XLNet: Generalized Autoregressive Pretraining for Language Understanding

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V.Le
Carnegie Mellon University, Google Brain

TRANSFORMER-XL: ATTENTIVE LANGUAGE MODELS BEYOND A FIXED-LENGTH CONTEXT

2019.1.18

Zihang Dai*[1], Zhilin Yang*[2], Yiming Yang[1], Jaime Carbonell[1],
Quoc V. Le[2], Ruslan Salakhutdinov[1]
[1]Carnegie Mellon University, [2]Google Brain
{dzihang,yiming,jgc,rsalakhu}@cs.cmu.edu, {zhiliny,qvl}@google.com

XLNet: Generalized Autoregressive Pretraining for Language Understanding

2019.6.19

Zhilin Yang*[1], Zihang Dai*[12], Yiming Yang[1], Jaime Carbonell[1],
Ruslan Salakhutdinov[1], Quoc V. Le[2]
[1]Carnegie Mellon University, [2]Google Brain
{zhiliny,dzihang,yiming,jgc,rsalakhu}@cs.cmu.edu, qvl@google.com

Contents

Recent Trends

- Unsupervised representation learning has been successful!
  - Pretrain on large-scale unlabeled text corpora → finetuning
  - CoVe(2017)*, ELMo(2018)**, GPT(2018)***, BERT(2018)****

- Pretraining methods : Autoregressive vs Autoencoding

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

Autoregressive

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

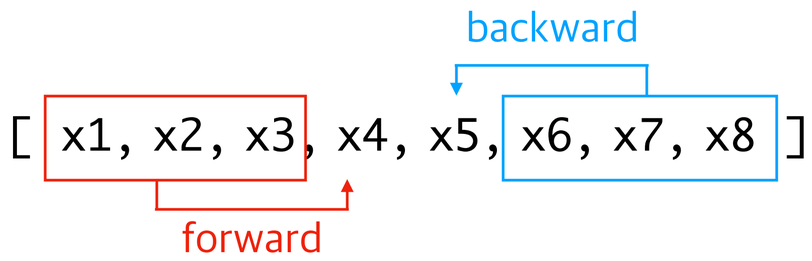
Autoencoding

*McCann et al 2017, Learned in translation: Contextualized word vectors. **Peters et al 2018, Deep contextualized word representations.
***Radford et al 2018, Improving language understanding by generative pre-training ****Devlin et al 2018, Bert: Pre-training of deep bidirectional transformers for language understanding.

Autoregressive vs Autoencoding

- AR language model (GPT)

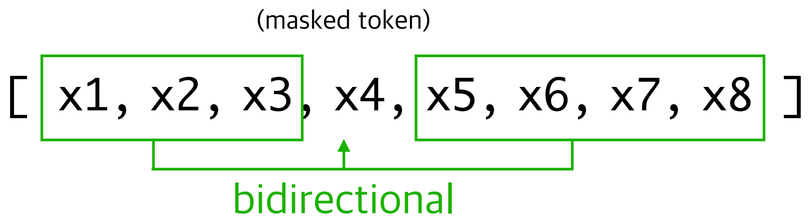
$$\max_{\theta} \log p_\theta(x) = \sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t})$$



Good at text generation

- AE language model (BERT)

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



Good at language understanding

Autoregressive vs Autoencoding

- BERT's limitations
  - Model assumes that all masked tokens are independent

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

  - Generalized model **should not rely on data corruption** (masking)
  - <mask> token doesn't appear in real world
  - It lacks long-term dependency

Changes

Permutation Language Modeling

Two-Stream Self-Attention

Transformer-XL

Permutation Language Modeling

origin sequence [1, 2, 3, 4, 5, 6, 7, 8]

Forward AR

[1, 2, 3, 4, 5, 6, 7, 8]

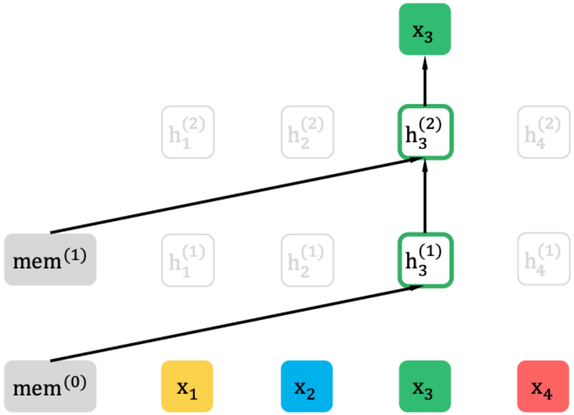
Backward AR

[8, 7, 6, 5, 4, 3, 2, 1]

