BERT: model, analysis and modifications

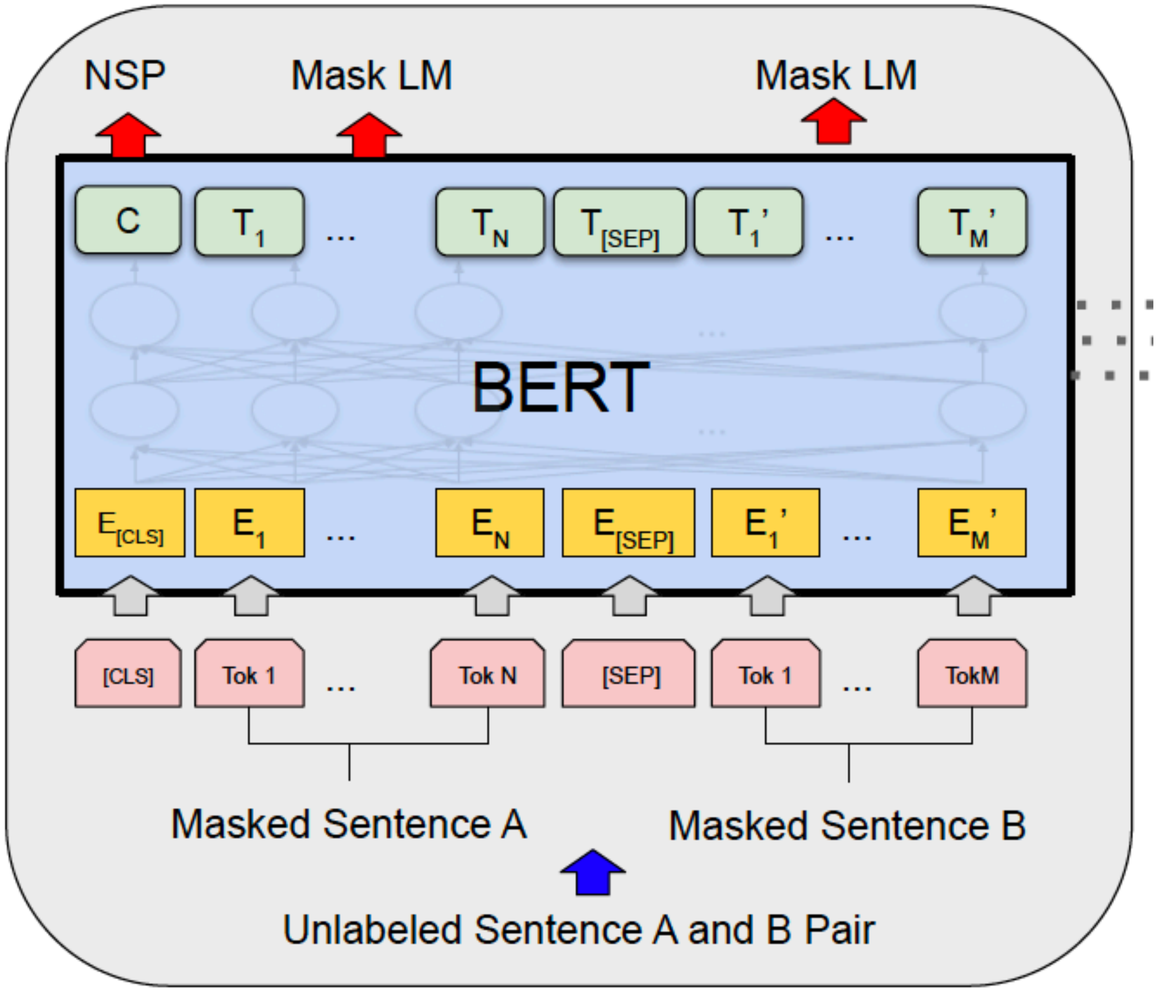
Ekaterina Lobacheva

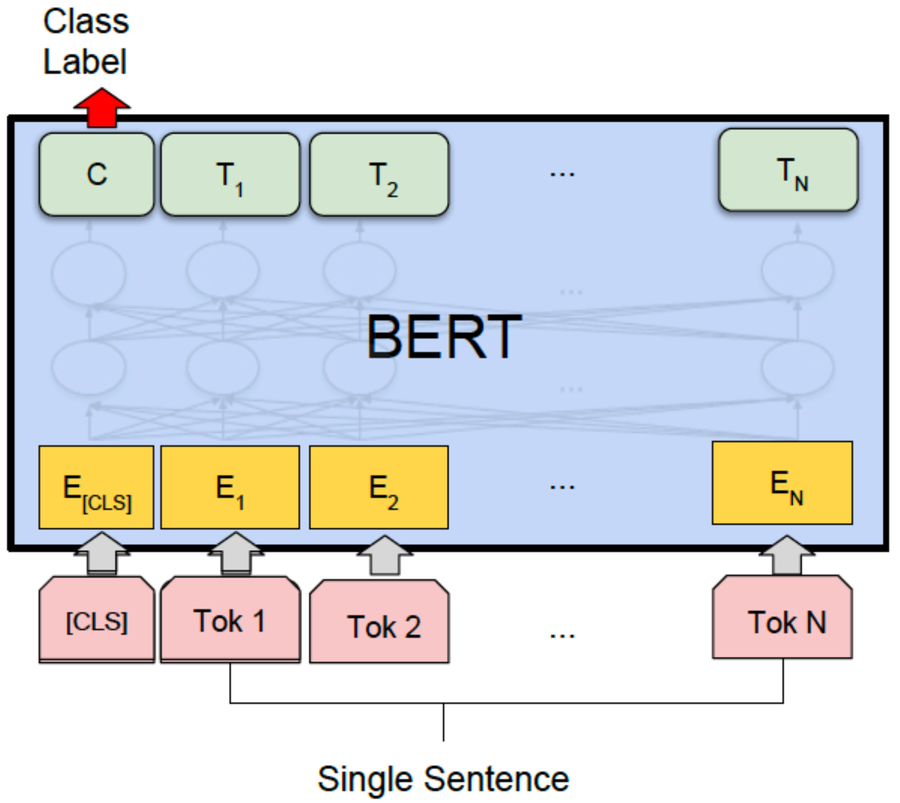
NATIONAL RESEARCH UNIVERSITY

SAMSUNG Research

p(B|A)yesgroup.ru

BERT: pre-training and fine-tuning

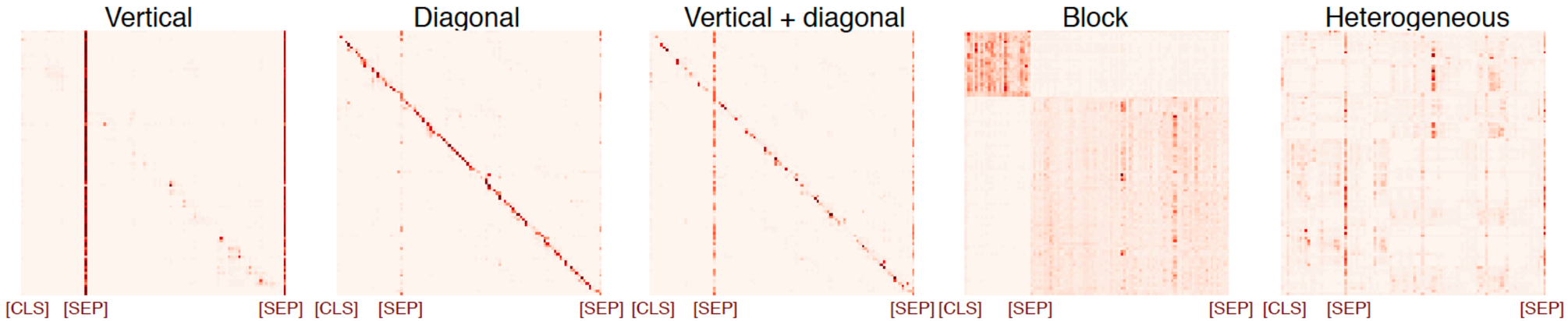




(b) Single Sentence Classification Tasks:
SST-2, CoLA

BERT’s self-attention patterns

BERT's self-attention patterns



