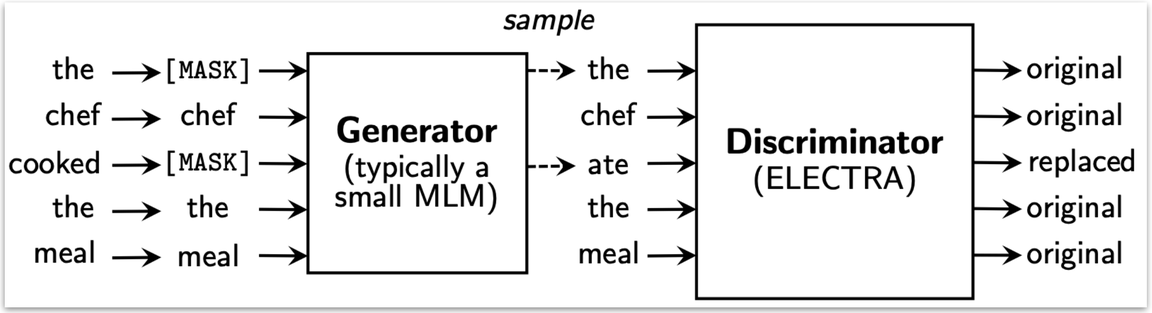


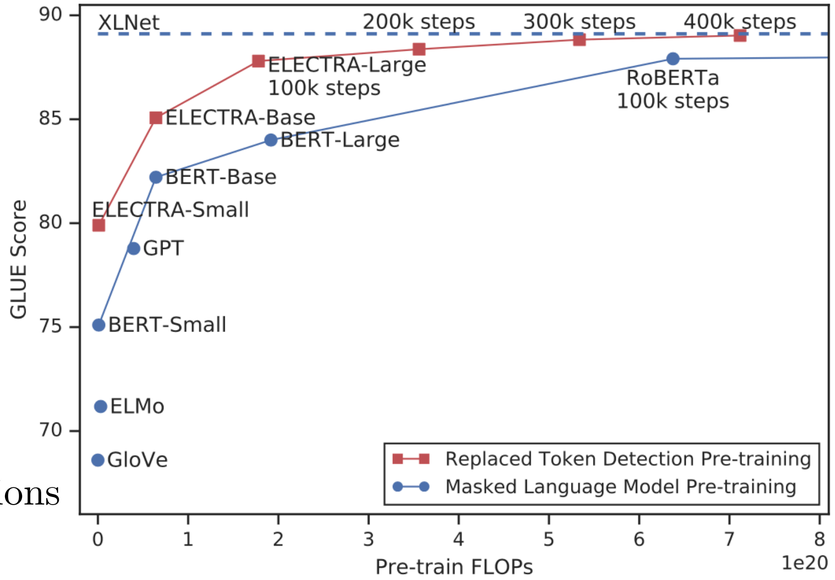
– Common Crawl dataset (filtered) – Webtext dataset
– Two internet-based books corpora – English language Wikipedia

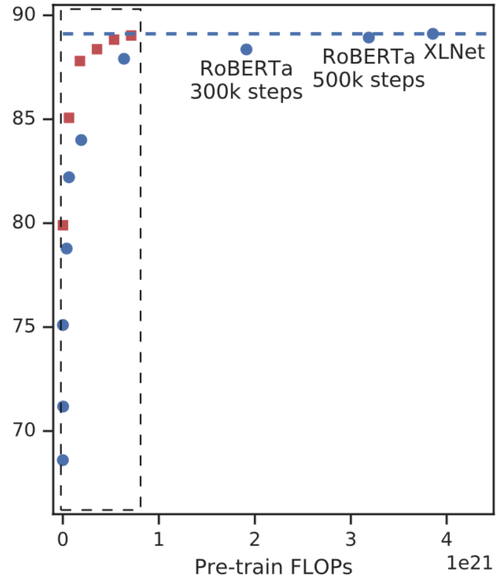
Brown, Tom B., et al. "Language models are few-shot learners." *arXiv preprint arXiv:2005.14165* (2020).

ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators

Masked Language Modeling (MLM) approaches incur a substantial compute cost because the network only learns from 15% of the tokens per example! Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA)







$G \rightarrow$ generator

$D \rightarrow$ discriminator

$x = [x_1, \ldots, x_n] \rightarrow$ sequence of input tokens

$h(x) = [h_1, \ldots, h_n] \rightarrow$ sequence of contextualized vector representations

$h \rightarrow$ encoder (transformer network)

$p_G(x_t|x) = \exp(e(x_t)^T h_G(x)_t) / \sum_{x'} \exp(e(x')^T h_G(x)_t)$

$e(x_t) \rightarrow$ token embedding

$D(x,t) = \text{sigmoid}(w^T h_D(x_t)) \rightarrow$ probability of token x_t being "real"

$m = [m_1, \ldots, m_k] \rightarrow$ random set of positions (integers between 1 and n) to mask out

$k = \lceil 0.15n \rceil \rightarrow 15\%$ of tokens are masked out

$m_i \sim \text{unif}\{1, n\}$ for $i = 1, \ldots, k$

$x^{\text{mask}} = \text{REPLACE}(x, m, [\text{MASK}])$

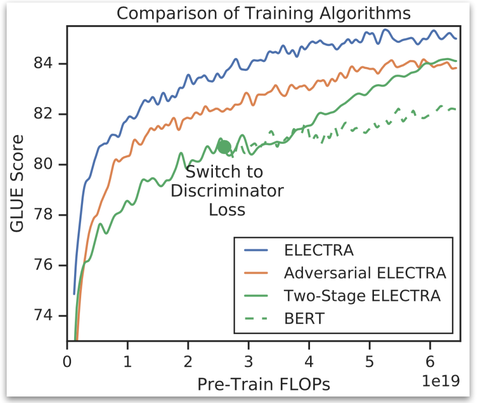
$\hat{x}_i \sim p_G(x_i|x^{\text{mask}})$ for $i \in m$

$x^{\text{corrupt}} = \text{REPLACE}(x, m, \hat{x})$

$$\min_{\theta_G, \theta_D} \sum_{x \in \mathcal{X}} \mathcal{L}_{\text{MLM}}(x, \theta_G) + \lambda \mathcal{L}_{\text{Disc}}(x, \theta_D)$$

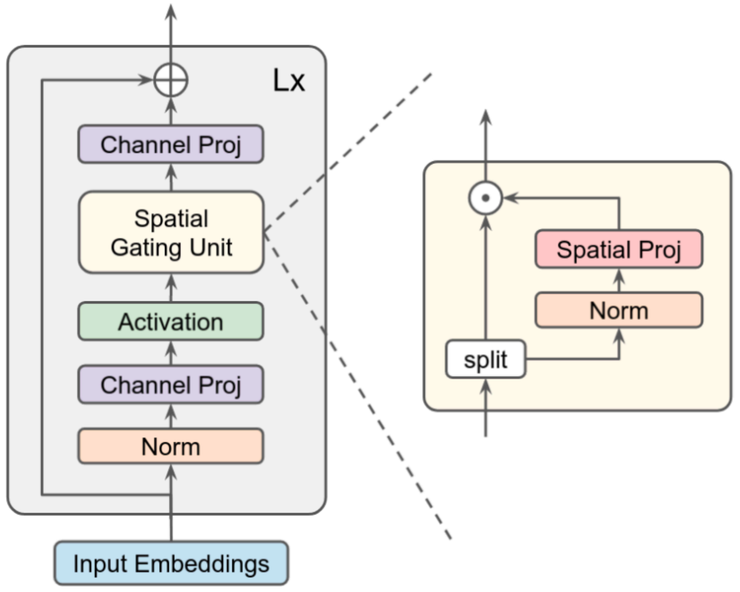
$$\mathcal{L}_{\text{MLM}}(x, \theta_G) = \mathbb{E}\left(\sum_{i \in m} -\log p_G(x_i|x^{\text{mask}})\right)$$

$$\mathcal{L}_{\text{Disc}}(x, \theta_D) = \mathbb{E}\left(\sum_{t=1}^{n} -\mathbb{1}(x_t^{\text{corrupt}} = x_t) \log D(x^{\text{corrupt}}, t) - \mathbb{1}(x_t^{\text{corrupt}} \neq x_t) \log(1 - D(x^{\text{corrupt}}, t))\right)$$



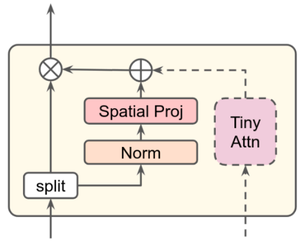
Clark, Kevin, et al. "Electra: Pre-training text encoders as discriminators rather than generators." *arXiv preprint arXiv:2003.10555* (2020).