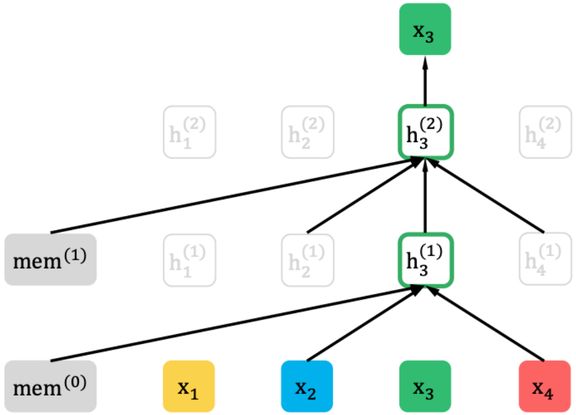
Permutation AR

[3, 2, 5, 6, 8, 1, 7, 4]

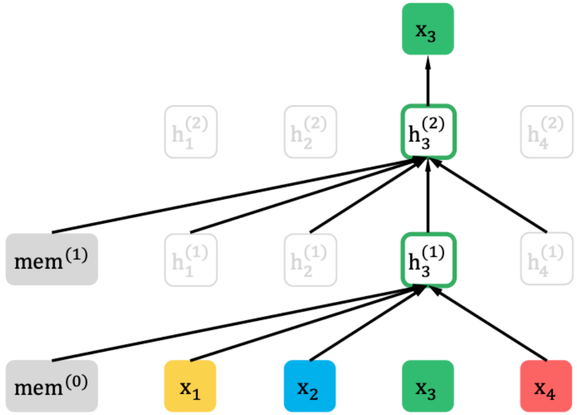
Permutation Language Modeling



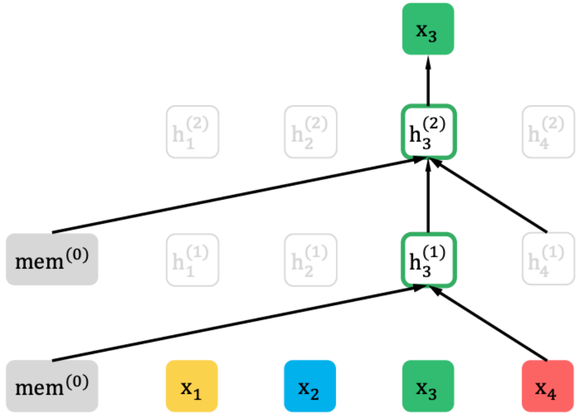
Factorization order: 3 → 2 → 4 → 1



Factorization order: 2 → 4 → 3 → 1



Factorization order: 1 → 4 → 2 → 3



Factorization order: 4 → 3 → 1 → 2

Permutation

[3, 2, 4, 1]

