XLNet

Generalized Autoregressive Pretraining for Language Understanding

Authors

Zhilin Yang & Zihang Dai, etc.

Carnegie Mellon University, Google Brain

Transformer XL first authors

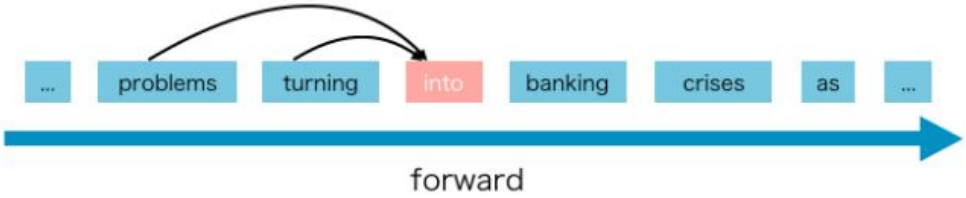
Unsupervised representation learning

1) pretrain neural networks on large-scale unlabeled text corpora

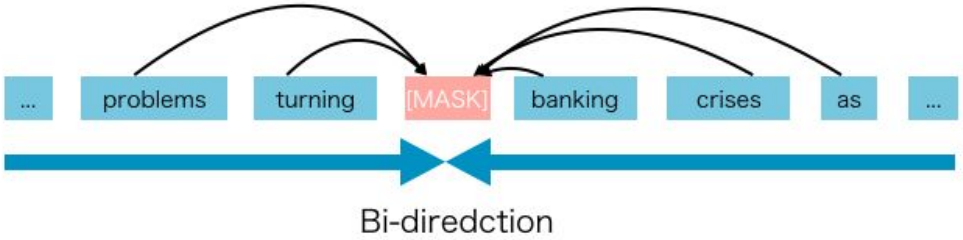
2) finetune the models or representations on downstream tasks

Unsupervised pretraining objectives

1) autoregressive (AR) language modeling (ex. GPT, ELMo)



2) autoencoding (AE) (ex. BERT)



AR

- **Objective**

$$input\ sequence : x = (x_1, x_2, \ldots, x_T)$$

$$forward\ likelihood : p(x) = \Pi_{t=1}^{T} p(x_t \mid x_{<t})$$

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

- **Problems**

  Lack of information (bidirectionality not considered)

AE

- **Objective**

$$input\ sequence : \overline{x} = (x_1, x_2, \ldots, x_T)$$

$$corrupted\ input : \widehat{x} = (x_1, [MASK], \ldots x_T)$$

$$likelihood : p(\overline{x} \mid \widehat{x}) \approx \Pi_{t=1}^{T} p(x_t \mid \widehat{x})$$

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

- **Problems**

  Independent assumption: dependency b/t masked tokens not considered

  J = log p(New | is, a, city) + log p(York | is, a, city), given [New_{masked}, $York_{masked}$, is, a, city]

  Discrepancy from downstream tasks

  No masking when fine-tuning

Transformer

- **Problems**
  Limited context

XLNet

- **What's new**

  Permutation (no masking)

    consistent with downstream task

    dependency b/t masked tokens not lost

    bidirectionality covered (as a subset of permutations)

  Transformer XL

    context extended

Permutation

- **Objective**

$$input\ sequence : x = (x_1, x_2, \ldots, x_T)$$

$$likelihood : \mathbb{E}_{z \sim Z_T}[\Pi_{t=1}^{T} p(x_{z_t} \mid x_{z<t})]$$

$$training\ objective : \max_{\theta}\ \mathbb{E}_{z \sim Z_T}[\Sigma_{t=1}^{T} \log\ p_\theta(x_{z_t} \mid x_{z<t})]$$

- **Example**

  z_1: [1, 2, 4, 3], z_2: [1, 3, 2, 4], …, z_{24}: [4, 3, 2, 1] , Z_T: [z_1, z_2, …, z_{24}]

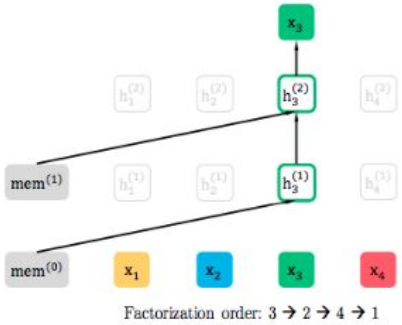
  $p(x) = p(x_3|mem)p(x_2|mem, x_3)p(x_4|mem, x_3, x_2)p(x_1|mem, x_3, x_2, x_4)$