Pay Attention to MLPs



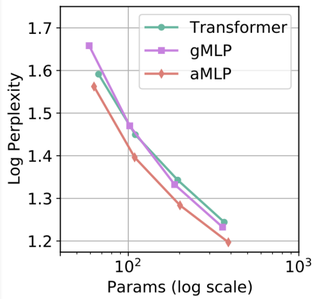
Pseudo-code for the gMLP block

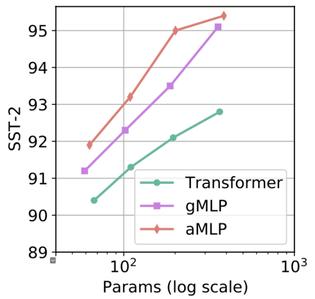
```
def gmlp_block(x, d_model, d_ffn):
  shortcut = x
  x = norm(x, axis="channel")
  x = proj(x, d_ffn, axis="channel")
  x = gelu(x)
  x = spatial_gating_unit(x)
  x = proj(x, d_model, axis="channel")
  return x + shortcut

def spatial_gating_unit(x):
  u, v = split(x, axis="channel")
  v = norm(v, axis="channel")
  n = get_dim(v, axis="spatial")
  v = proj(v, n, axis="spatial", init_bias=1)
  return u * v
```

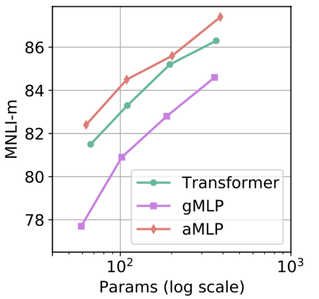


Pseudo-code for the tiny attention module

```
def tiny_attn(x, d_ffn, d_attn=64):
  qkv = proj(x, 3 * d_attn, axis="channel")
  q, k, v = split(qkv, 3, axis="channel")
  w = einsum("bnd,bmd->bnm", q, k)
  a = softmax(w * rsqrt(d_attn))
  x = einsum("bnm,bmd->bnd", a, v)
  return proj(x, d_ffn, axis="channel")
```







L blocks with identical size and structure

$X \in \mathbb{R}^{n \times d} \rightarrow$ token representations

$n \rightarrow$ sequence length

$d \rightarrow$ dimension

For brevity, let's omit shortcuts, normalizations and biases!

$Z = \sigma(XU), \sigma \rightarrow$ GeLU (Gaussian error Linear Unit)

$\tilde{Z} = s(Z) \rightarrow$ spatial interactions

$Y = \tilde{Z}V$

$U, V \rightarrow$ linear projections across the channel dimension

Spatial Gating Unit (SGU)