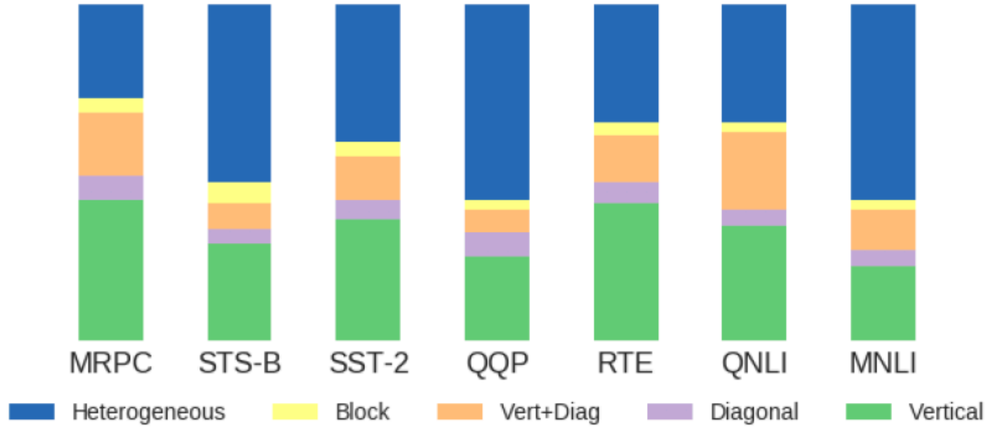
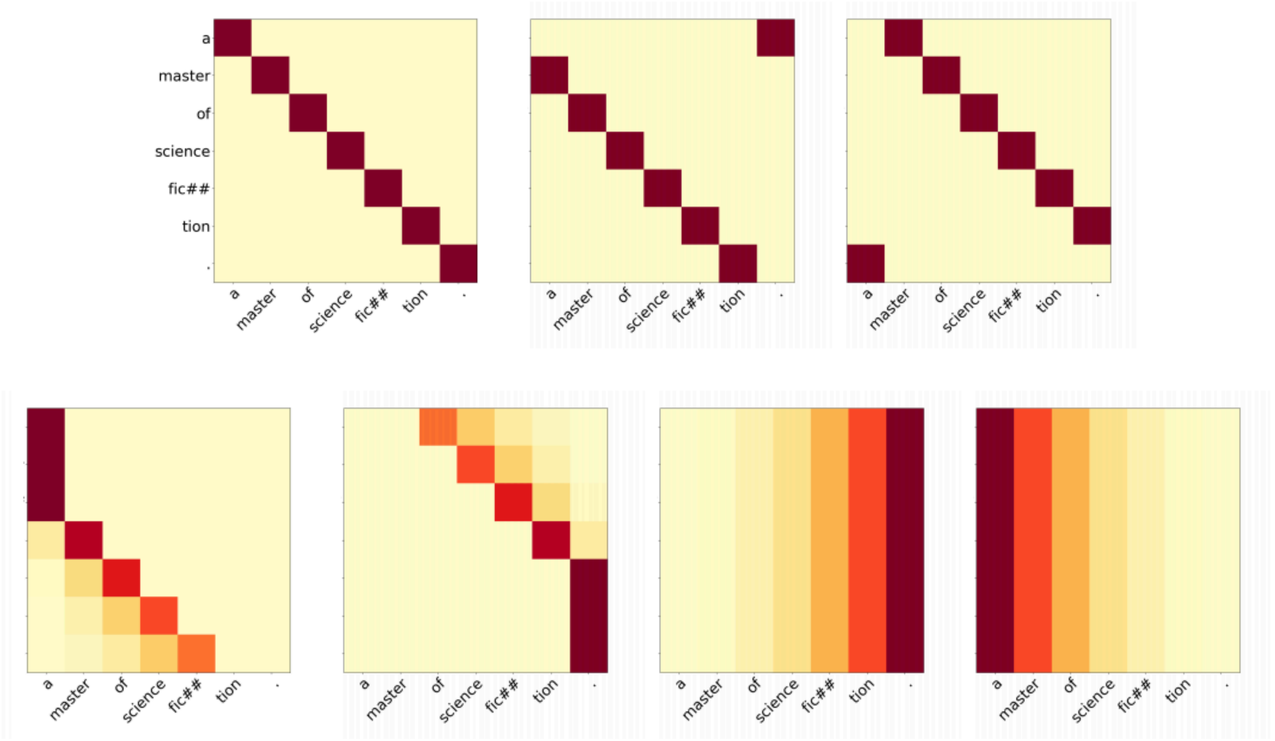


Figure 2: Estimated percentages of the identified self-attention classes for each of the selected GLUE tasks.

A lot of self-attention heads do not directly encode any nontrivial linguistic information

Fixed Encoder Self-Attention in Transformer



+ 1 trainable head

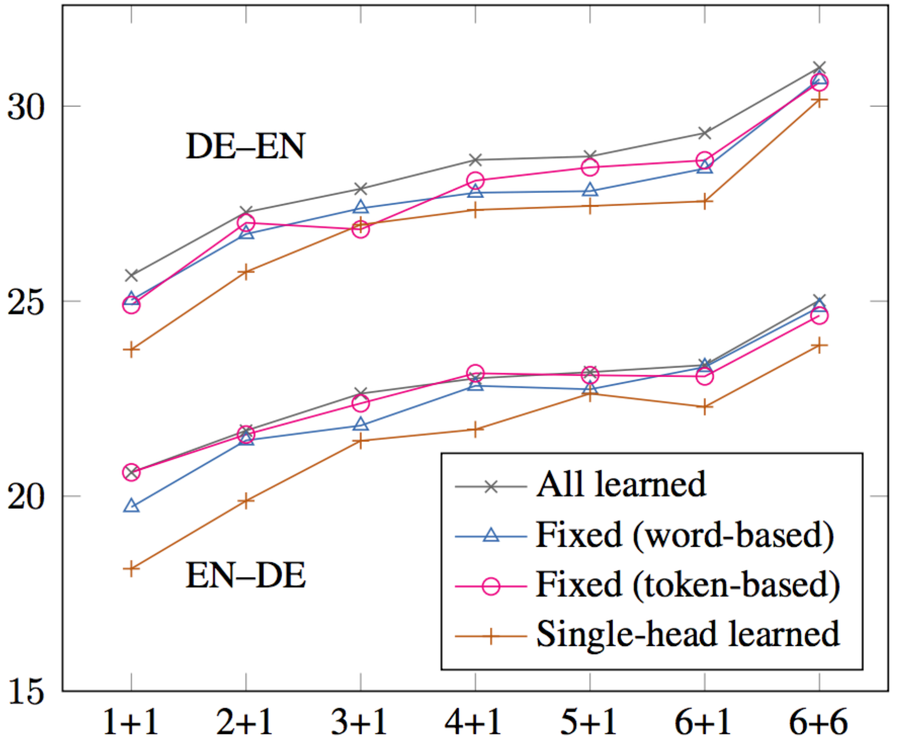
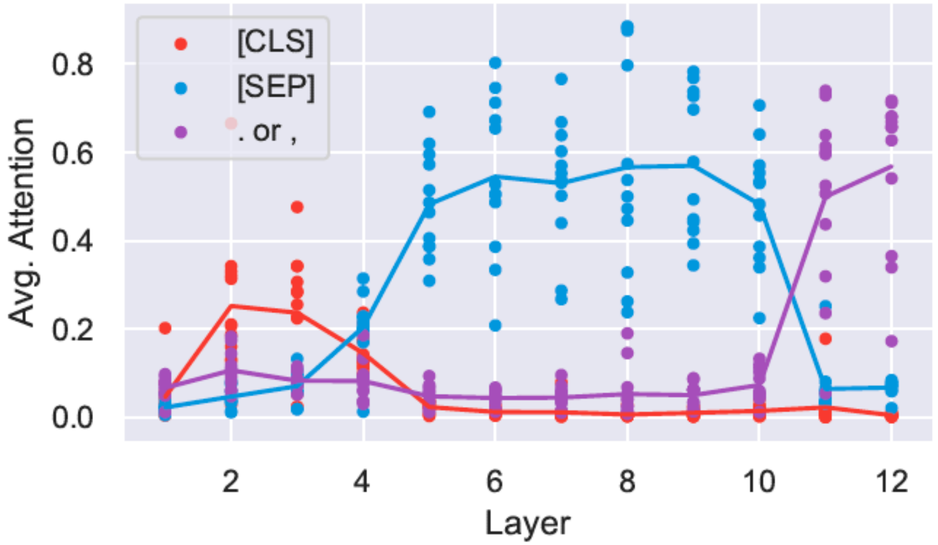


