XLNet:

Generalized Autoregressive Pretraining for Language Understanding

2021.06.10

백서인

Contents

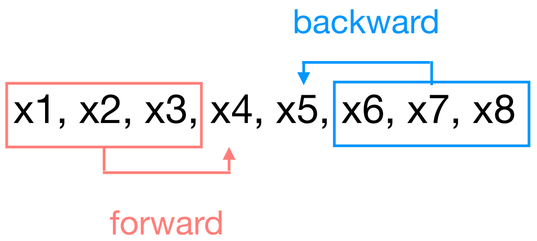
Abstract

- Denoising autoencoding based pertaining like BERT achieves better performance than pertaining approaches based on autoregressive language modeling
- However, BERT neglects dependency between the masked positions and suffers from a pretrain-finetune discrepancy
- Propose XLNet, a generalized autoregressive pretraining method that
  - (1) enables learning bidirectional contexts
  - (2) overcomes the limitations of BERT thanks to its autoregressive formulation
- Integrate ideas from Transformer-XL
- XLNet outperforms BERT on 20 tasks

Introduction

- Autoregressive(AR, GPT)

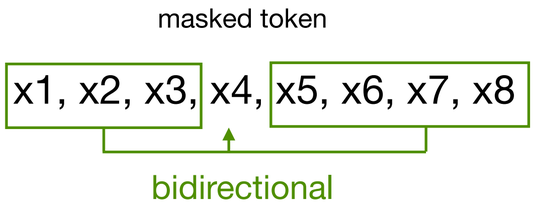
$$\max_{\theta} \log p_\theta(x) = \max_{\theta} \Sigma_{t=1}^{T} \log p(x_t \mid x_{<t})$$



- AR은 방향성(forward, backward)이 정해져야 하므로, 한쪽 방향의 정보만 이용 가능
- 양방향 문맥을 활용하지 못하고 문장을 깊이 이해하기 어려움
- ELMO의 경우 양방향을 이용하지만, 각 방향에 대해 독립적으로 학습된 모델을 사용하여 얕은 이해만 가능

- Autoencoding(AE, BERT)

$$\max_{\theta} \Sigma_{t=1}^{T} m_t \log p(x_t \mid \widehat{x})$$



- mask된 토큰을 맞추기 위해 양방향의 정보를 이용
- 하지만 independence assumption으로 모든 masked token이 독립적으로 예측됨으로써 이들 간의 dependency를 학습할 수 없음
- 또한 nose(masked token)은 실제 fine-tuning 과정에서는 등장하지 않아, pre-training과 fine-tuning 사이의 불일치 발생