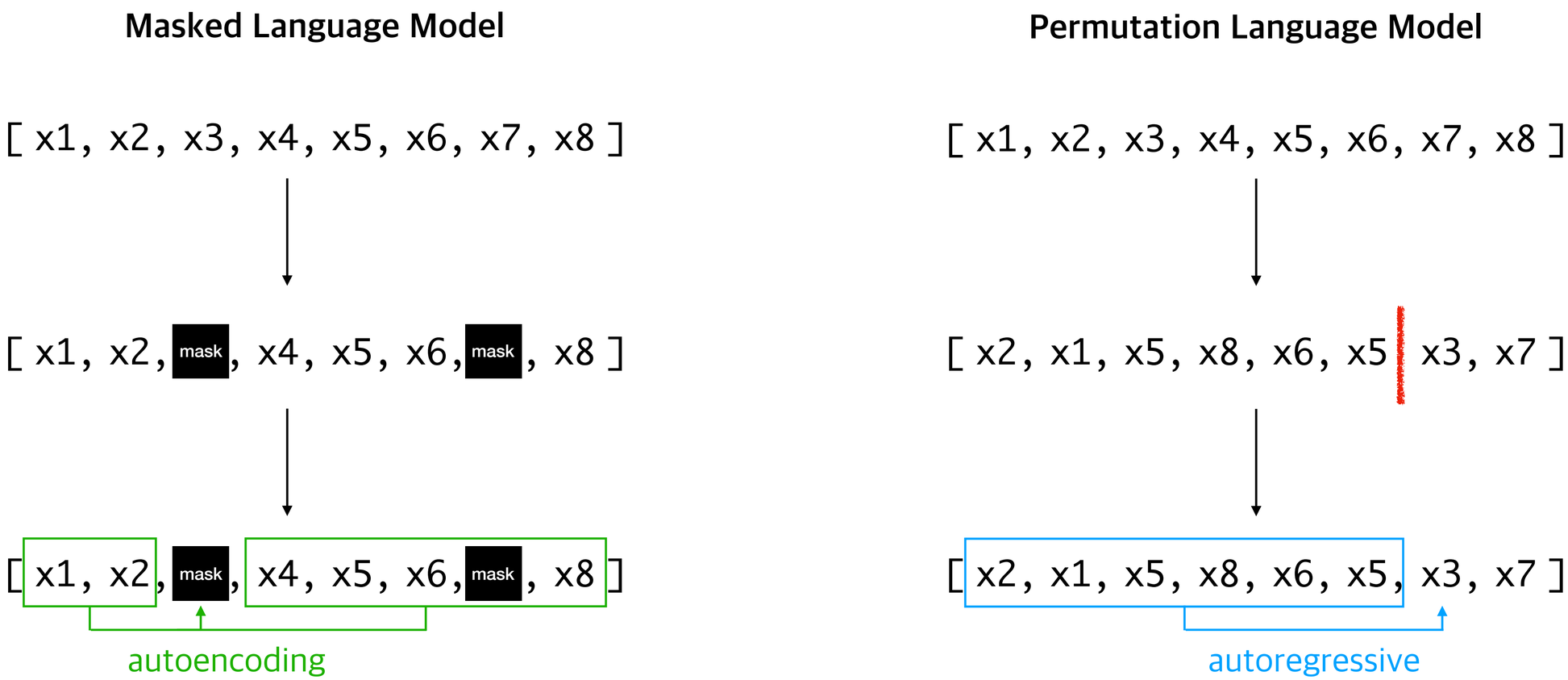
[2, 4, 3, 1]

[1, 4, 2, 3]

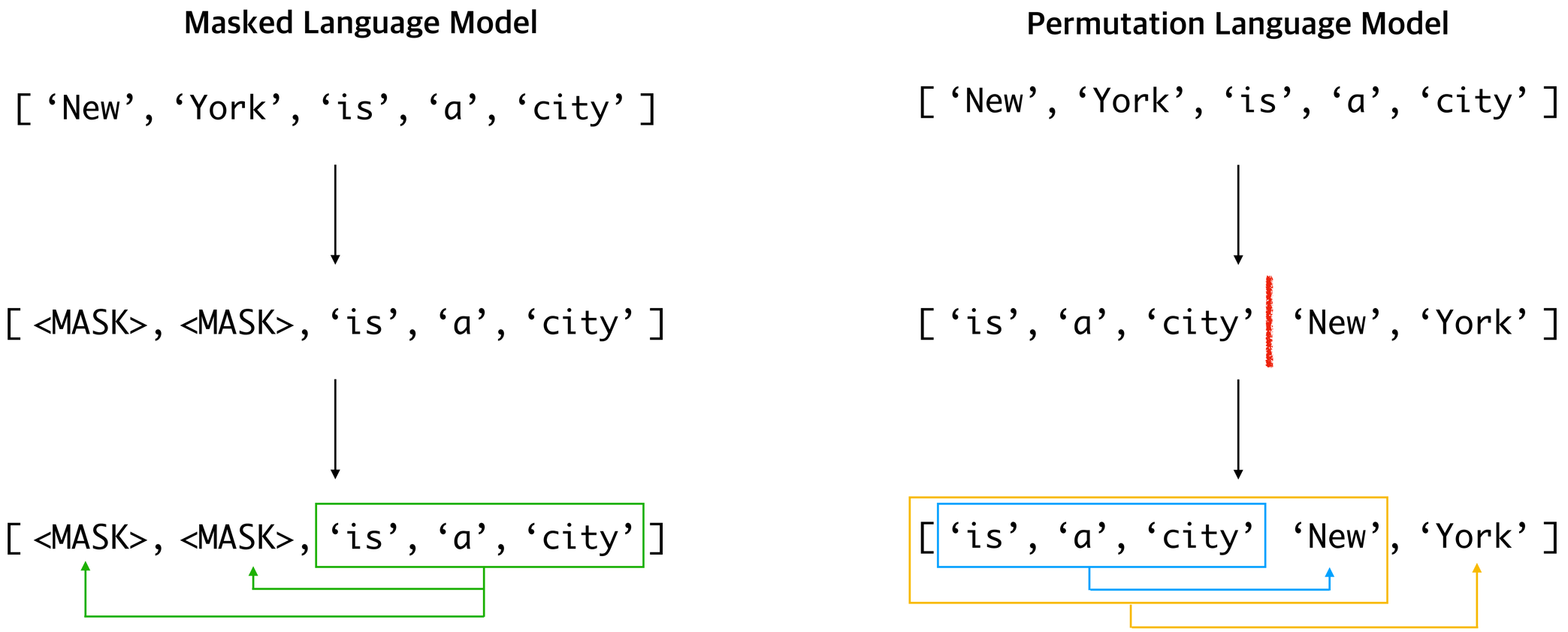
[4, 3, 1, 2]