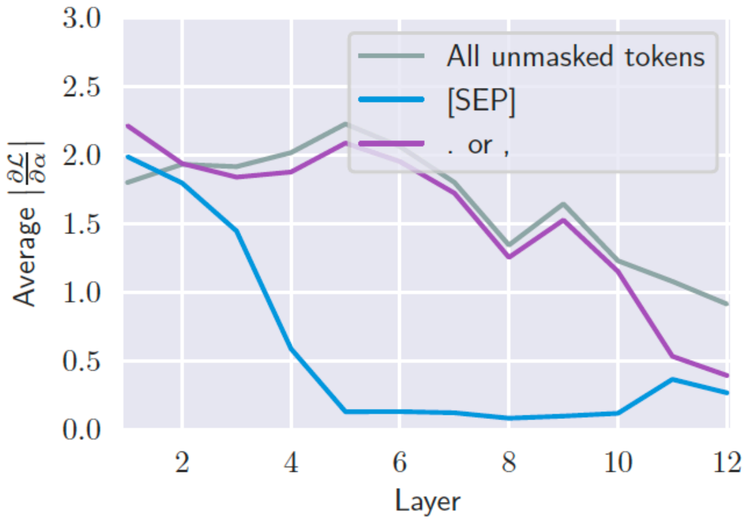
Figure 2: BLEU scores for the German ↔ English standard scenario. The x-axis shows different configurations of encoder and decoder layers.

BERT's self-attention patterns

A lot of attention to CLS, SEP, and punctuation tokens

- CLS - summary
- SEP - no-op (look at SEP if this head relation is not applicable)
- punctuation - similar to SEP???





Attention identifiability

Output of one head from MHA: $\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\boldsymbol{H} = \boldsymbol{A}\boldsymbol{E}\boldsymbol{W}^V\boldsymbol{H} = \boldsymbol{A}\boldsymbol{T}$

$$\begin{aligned}\text{rank}\,(\boldsymbol{T}) &\leq \min\left(\text{rank}(\boldsymbol{E}), \text{rank}(\boldsymbol{W}^V), \text{rank}(\boldsymbol{H})\right) \\ &\leq \min(d_s, d, d, d_v, d_v, d) \\ &= \min\left(d_s, d_v\right).\end{aligned}$$

BERT:
d_s = 512
d_v = 64

Null space of T:

$$\dim(\text{LN}(\boldsymbol{T})) = d_s - \text{rank}\,(\boldsymbol{T}) \geq d_s - \min\left(d_s, d_v\right) = \begin{cases} d_s - d_v, & \text{if } d_s > d_v \\ 0, & \text{otherwise} \end{cases}$$

+ they take into account probability constraints

Effective attention: $\boldsymbol{A}^{\perp} = \boldsymbol{A} - \text{Projection}_{\text{LN}(T)}\boldsymbol{A}.$

Attention identifiability

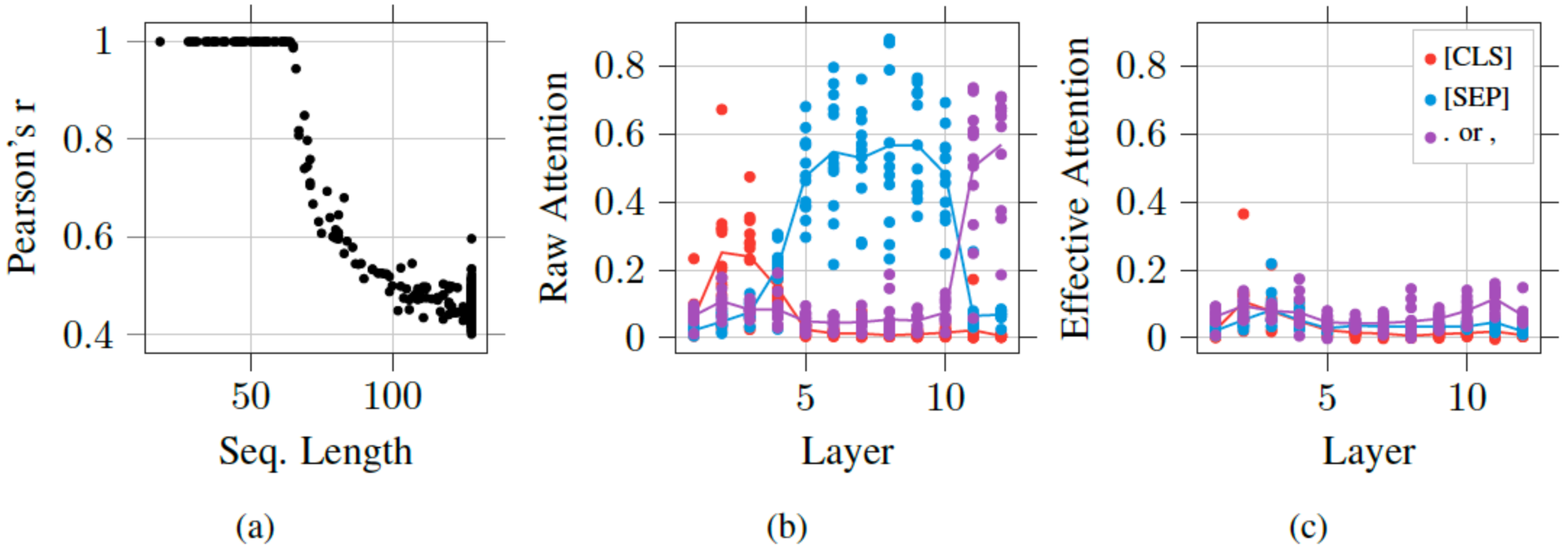
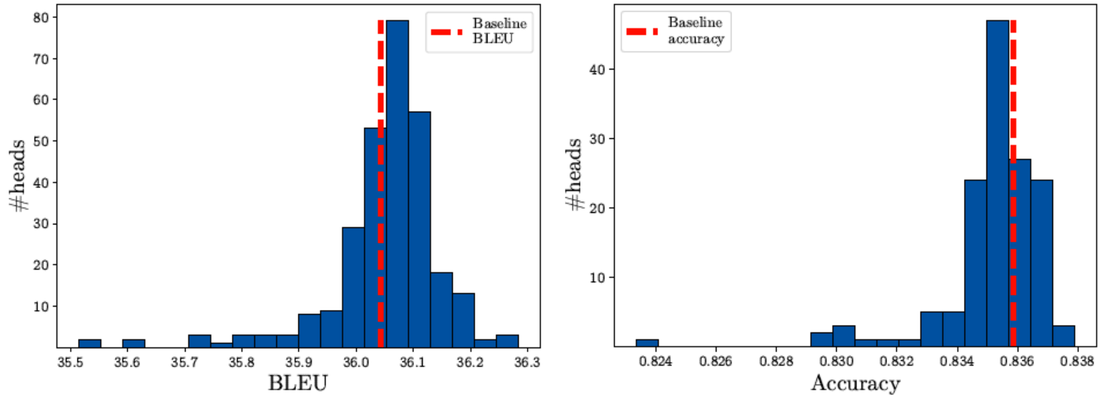


Figure 1: (a) Each point represents the Pearson correlation coefficient of effective attention and raw attention as a function of token length. (b) Raw attention vs. (c) effective attention, where each point represents the average (effective) attention of a given head to a token type.