Proposed Method: XLNet

1. Permutation Language Modeling
2. Two-Stream Self-Attention
3. Transformer-XL

Permutation Language Modeling

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 5 1 6 8 7 2 4

…

8! 개

Permutation Language Modeling

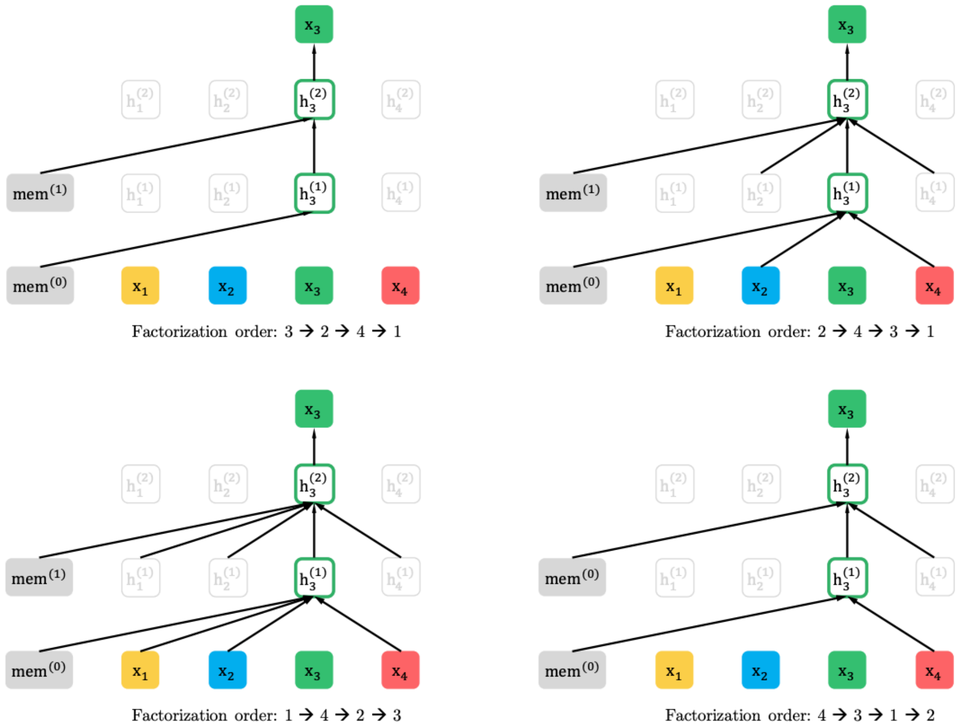


Figure 4: Illustration of the permutation language modeling objective for predicting x_3 given the same input sequence $\mathbf{x}$ but with different factorization orders.

- x_3을 예측하기 위해 $[x_1, x_2, x_4]$의 모든 부분집합 사용
- 특정 token에 대한 양방향 context를 고려한 AR 모델링 가능, 기존 AR 방식의 한계 극복
- AR 방식이므로 independent assumption이 필요 없고, masked token을 이용하지 않으므로 pre-training과 fine-tuning 간 불일치 없음
- Positional encodings을 사용하여 original sequence를 유지해줌

Two-Stream Self-Attention

GPT

x1, x2, x3, x4 → x5

BERT

x3

Model

x1 x2 mask x4 x5

Permutation LM

x1, x2, x3, x4 → x5

→ x6

→ x7

Two-Stream Self-Attention

GPT

x1, x2, x3, x4 → x5

BERT

x3

Model

x1 x2 mask x4 x5

Permutation LM

x1, x2, x3, x4 → x5

→ x6

→ x7

→ 이를 해결하기 위해 이전의 context token들의 정보뿐만 아니라 target index의 position 정보도 함께 이용하는 Target Position-Aware Representation 제안

Two-Stream Self-Attention

- Target position 정보를 추가적으로 이용하기 위한 g_θ 의 조건

1. 특정 시점 t 에서 target position z_t의 토큰 x_{z_t}을 예측하기 위해, hidden representation g($x_{z<t}$, z_t)는 t 시점 이전의 context 정보 $x_{z<t}$ 와 target position 정보 z_t만을 이용
2. 특정 시험 t 이후인 $j(>t)$ 에 해당하는 x_{z_j}를 예측하기 위해, hidden representation g($x_{z<t}$, z_t)가 t 시점의 content인 x_{z_t}를 인코딩

- 일반적인 Transformer 모델에서는 각 layer에서 한 토큰당 하나의 representation을 갖는 구조, 이 구조는 위의 두 조건을 만족시킬 수 없음
- 1번 조건에 따르면 t 시점의 hidden representation은 t 시점의 context를 포함할 수 없음
- 2번 조건에 따르면 t 시점 이후의 hidden representation을 계산하기 위해서는 t 시점의 hidden representatation을 이용해야하는데, 이는 t 시점의 context를 포함해야 함