- **Pros**

  consistent with downstream task (no masking)
  dependency b/t masked tokens not lost (no independence assumption)
  bidirectionality covered (as a subset of permutation set)

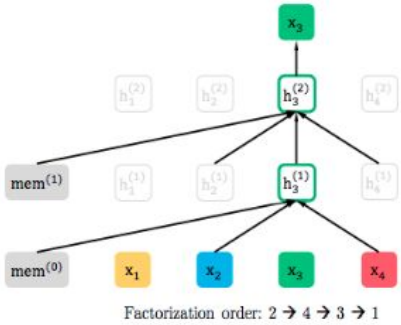
Permutation with Attention

[**a**, am, student, I]



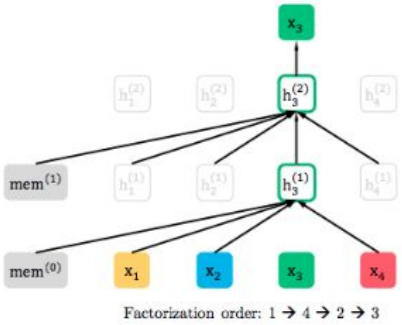
Factorization order: 3 → 2 → 4 → 1

[am, student, **a**, I]



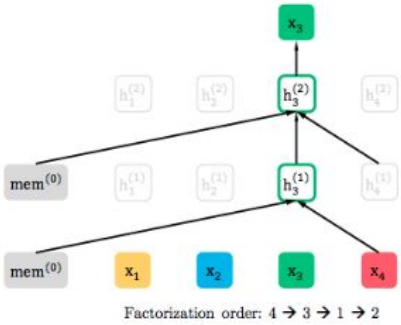
Factorization order: 2 → 4 → 3 → 1

[I, student, am, **a**]



Factorization order: 1 → 4 → 2 → 3

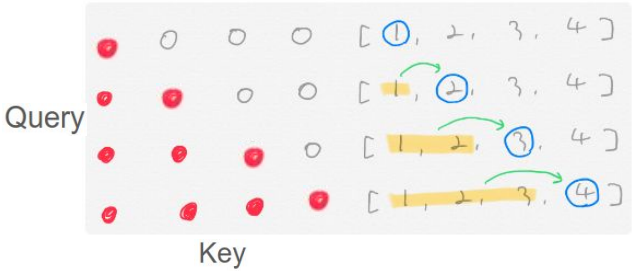
[student, **a**, I, am]



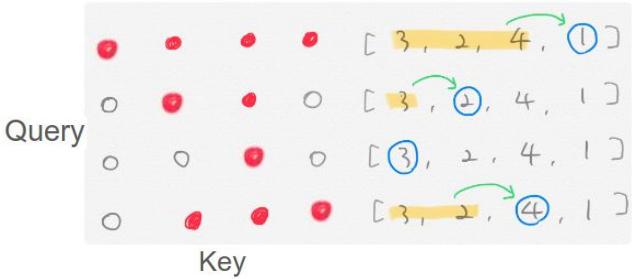
Factorization order: 4 → 3 → 1 → 2

Permutation with Attention

- [1, 2, 3, 4]



- [3, 2, 4, 1]



Problems with Permutation

- Given [I, am, a, student]$_{permutation1}$
  [I, am, student, a]$_{permutation2}$

  p(a | I, am) = p(student | I, am)?

- Slow convergence & expensive

Target Position Aware Representation

$$h_\theta(x_{z<t}) \rightarrow g_\theta(x_{z<t}, z_t)$$

Target position is not obvious anymore

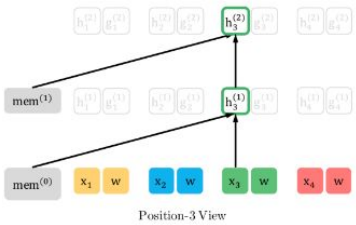
content info (x) **AND** target position info (z)

TWO representations per time step t

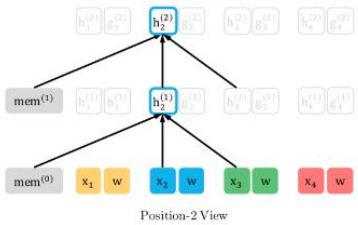
**x_i still has its positional encoding

Two-stream Self-attention

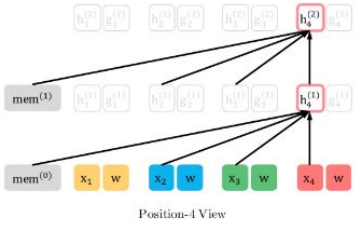
- Content: [3, 2, 4, 1]