Disabling self-attention heads

$$\text{MHAtt}(\mathbf{x}, q) = \sum_{h=1}^{N_h} \xi_h \text{Att}_{W_k^h, W_q^h, W_v^h, W_o^h}(\mathbf{x}, q)$$

Disable one head:



(a) WMT (b) BERT

Figure 1: Distribution of heads by model score after masking.

Disable all heads but one in a layer:

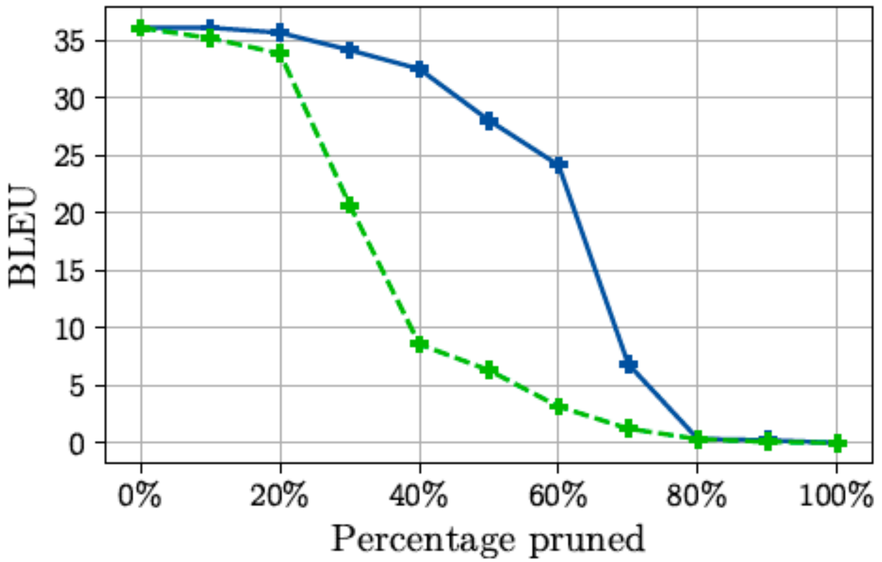
Layer	Enc-Enc	Enc-Dec	Dec-Dec
1	**-1.31**	**0.24**	-0.03
2	-0.16	0.06	0.12
3	0.12	0.05	0.18
4	-0.15	-0.24	0.17
5	0.02	**-1.55**	-0.04
6	**-0.36**	**-13.56**	0.24

Table 2: Best delta BLEU by layer when only one head is kept in the WMT model. Underlined numbers indicate that the change is statistically significant with $p < 0.01$.

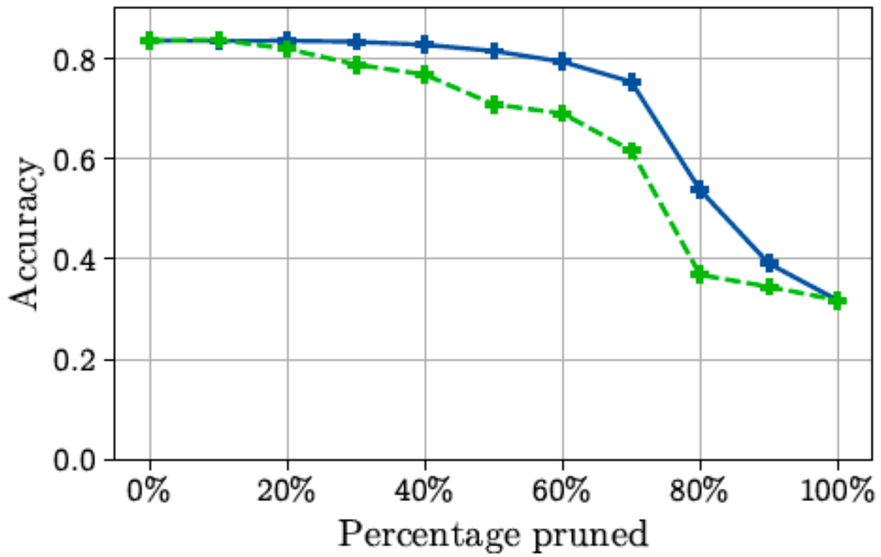
Layer		Layer	
1	-0.01%	7	0.05%
2	0.10%	8	-0.72%
3	-0.14%	9	-0.96%
4	-0.53%	10	0.07%
5	-0.29%	11	-0.19%
6	-0.52%	12	-0.12%

Table 3: Best delta accuracy by layer when only one head is kept in the BERT model. None of these results are statistically significant with $p < 0.01$.

Iterative Pruning of Attention Heads



(a) Evolution of BLEU score on `newstest2013` when heads are pruned from WMT.



(b) Evolution of accuracy on the MultiNLI-matched validation set when heads are pruned from BERT.

Figure 3: Evolution of accuracy by number of heads pruned according to I_h (solid blue) and individual oracle performance difference (dashed green).

BERTs friends

What to change?

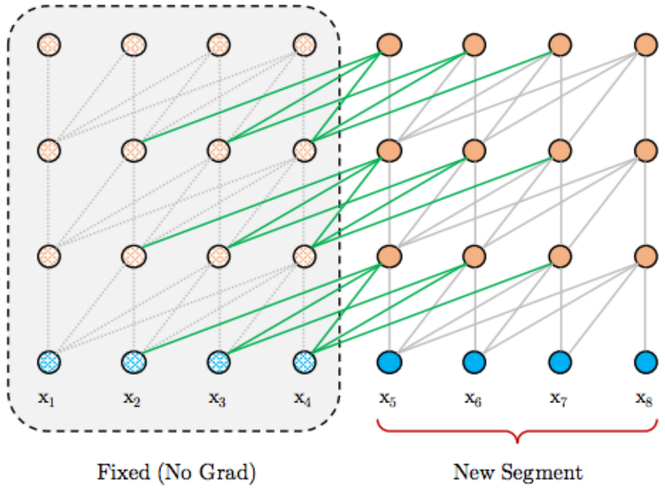
- Longer training
- Better optimization
- More data

What to change?