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

Permutation Language Modeling



Permutation Language Modeling

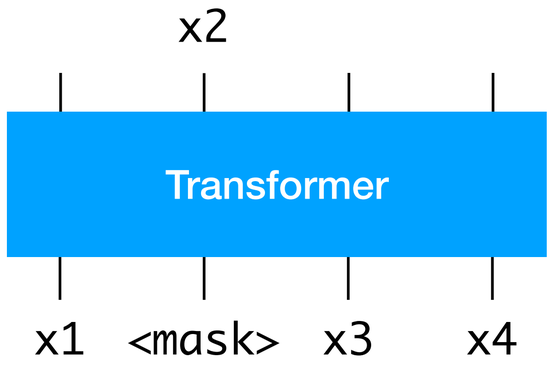


Two-Stream Self-Attention

GPT

[x1, x2, x3, x4] → [x5]

BERT



Permutation LM

[x2, x1, x4, x3] → [x5]

→ [x6] ???

→ [x7]

Two-Stream Self Attention

not used when fine-tuning

Query Stream $g_{z_t}^{(m)} \leftarrow \text{Attention}(\text{Q} = g_{z_t}^{(m-1)}, \text{KV} = \mathbf{h}_{\mathbf{z}_{<t}}^{(m-1)}; \theta),$

Content Stream $h_{z_t}^{(m)} \leftarrow \text{Attention}(\text{Q} = h_{z_t}^{(m-1)}, \text{KV} = \mathbf{h}_{\mathbf{z}_{\leq t}}^{(m-1)}; \theta),$