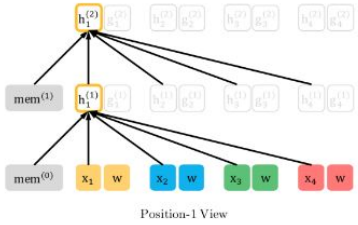
Position-3 View



Position-2 View

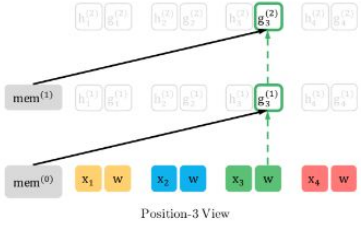


Position-4 View

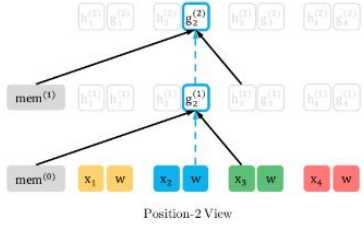


Position-1 View

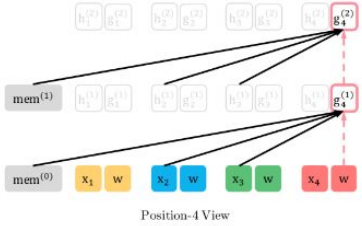
- Query: [3, 2, 4, 1]



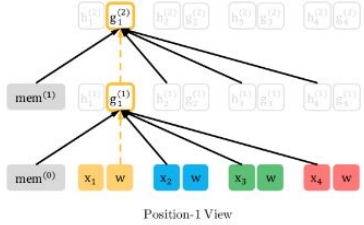
Position-3 View



Position-2 View



Position-4 View

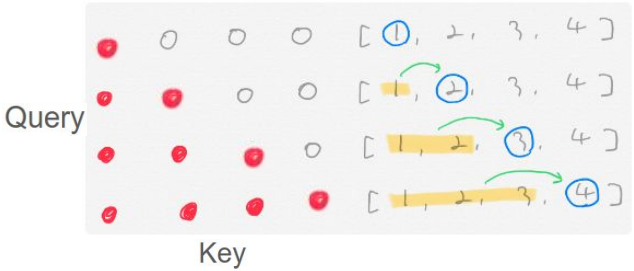


Position-1 View

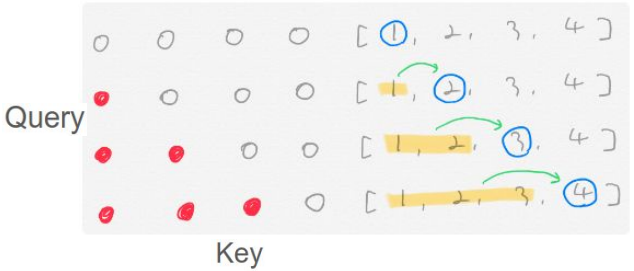
Two-stream Self-attention

- [1, 2, 3, 4]

Content stream

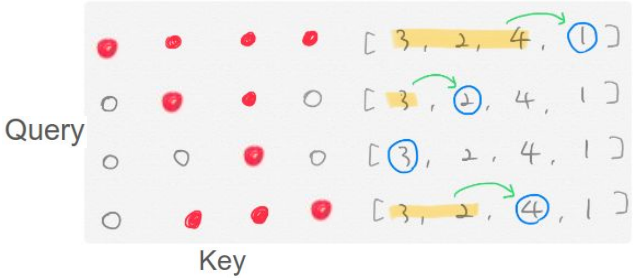


Query stream

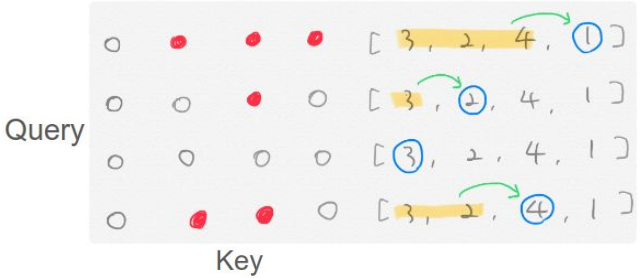


- [3, 2, 4, 1]