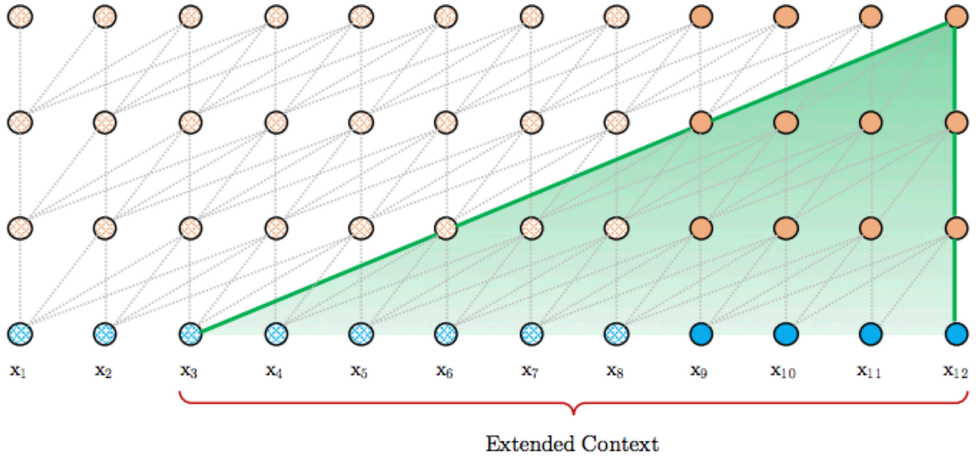
Transformer type:

- Transformer-XL in XLNet

Training



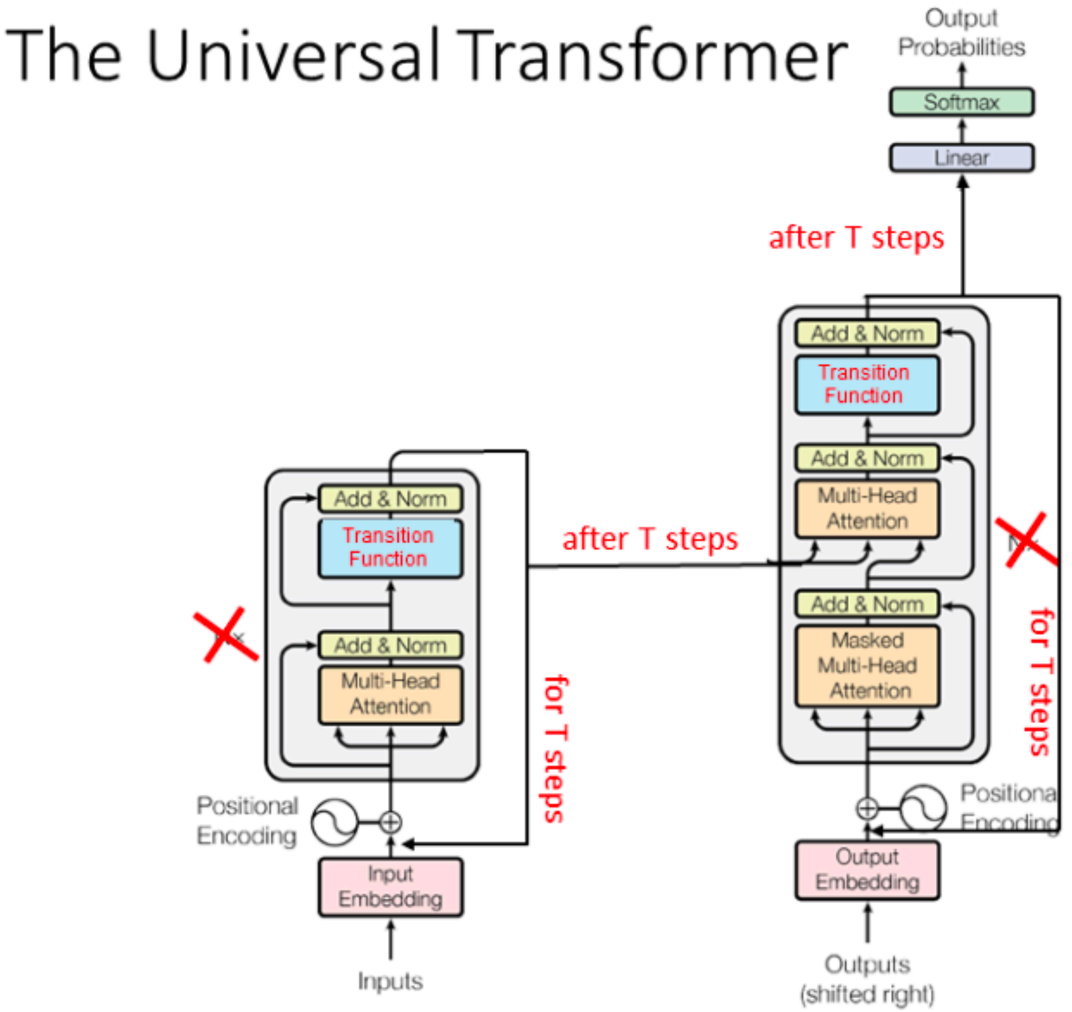
Evaluation



What to change?

Transformer type:

- Transformer-XL in XLNet
- Universal Transformer in ALBERT



https://glassboxmedicine.com/2019/09/07/universal-transformers/

What to change?

NSP objective:

- None in RoBERTa, SpanBERT
- Sentence-order prediction (SOP) in ALBERT
- Predict both previous and next sentences in StructBERT

What to change?

MLM objective:

- Dynamic masking in RoBERTa
- Phrase masking and named entity masking in ERNIE

Harry Potter is a series of fantasy novel written by J. K. Rowling

```
- Learned by BERT : [mask] Potter is a series [mask] fantasy novel [mask] by J. [mask] Rowling
```

```
- Learned by ERNIE: Harry Potter is a series of [mask] [mask] written by [mask] [mask] [mask]
```

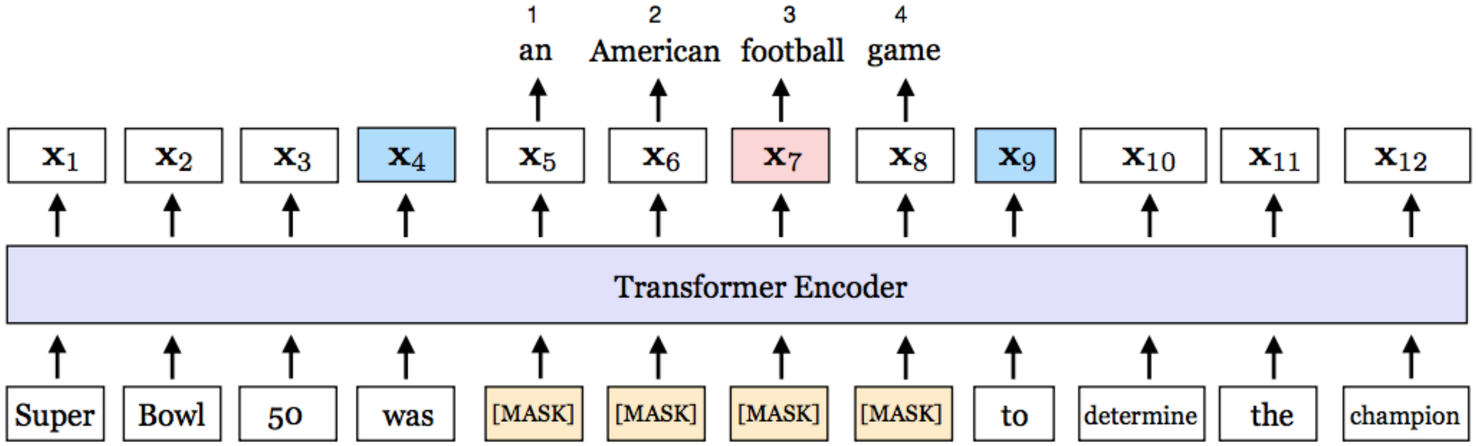
https://github.com/PaddlePaddle/ERNIE#ernie-10-enhanced-representation-through-knowledge-integration

What to change?