Two-Stream Self-Attention

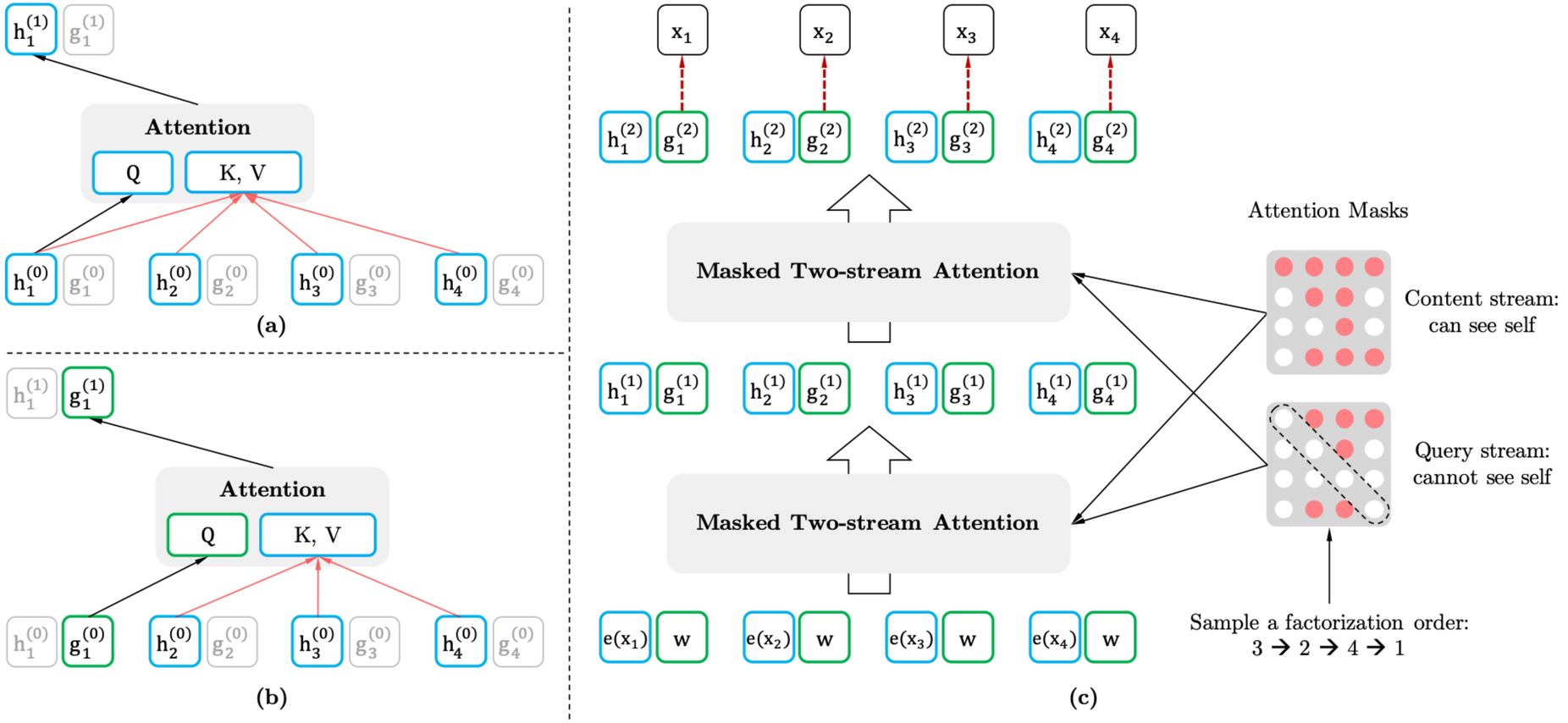
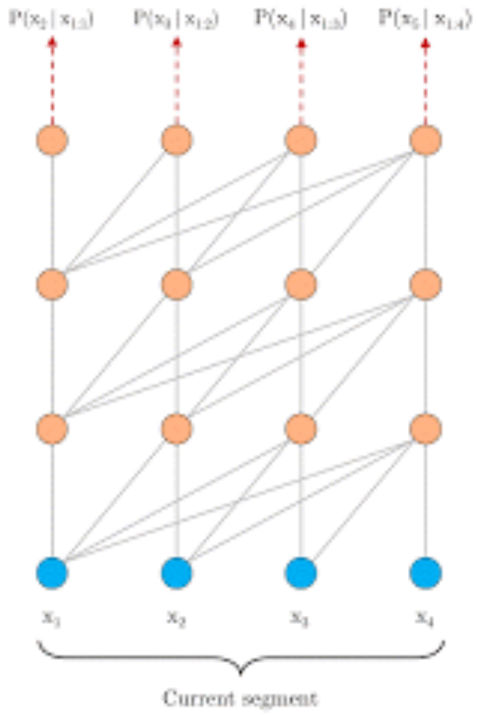


Figure 2: (a): Content stream attention, which is the same as the standard self-attention. (b): Query stream attention, which does not have access information about the content x_{z_t}. (c): Overview of the permutation language modeling training with two-stream attention.