→ 이를 해결하기 위해 2개의 hidden representation을 이용하는 Transformer 구조 제안

Two-Stream Self-Attention

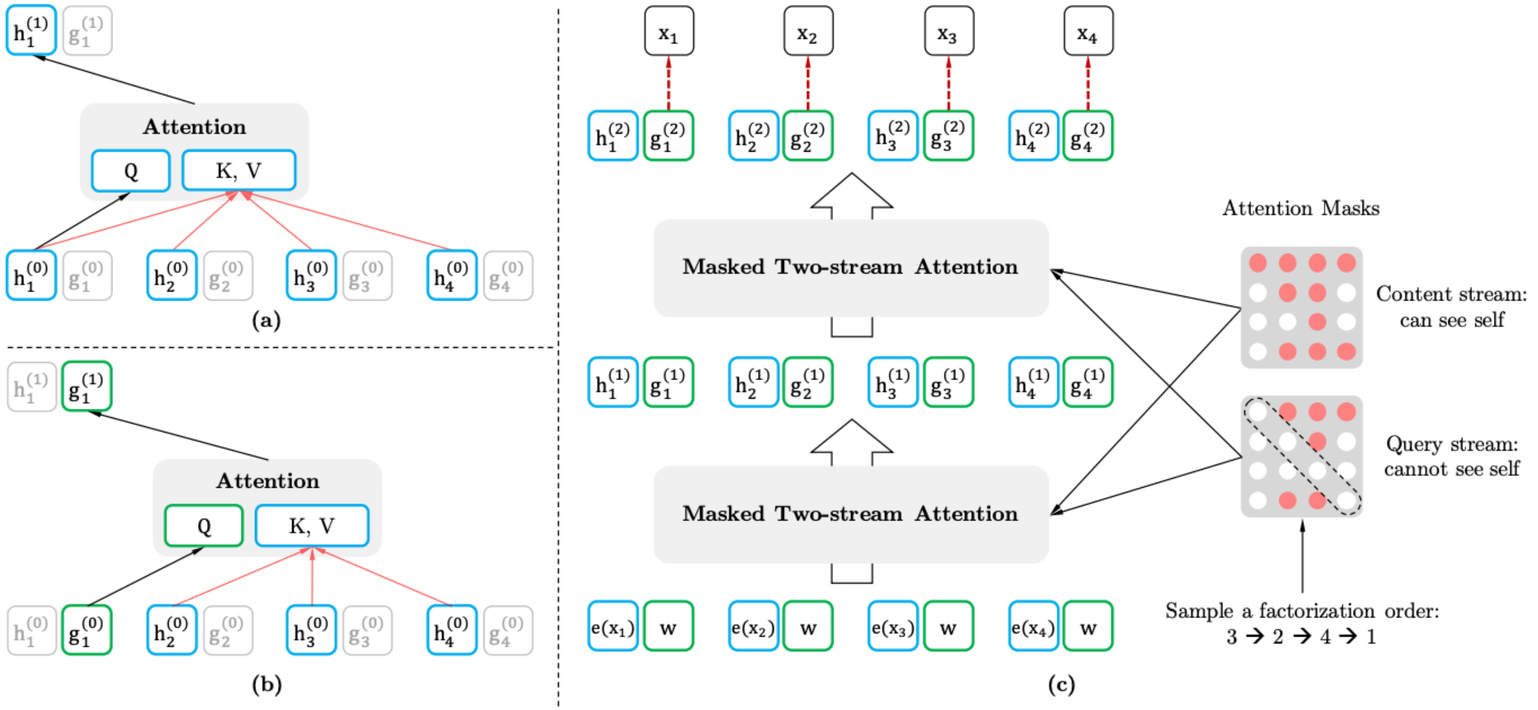


Figure 1: (a): Content stream attention, which is the same as the standard self-attention. (b): Query stream attention, which does not have access information about the content x_{z_t}. (c): Overview of the permutation language modeling training with two-stream attention.