MLM objective:

- Dynamic masking in RoBERTa
- Phrase masking and named entity masking in ERNIE
- Span boundary objective in SpanBERT

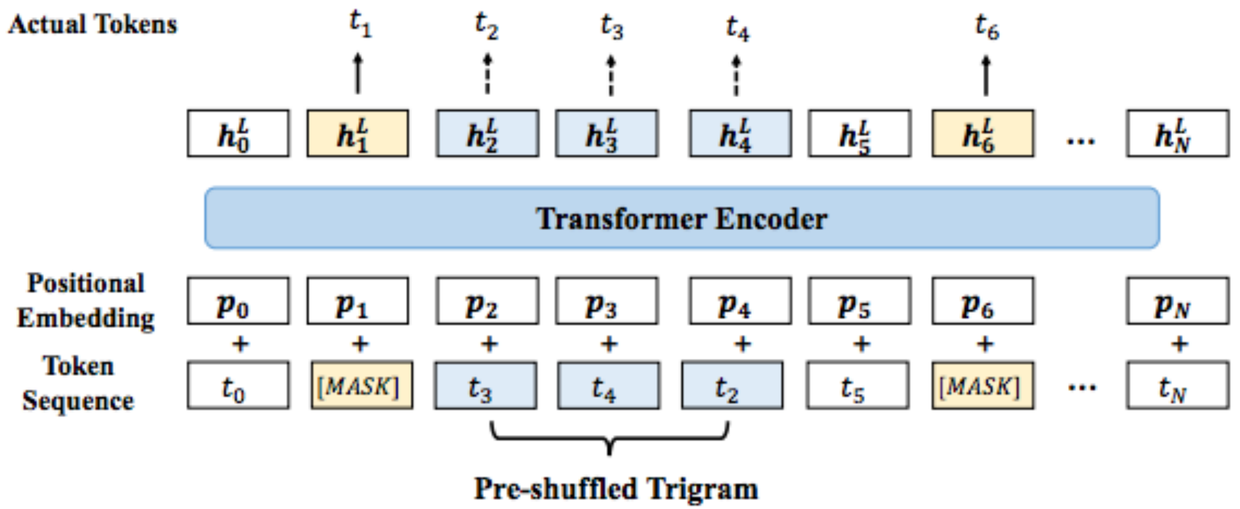
$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$
$$= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)$$



What to change?

MLM objective:

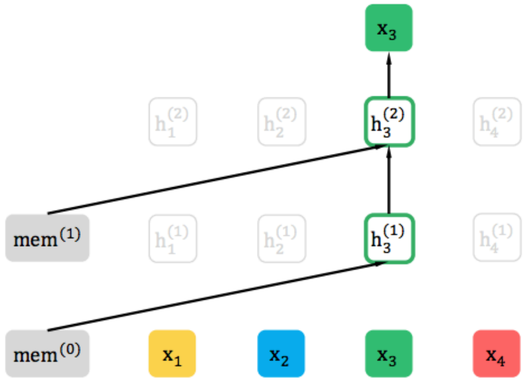
- Dynamic masking in RoBERTa
- Phrase masking and named entity masking in ERNIE
- Span boundary objective in SpanBERT
- Predict right word order in StructBERT



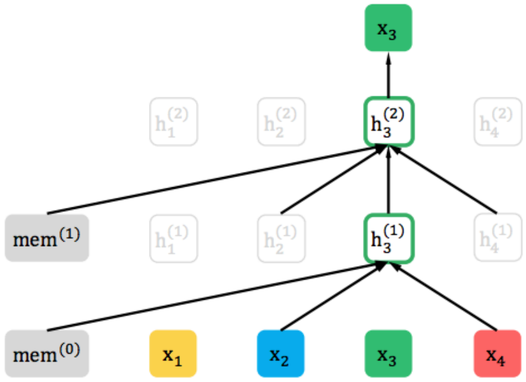
What to change?

MLM objective:

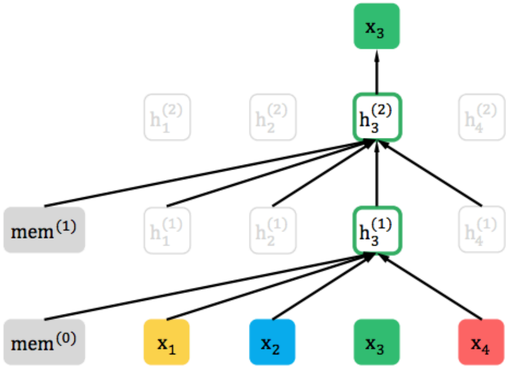
- Dynamic masking in RoBERTa
- Phrase masking and named entity masking in ERNIE
- Span boundary objective in SpanBERT
- Predict right word order in StructBERT
- Permutation language modeling in XLNet



Factorization order: 3 → 2 → 4 → 1



Factorization order: 2 → 4 → 3 → 1

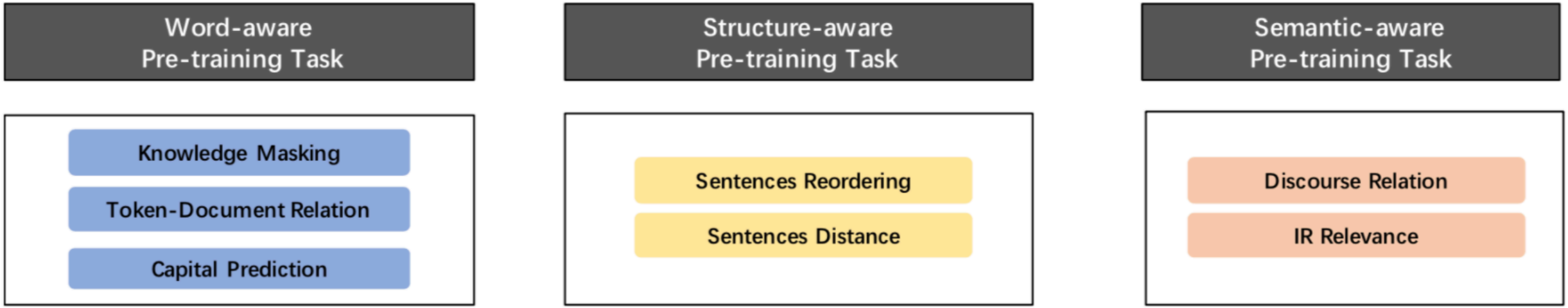


Factorization order: 1 → 4 → 2 → 3

What to change?

Different objective:

- Multi-task learning in ERNIE 2.0



https://github.com/PaddlePaddle/ERNIE

What to change?

Different objective:

- Multi-task learning in ERNIE 2.0
- + Discriminator for replaced token detection in ELECTRA

What to change?

Pre-training procedure:

- Recursive pertaining

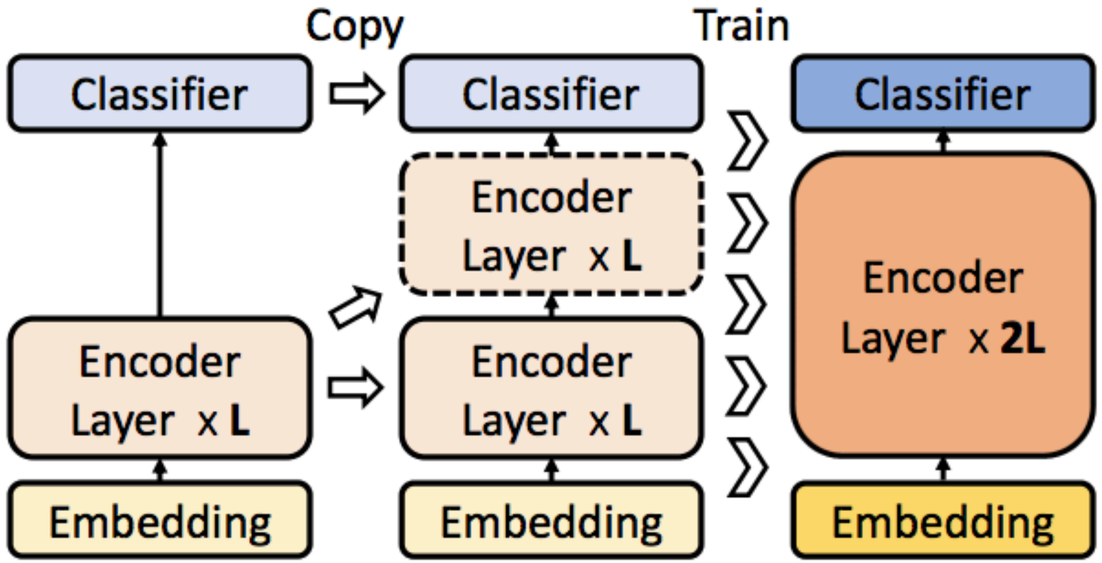


Figure 3. The diagram of the *stacking* algorithm.