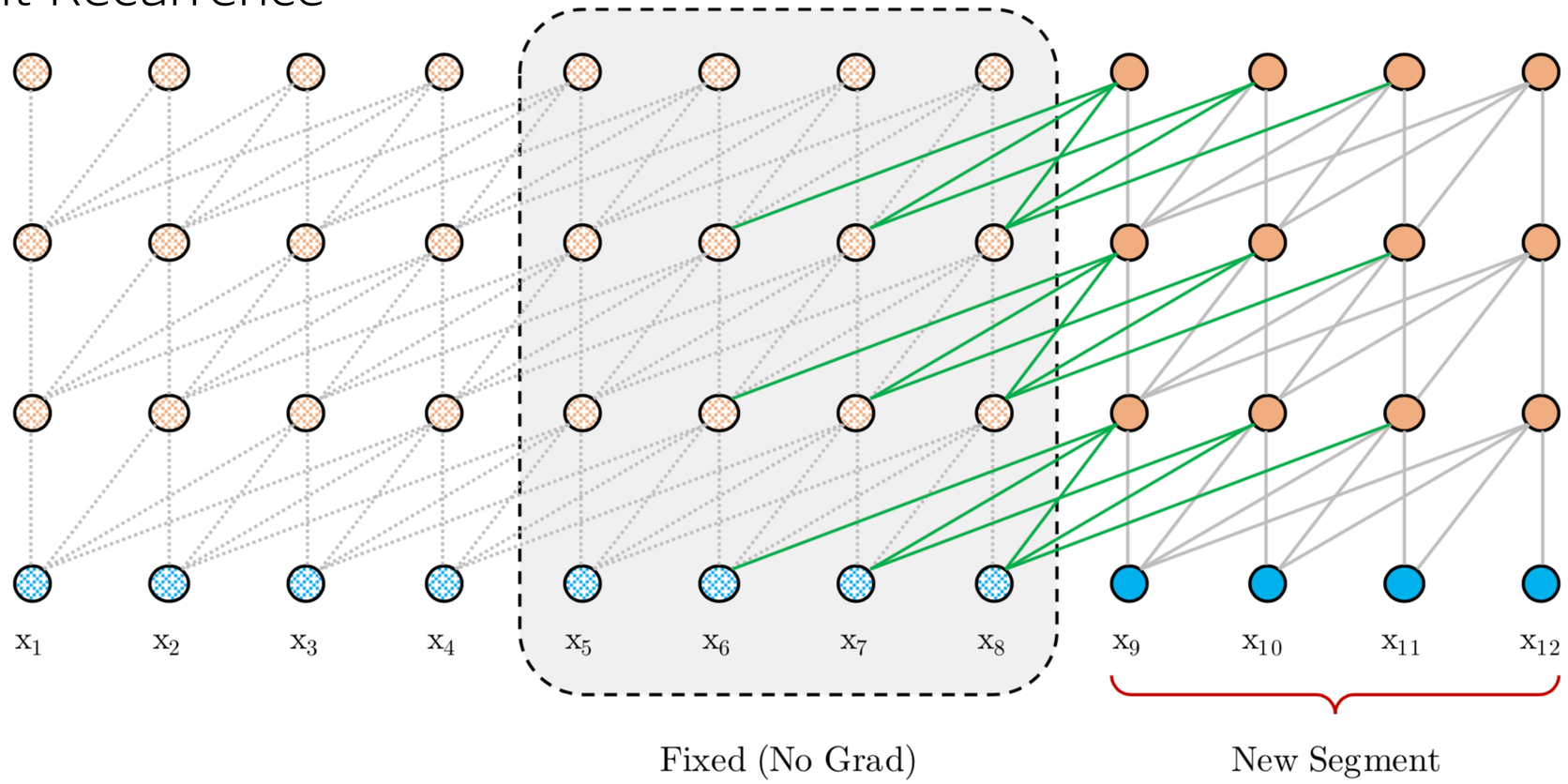
Transformer-XL (PR-161)

- Segment Recurrence



Transformer-XL (PR-161)

- Segment Recurrence



Google AI blog, Transformer-XL: Unleashing the Potential of Attention Models

Transformer-XL (PR-161)

- Relative Positional Embedding
  - When reuse old hidden states, how can we define positional encodings?

`[0, 1, 2, 3]` **segment 1** `[0, 1, 2, 3]` **segment 2** `[0, 1, 2, 3]` **segment 3**

  - It is enough to know **relative distance** between key vector and query vector ($i - j$)
    - Standard Transformer

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

    - Transformer-XL

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

Experiments

- Pretraining Datasets
  - BERT : BookCorpus + English Wikipedia
  - XLNet : BookCorpus + English Wikipedia + Giga5 + ClueWeb + Common Crawl

- Model Size
  - XLNet-Large $\approx$ BERT-Large

- Training Time
  - 512 TPU v3 chips for 500K steps with adam optimizer (batch size 2048, 2.5days)

    → **Still underfits!**

Experiments

- SQuAD Dataset

SQuAD1.1	EM	F1	SQuAD2.0	EM	F1
Dev set results without data augmentation					
BERT [10]	84.1	90.9	BERT† [10]	78.98	81.77
XLNet	**88.95**	**94.52**	XLNet	**86.12**	**88.79**
Test set results on leaderboard, with data augmentation (as of June 19, 2019)					
Human [27]	82.30	91.22	BERT+N-Gram+Self-Training [10]	85.15	87.72
ATB	86.94	92.64	SG-Net	85.23	87.93
BERT* [10]	87.43	93.16	BERT+DAE+AoA	85.88	88.62
XLNet	**89.90**	**95.08**	XLNet	**86.35**	**89.13**

Table 2: A single model XLNet outperforms human and the best ensemble by 7.6 EM and 2.5 EM on SQuAD1.1. $*$ means ensembles, † marks our runs with the official code.