Query Representation

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

- Query: 이전 layer의 g state ($z = t$)
- Key, Value: 이전 layer의 h state ($z < t$)

→ 1번 조건 만족

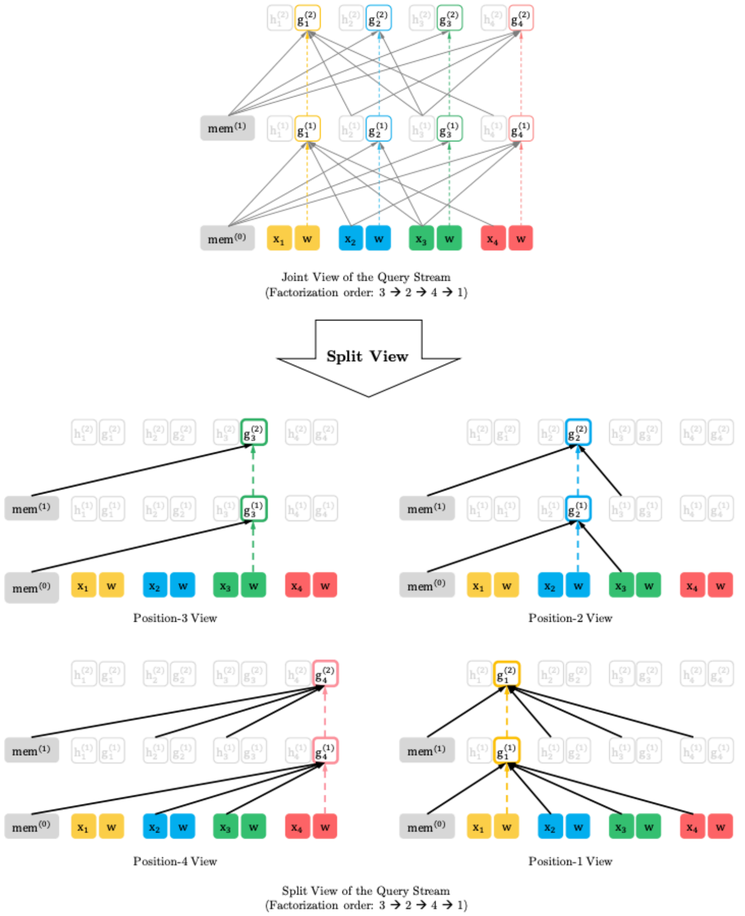


Figure 6: A detailed illustration of the **query stream** of the proposed objective with both the joint view and split views based on a length-4 sequence under the factorization order [3, 2, 4, 1]. The dash arrows indicate that the query stream cannot access the token (content) at the same position, but only the location information.

Context Representation

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

- Query: 이전 layer의 h state ($z = t$)
- Key, Value: 이전 layer의 h state ($z \leq t$)