BERT compression

		Compression	Performance	Speedup	Model	Evaluation
Distillation	DistilBERT (Sanh et al., 2019)	×2.5	90%	×1.6	$BERT_6$	All GLUE tasks
	$BERT_6$-PKD (Sun et al., 2019a)	×1.6	97%	×1.9	$BERT_6$	No WNLI, CoLA and STS-B
	$BERT_3$-PKD (Sun et al., 2019a)	×2.4	92%	×3.7	$BERT_3$	No WNLI, CoLA and STS-B
	(Aguilar et al., 2019)	×2	94%	-	$BERT_6$	CoLA, MRPC, QQP, RTE
	BERT-48 (Zhao et al., 2019)	×62	87%	×77	$BERT_{12}^{*\dagger}$	MNLI, MRPC, SST-2
	BERT-192 (Zhao et al., 2019)	×5.7	94%	×22	$BERT_{12}^{*\dagger}$	MNLI, MRPC, SST-2
	TinyBERT (Jiao et al., 2019)	×7.5	96%	×9.4	$BERT_4^{*\dagger}$	All GLUE tasks
	MobileBERT (Sun et al.)	×4.3	100%	×4	$BERT_{24}^{\dagger}$	No WNLI
	PD (Turc et al., 2019)	×1.6	98%	$\times 2.5^3$	$BERT_6^{\dagger}$	No WNLI, CoLA and STS-B
	MiniBERT(Tsai et al., 2019)	$\times 6^{\S}$	98%	$\times 27^{\S}$	$mBERT_3^{\dagger}$	CoNLL-2018 POS and morphology
	BiLSTM soft (Tang et al., 2019)	×110	91%	$\times 434^{\ddagger}$	$BiLSTM_1$	MNLI, QQP, SST-2
Quant.	Q-BERT (Shen et al., 2019)	×13	99%	-	$BERT_{12}$	MNLI, SST-2
	Q8BERT (Zafrir et al., 2019)	×4	99%	-	$BERT_{12}$	All GLUE tasks
Other	ALBERT-base (Lan et al., 2019)	×9	97%	×5.6	$BERT_{12}^{**}$	MNLI, SST-2
	ALBERT-xxlarge (Lan et al., 2019)	×0.47	107%	×0.3	$BERT_{12}^{**}$	MNLI, SST-2
	BERT-of-Theseus (Xu et al., 2020)	×1.6	98%	-	$BERT_6$	No WNLI

Table 1: Comparison of BERT compression studies. Compression, performance retention, and inference time speedup figures are given with respect to $BERT_{base}$, unless indicated otherwise. Performance retention is measured as a ratio of average scores achieved by a given model and by $BERT_{base}$. The subscript in the model description reflects the number of layers used. *Smaller vocabulary used. †The dimensionality of the hidden layers is reduced. **The dimensionality of the embedding layer is reduced. ‡Compared to $BERT_{large}$. §Compared to mBERT.

RoBERTa

A Robustly Optimized BERT Pretraining Approach

ICLR 2020

reject

RoBERTa: overview

- training the model longer: 31K(1M with small batches) -> 500K
- bigger batches: 256 sequences -> 8K
- more training data: 16GB -> 160GB
- larger BPE vocabulary: 30K -> 50K
- training on longer sequences: always 512

- removing the next sentence prediction objective
- dynamically changing the masking pattern applied to the training data

Result: longer training, the same size model, better performance

RoBERTa: dynamic masking

BERT:

- training data is duplicated 10 times
- static masking - generate one mask for each training instance before training

RoBERTa:

- dynamic masking - masks are generated at every epoch

Important for long training!

RoBERTa: no NSP objective

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

- removing the NSP loss matches or slightly improves performance
- BERT: removed the NSP while still retaining the SEGMENT-PAIR input format
- RoBERTa use FULL-SENTENCES to have consistent batch sizes

RoBERTa: GLUE

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
$\text{BERT}_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
$\text{XLNet}_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**	**91.3**	-
Ensembles on test (from leaderboard as of July 25, 2019)										
ALICE	88.2/87.9	95.7	**90.7**	83.5	95.2	92.6	**68.6**	91.1	80.8	86.3
MT-DNN	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2/89.8	98.6	90.3	86.3	**96.8**	**93.0**	67.8	91.6	**90.4**	88.4
RoBERTa	**90.8/90.2**	**98.9**	90.2	**88.2**	96.7	92.3	67.8	**92.2**	89.0	**88.5**

Table 5: Results on GLUE. All results are based on a 24-layer architecture. $\text{BERT}_{\text{LARGE}}$ and $\text{XLNet}_{\text{LARGE}}$ results are from Devlin et al. (2019) and Yang et al. (2019), respectively. RoBERTa results on the development set are a median over five runs. RoBERTa results on the test set are ensembles of *single-task* models. For RTE, STS and MRPC we finetune starting from the MNLI model instead of the baseline pretrained model. Averages are obtained from the GLUE leaderboard.

ALBERT

A Lite BERT

ICLR 2020

ALBERT: overview

- factorized embedding parameterization:
  (V x H) to (V x E) and (E x H)
- cross-layer parameter sharing:
  share all parameters as in Universal Transformer or DEQ
- inter-sentence coherence loss:
  NSP is too easy as compared to MLM
  NSP -> sentence-order prediction (SOP) - IsNext or IsPrevious

Result: smaller/faser/worse performance or smaller/slower/better performance

ALBERT vs BERT

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

Table 2: Dev set results for models pretrained over BOOKCORPUS and Wikipedia for 125k steps. Here and everywhere else, the Avg column is computed by averaging the scores of the downstream tasks to its left (the two numbers of F1 and EM for each SQuAD are first averaged).

ALBERT: ablation study

- factorized embedding parameterization:

  hurts performance
- cross-layer parameter sharing:

  hurts performance
- inter-sentence coherence loss:

  SOP > NSP or None

ALBERT: SOTA (+ more data and dropout)

Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
BERT-large	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet-large	89.8	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa-large	90.2	94.7	**92.2**	86.6	96.4	**90.9**	68.0	92.4	-	-
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7	-	-
ALBERT (1.5M)	**90.8**	**95.3**	**92.2**	**89.2**	**96.9**	**90.9**	**71.4**	**93.0**	-	-
Ensembles on test (from leaderboard as of Sept. 16, 2019)										
ALICE	88.2	95.7	**90.7**	83.5	95.2	92.6	**69.2**	91.1	80.8	87.0
MT-DNN	87.9	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2	98.6	90.3	86.3	96.8	93.0	67.8	91.6	90.4	88.4
RoBERTa	90.8	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0	88.5
Adv-RoBERTa	91.1	98.8	90.3	88.7	96.8	93.1	68.0	92.4	89.0	88.8
ALBERT	**91.3**	**99.2**	90.5	**89.2**	**97.1**	**93.4**	69.1	**92.5**	**91.8**	**89.4**