Experiments

- Text Classification

Model	IMDB	Yelp-2	Yelp-5	DBpedia	AG	Amazon-2	Amazon-5
CNN [14]	-	2.90	32.39	0.84	6.57	3.79	36.24
DPCNN [14]	-	2.64	30.58	0.88	6.87	3.32	34.81
Mixed VAT [30, 20]	4.32	-	-	0.70	4.95	-	-
ULMFiT [13]	4.6	2.16	29.98	0.80	5.01	-	-
BERT [35]	4.51	1.89	29.32	0.64	-	2.63	34.17
XLNet	**3.79**	**1.55**	**27.80**	**0.62**	**4.49**	**2.40**	**32.26**

Table 3: Comparison with state-of-the-art error rates on the test sets of several text classification datasets. All BERT and XLNet results are obtained with a 24-layer architecture with similar model sizes (aka BERT-Large).

Experiments

- RACE Dataset

RACE	Accuracy	Middle	High
GPT [25]	59.0	62.9	57.4
BERT [22]	72.0	76.6	70.1
BERT+OCN* [28]	73.5	78.4	71.5
BERT+DCMN* [39]	74.1	79.5	71.8
XLNet	**81.75**	**85.45**	**80.21**

Table 1: Comparison with state-of-the-art results on the test set of RACE, a reading comprehension task. $*$ indicates using ensembles. "Middle" and "High" in RACE are two subsets representing middle and high school difficulty levels. All BERT and XLNet results are obtained with a 24-layer architecture with similar model sizes (aka BERT-Large). Our single model outperforms the best ensemble by 7.6 points in accuracy.

Experiments

- GLUE Dataset

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	WNLI
Single-task single models on dev									
BERT [2]	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-
XLNet	**89.8/-**	**93.9**	**91.8**	**83.8**	**95.6**	**89.2**	**63.6**	**91.8**	-
Single-task single models on test									
BERT [10]	86.7/85.9	91.1	89.3	70.1	94.9	89.3	60.5	87.6	65.1
Multi-task ensembles on test (from leaderboard as of June 19, 2019)									
Snorkel* [29]	87.6/87.2	93.9	89.9	80.9	96.2	91.5	63.8	90.1	65.1
ALICE*	88.2/87.9	95.7	**90.7**	83.5	95.2	92.6	**68.6**	91.1	80.8
MT-DNN* [18]	87.9/87.4	96.0	89.9	**86.3**	96.5	92.7	68.4	91.1	89.0
XLNet*	**90.2/89.7**†	**98.6**†	90.3†	**86.3**	**96.8**†	**93.0**	67.8	**91.6**	**90.4**

Table 4: Results on GLUE. ∗ indicates using ensembles, and † denotes single-task results in a multi-task row. All results are based on a 24-layer architecture with similar model sizes (aka BERT-Large). See the upper-most rows for direct comparison with BERT and the lower-most rows for comparison with state-of-the-art results on the public leaderboard.

Ablation Study

- Importance of Transformer-XL

#	Model	RACE	SQuAD2.0 F1	SQuAD2.0 EM	MNLI m/mm	SST-2
1	BERT-Base	64.3	76.30	73.66	84.34/84.65	92.78
2	DAE + Transformer-XL	65.03	79.56	76.80	84.88/84.45	92.60
3	XLNet-Base ($K = 7$)	66.05	**81.33**	**78.46**	**85.84/85.43**	92.66
4	XLNet-Base ($K = 6$)	66.66	80.98	78.18	85.63/85.12	**93.35**
5	- memory	65.55	80.15	77.27	85.32/85.05	92.78
6	- span-based pred	65.95	80.61	77.91	85.49/85.02	93.12
7	- bidirectional data	66.34	80.65	77.87	85.31/84.99	92.66
8	+ next-sent pred	**66.76**	79.83	76.94	85.32/85.09	92.89

Table 6: Ablation study. The results of BERT on RACE are taken from [39]. We run BERT on the other datasets using the official implementation and the same hyperparameter search space as XLNet. K is a hyperparameter to control the optimization difficulty (see Section 2.3). All models are pretrained on the same data.