→ 2번 조건 만족

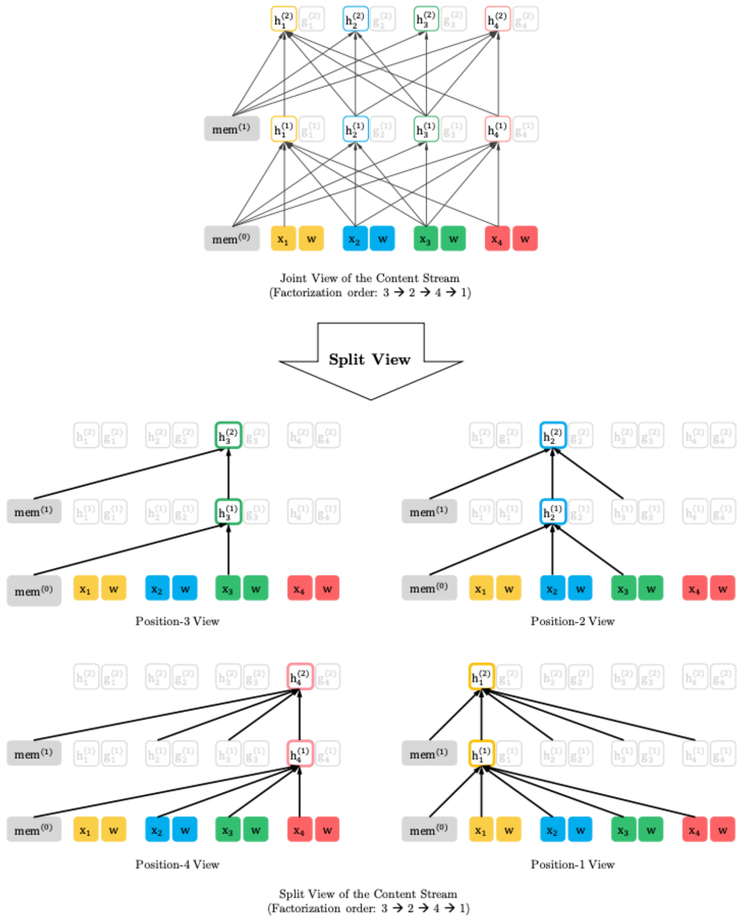


Figure 5: A detailed illustration of the **content stream** of the proposed objective with both the joint view and split views based on a length-4 sequence under the factorization order [3, 2, 4, 1]. Note that if we ignore the query representation, the computation in this figure is simply the standard self-attention, though with a particular attention mask.

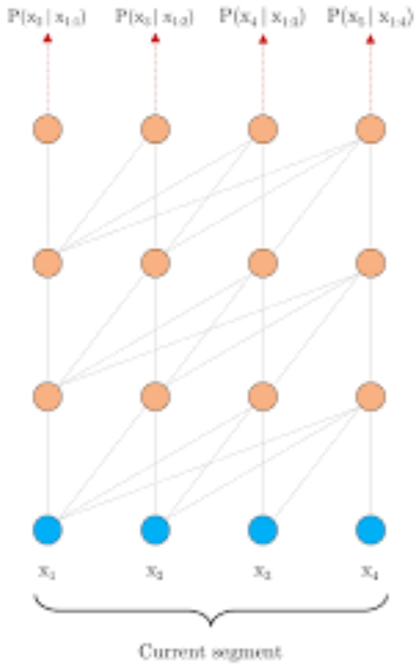
Partial Prediction

- Permutation LM에서는 모든 조합을 계산 → 느린 수렴
- 이를 해결하기 위해 마지막 몇 개의 예측만 이용
- 예를 들어 3 → 2 → 4 → 1 순서에서 마지막 2개만 예측에 이용

$$p(x_3)p(x_2 \mid x_3)p(x_4 \mid x_2, x_3)p(x_1 \mid x_3, x_2, x_4) \rightarrow p(x_4 \mid x_2, x_3)p(x_1 \mid x_3, x_2, x_4)$$

Transformer-XL

- Segment Recurrence



Transformer-XL

- Relative Segment Encodings

[0, 1, 2, 3]	[0, 1, 2, 3]	[0, 1, 2, 3]
Segment 1	Segment 2	Segment 3

- Attention score in standard Transformer

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

- Attention score with Relative Positional Encoding

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

Discussion

[New, York, is, a, city]

BERT

[<mask>, <mask>, is, a, city]

[is, a, city] → [New]

[is, a, city] → [York]

XLNet

[is, a, city | New, York]

[is, a, city] → [New]

[New, is, a, city] → [York]

Experiments

- Dataset: BooksCorpus + English Wikipedia + Giga5 + ClueWeb 2012-B + Common Crawl
- Model Size: BERT-Large ≈ XLNet-Large
- TPU v3, 2.5days, 500K step

Experiments

- Race Dataset

RACE	Accuracy	Middle	High	Model	NDCG@20	ERR@20
GPT [28]	59.0	62.9	57.4	DRMM [13]	24.3	13.8
BERT [25]	72.0	76.6	70.1	KNRM [8]	26.9	14.9
BERT+DCMN* [38]	74.1	79.5	71.8	Conv [8]	28.7	18.1
RoBERTa [21]	83.2	86.5	81.8	BERT†	30.53	18.67
XLNet	**85.4**	**88.6**	**84.0**	XLNet	**31.10**	**20.28**