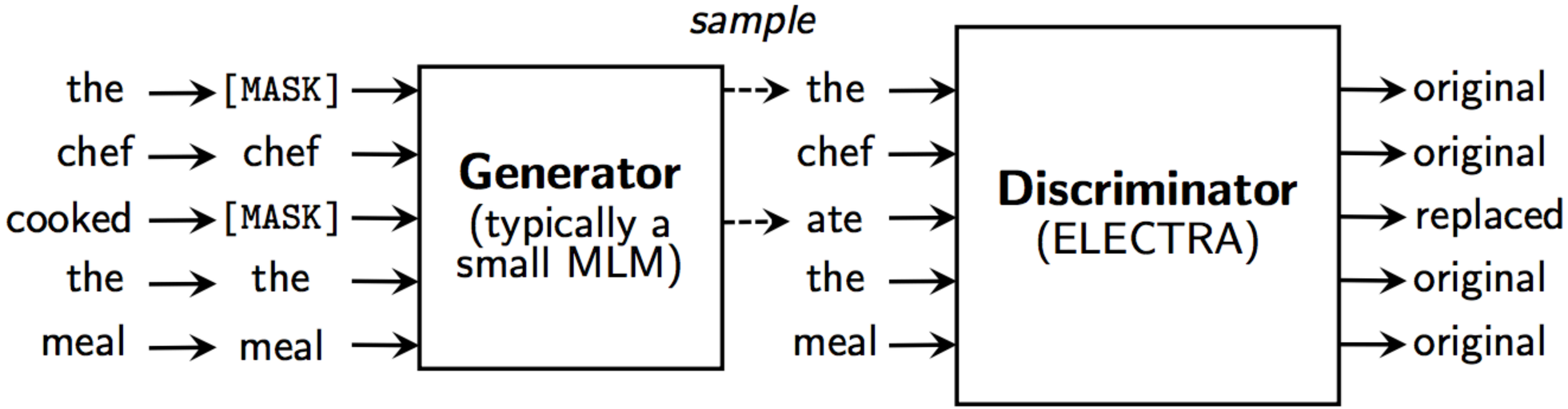
Table 9: State-of-the-art results on the GLUE benchmark. For single-task single-model results, we report ALBERT at 1M steps (comparable to RoBERTa) and at 1.5M steps. The ALBERT ensemble uses models trained with 1M, 1.5M, and other numbers of steps.

ELECTRA

Efficiently Learning an Encoder that Classifies Token Replacements Accurately

ICLR 2020

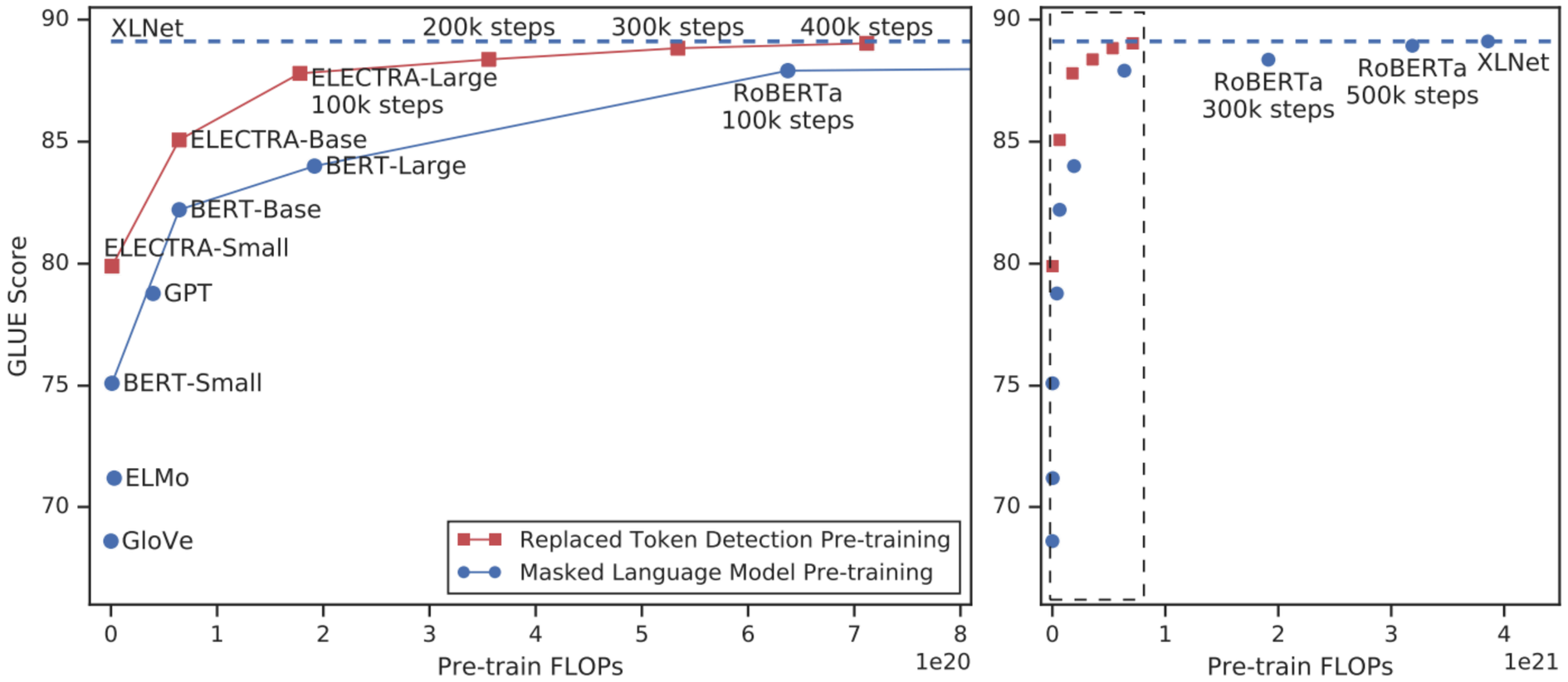
ELECTRA: idea



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

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

ELECTRA: results



- Less gap between training and testing
- Predictions on all tokens -> training is faster









Miles Brundage
@Miles_Brundage

2018: Language model papers have to introduce Sesame Street-related acronyms

2019: Language model papers need Sesame Street jokes in the title, all talks need at least one Sesame Street image.

2020: ACL/NAACL co-located with Sesame Street convention, Big Bird gives a keynote.

293 2:46 AM - Jun 12, 2019

57 people are talking about this

References

- Basics:
  - Transformer - https://arxiv.org/pdf/1706.03762.pdf
  - ELMO - https://arxiv.org/pdf/1802.05365.pdf
  - GPT - https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf
  - BERT - https://arxiv.org/pdf/1810.04805.pdf
- BERTology - https://arxiv.org/pdf/2002.12327.pdf
- Analysis of BERT (and Transformer):
  - https://arxiv.org/pdf/1909.00512.pdf
  - https://arxiv.org/pdf/1909.10430.pdf
  - https://arxiv.org/pdf/1908.08593v2.pdf
  - https://arxiv.org/pdf/1906.04341v1.pdf

References

- Analysis of BERT (and Transformer):
  - https://arxiv.org/pdf/1908.04211.pdf
  - https://arxiv.org/abs/1905.10650
  - https://arxiv.org/pdf/2002.10260.pdf
- RoBERTa - https://arxiv.org/abs/1907.11692
- ALBERT - https://arxiv.org/pdf/1909.11942.pdf
- Other BERT modifications:
  - https://proceedings.mlr.press/v97/gong19a/gong19a.pdf
  - https://arxiv.org/abs/1904.09223
  - https://arxiv.org/abs/1907.12412
  - https://arxiv.org/abs/1907.10529

References

- Other BERT modifications:
  - https://proceedings.mlr.press/v97/gong19a/gong19a.pdf
  - https://arxiv.org/abs/1904.09223
  - https://arxiv.org/abs/1907.12412
  - https://arxiv.org/abs/1907.10529
  - https://arxiv.org/abs/1908.04577