Ablation Study

- Effectiveness of Permutation Language Modeling

#	Model	RACE	SQuAD2.0 F1	SQuAD2.0 EM	MNLI m/mm	SST-2
1	BERT-Base	64.3	76.30	73.66	84.34/84.65	92.78
2	DAE + Transformer-XL	65.03	79.56	76.80	84.88/84.45	92.60
3	XLNet-Base ($K = 7$)	66.05	**81.33**	**78.46**	**85.84/85.43**	92.66
4	XLNet-Base ($K = 6$)	66.66	80.98	78.18	85.63/85.12	**93.35**
5	- memory	65.55	80.15	77.27	85.32/85.05	92.78
6	- span-based pred	65.95	80.61	77.91	85.49/85.02	93.12
7	- bidirectional data	66.34	80.65	77.87	85.31/84.99	92.66
8	+ next-sent pred	**66.76**	79.83	76.94	85.32/85.09	92.89

Table 6: Ablation study. The results of BERT on RACE are taken from [39]. We run BERT on the other datasets using the official implementation and the same hyperparameter search space as XLNet. K is a hyperparameter to control the optimization difficulty (see Section 2.3). All models are pretrained on the same data.

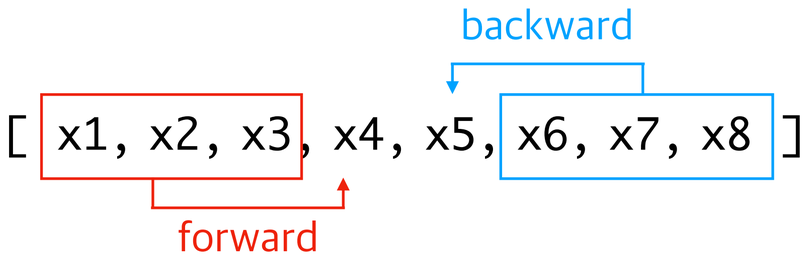
Conclusion

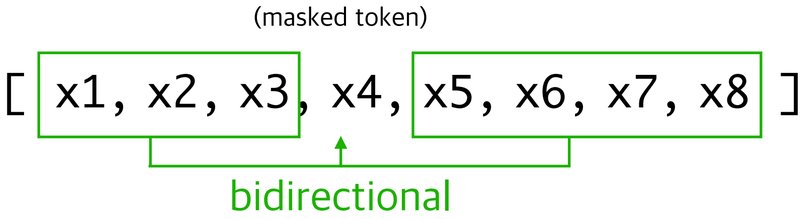
- Permutation language model can cover both AR, AE models

[1, 2, 3, 4]

[8, 7, 6, 5]

[1, 4, 2, 3]





- Transformer-XL is very good for downstream tasks as well as language modeling

[−] **reject**

ICLR 2019 Conference Paper717 Area Chair1

14 Dec 2018 (modified: 21 Dec 2018) ICLR 2019 Conference Paper717 Meta Review Readers: Everyone

Metareview: despite the (significant) improvement in language modelling, it has always been a thorny issue whether better language models (at this level) lead to better performance in the downstream task or whether such a technique could be used to build a better conditional language model which often focuses on the aspect of generation. in this context, the reviewers found it difficult to see the merit of the proposed approach, as the technique itself may be considered a rather trivial application of earlier approaches such as truncated backprop. it would be good to apply this technique to e.g. document-level generation and see if the proposed approach can strike an amazing balance between computational efficiency and generation performance.

Confidence: 4: The area chair is confident but not absolutely certain

Recommendation: Reject

Summary

Generalized Autoregressive Pretraining for Language Understanding

Summary

Generalized Autoregressive Pretraining for Language Understanding

Pretrain without data corruption(masking) by using **Permutation LM**

Summary

Generalized **Autoregressive** Pretraining for Language Understanding

Autoregressive language model but **also utilizes bidirectional context**

Summary

Generalized Autoregressive Pretraining for **Language Understanding**

Outperformed BERT on 20 NLP tasks, 18 of them are State Of The Art