Table 2: Comparison with state-of-the-art results on the test set of RACE, a reading comprehension task, and on ClueWeb09-B, a document ranking task. ∗ indicates using ensembles. † indicates our implementations. "Middle" and "High" in RACE are two subsets representing middle and high school difficulty levels. All BERT, RoBERTa, and XLNet results are obtained with a 24-layer architecture with similar model sizes (aka BERT-Large).

Experiments

- SQuAD Dataset

SQuAD2.0	EM	F1	SQuAD1.1	EM	F1
Dev set results (single model)					
BERT [10]	78.98	81.77	BERT† [10]	84.1	90.9
RoBERTa [21]	86.5	89.4	RoBERTa [21]	88.9	94.6
XLNet	**87.9**	**90.6**	XLNet	**89.7**	**95.1**
Test set results on leaderboard (single model, as of Dec 14, 2019)					
BERT [10]	80.005	83.061	BERT [10]	85.083	91.835
RoBERTa [21]	86.820	89.795	BERT* [10]	87.433	93.294
XLNet	**87.926**	**90.689**	XLNet	**89.898**‡	**95.080**‡

Table 3: Results on SQuAD, a reading comprehension dataset. † marks our runs with the official code. ∗ indicates ensembles. ‡: We are not able to obtain the test results of our latest model on SQuAD1.1 from the organizers after submitting our result for more than one month, and thus report the results of an older version for the SQuAD1.1 test set.

Experiments

- Text Classification

Model	IMDB	Yelp-2	Yelp-5	DBpedia	AG	Amazon-2	Amazon-5
CNN [15]	-	2.90	32.39	0.84	6.57	3.79	36.24
DPCNN [15]	-	2.64	30.58	0.88	6.87	3.32	34.81
Mixed VAT [31, 23]	4.32	-	-	0.70	4.95	-	-
ULMFiT [14]	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	**3.20**	**1.37**	**27.05**	**0.60**	**4.45**	**2.11**	**31.67**

Table 4: 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

- 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	-
RoBERTa [21]	90.2/90.2	94.7	92.2	**86.6**	96.4	**90.9**	68.0	92.4	-
XLNet	**90.8/90.8**	**94.9**	**92.3**	85.9	**97.0**	90.8	**69.0**	**92.5**	-
Multi-task ensembles on test (from leaderboard as of Oct 28, 2019)									
MT-DNN* [20]	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0
RoBERTa* [21]	90.8/90.2	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0
XLNet*	**90.9/90.9**†	**99.0**†	**90.4**†	**88.5**	**97.1**†	**92.9**	**70.2**	**93.0**	**92.5**

Table 5: Results on GLUE. $*$ indicates using ensembles, and † denotes single-task results in a multi-task row. All dev results are the median of 10 runs. The upper section shows direct comparison on dev data and the lower section shows comparison with state-of-the-art results on the public leaderboard.

Ablation Study

#	Model	RACE	SQuAD2.0		MNLI	SST-2
			F1	EM	m/mm	
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: The results of BERT on RACE are taken from [38]. 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).

Conclusion

- Generalized AR pertaining method that uses a permutation language modeling objective to combine the advantage of AR and AE methods
- Integrate Transformer-XL and the careful design of the two-stream attention mechanism
- XLNet achieves substantial improvement over precious pertaining objectives on various task

References

- 유튜브 채널 - 박성남, PR-175: XLNet: Generalized Autoregressive Pretraining for Language Understanding
- PINGPONG 블로그 - 꼼꼼하고 이해하기 쉬운 XLNet 논문 리뷰
- Google AI Blog, Transformer-XL: Unleashing the Potential of Attention Models