cross-token interactions

$f_{W,b}(Z) = WZ + b, W \in \mathbb{R}^{n \times n}$

$$Z = (Z_1, Z_2)$$

$$s(Z) = Z_1 \odot f_{W,b}(Z_2)$$

gMLP: MLP with gating

aMLP: gMLP with a small attention

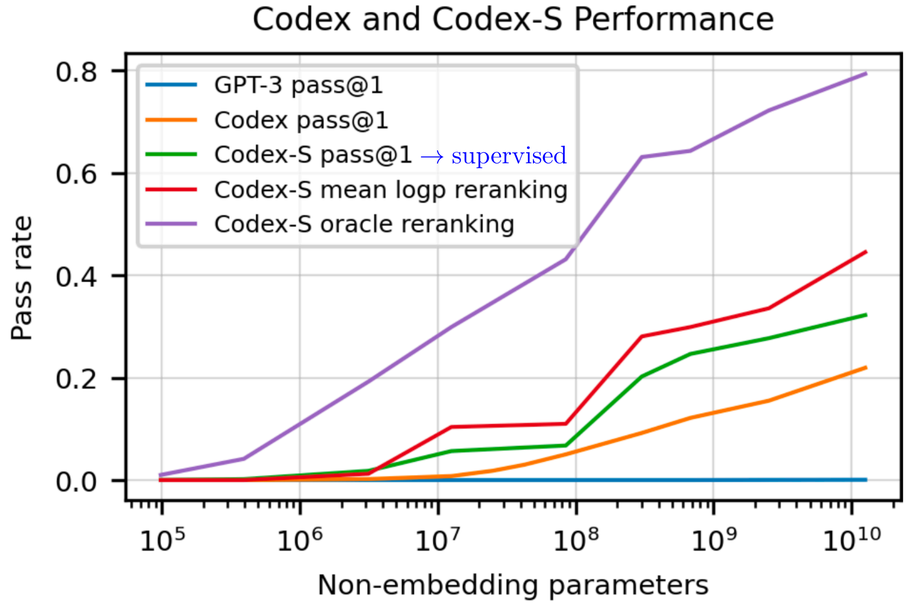
	Perplexity	SST-2	MNLI (m/mm)	SQuAD v1.1	SQuAD v2.0	Attn Size	Params (M)
$\text{BERT}_{\text{base}}$ [2]	–	92.7	84.4/-	88.5	76.3	768 (64 × 12)	110
$\text{BERT}_{\text{base}}$ (ours)	4.17	93.8	85.6/85.7	90.2	78.6	768 (64 × 12)	110
$\text{gMLP}_{\text{base}}$	4.28	94.2	83.7/84.1	86.7	70.1	–	130
$\text{aMLP}_{\text{base}}$	3.95	93.4	85.9/85.8	90.7	80.9	64	109
$\text{BERT}_{\text{large}}$ [2]	–	93.7	86.6/-	90.9	81.8	1024 (64 × 16)	336
$\text{BERT}_{\text{large}}$ (ours)	3.35	94.3	87.0/87.4	92.0	81.0	1024 (64 × 16)	336
$\text{gMLP}_{\text{large}}$	3.32	94.8	86.2/86.5	89.5	78.3	–	365
$\text{aMLP}_{\text{large}}$	3.19	94.8	88.4/88.4	92.2	85.4	128	316

Liu, Hanxiao, et al. "Pay Attention to MLPs." *arXiv preprint arXiv:2105.08050* (2021).

Evaluating Large Language Models Trained on Code

Codex: A GPT language model fine-tuned on publicly available code from GitHub
GitHub Copilot: Powered by a distinct production version of Codex
Program synthesis: Generating programs from docstrings

```python
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```



Limitations

Codex is not sample efficient to train. It is trained on hundreds of millions of lines of code.

Codex can recommend syntactically incorrect or undefined code, and can invoke functions, variables, and attributes that are undefined or outside the scope of the codebase. Moreover, Codex struggles to parse through increasingly long and higher-level or system-level specifications. Codex can make mistakes binding operations to variables, especially when the number of operations and variables in the docstring is large.

```python
def do_work(x, y, z, w):
    """ Add 3 to y, then subtract 4
    from both x and w. Return the
    product of the four numbers. """
    t = y + 3
    u = x - 4
    v = z * w
    return v
```

A system is misaligned if there's some task X that we want it to do, and it is "capable" of doing X but "chooses" not to. In contrast, if a system fails to do X because it does not have the ability to do so, then this system is not misaligned; it is just incompetent.

Evaluation Framework

HumanEval dataset: 164 hand-written programming problems

https://www.github.com/openai/human-eval

Functional correctness (i.e., passing a set of unit tests) v.s. match-based metrics (e.g., CodeBLEU score)

pass@k → k code samples are generated per problem, a problem is considered solved if any sample passes the unit tests, and the total fraction of problems solved is reported.

This could have high-variance. Instead, an unbiased estimator is used (see the paper!).

Fine Tuning on Code

GPT-3 text tokenizer in addition to a set of tokens for representing whitespace runs of different lengths

stop sequences: '`\nclass`', '`\ndef`', '`\n#`', '`\nif`', **or** '`\nprint`'

Nucleus sampling instead of beam search

Chen, Mark, et al. "Evaluating large language models trained on code." *arXiv preprint arXiv:2107.03374* (2021).

Boulder

Questions?

YouTube Playlist