Content stream



Query stream



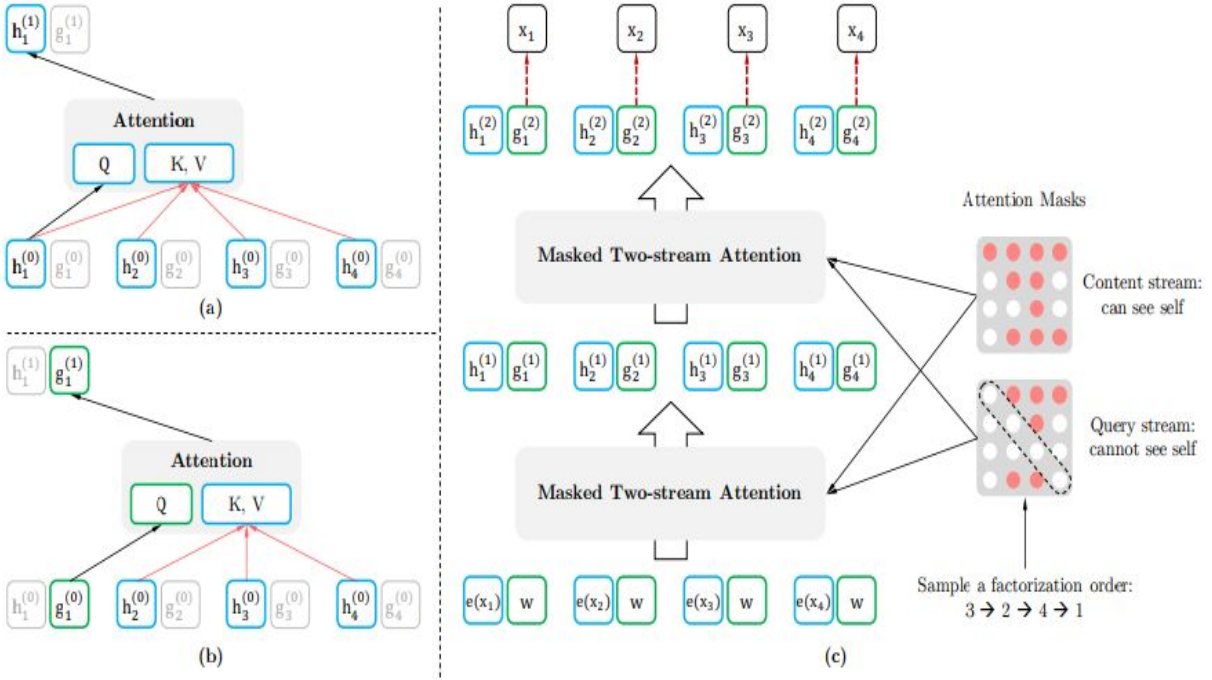
Two-stream Self-attention

Content (self-included)

$$\mathbf{h}_{z_t}^{(m)} \leftarrow \text{Attention}\left(\mathbf{Q} = \mathbf{h}_{z_t}^{(m-1)}, \mathbf{KV} = \mathbf{h}_{z \le t}^{(m-1)}; \theta\right)$$

Query (self-excluded)

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



Partial Prediction

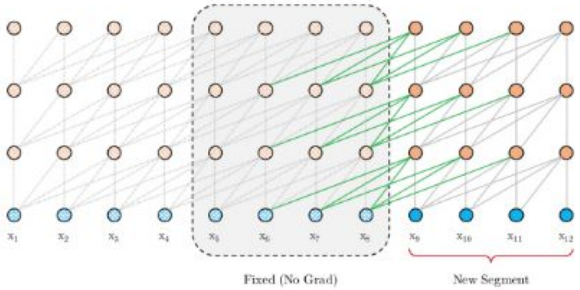
Predict the last few

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

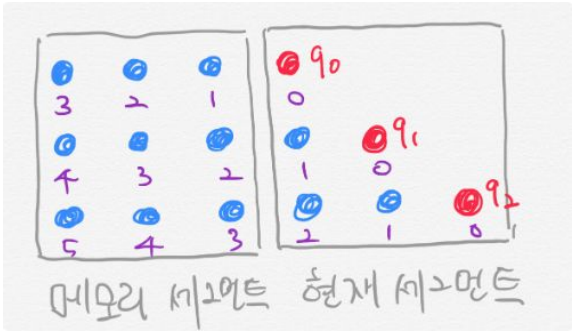
$$\max_{\theta} \quad \mathbb{E}_{\mathbf{z} \sim \mathcal{Z}_T} \left[\log p_\theta(\mathbf{x}_{\mathbf{z}_{>c}} \mid \mathbf{x}_{\mathbf{z}_{\leq c}}) \right] = \mathbb{E}_{\mathbf{z} \sim \mathcal{Z}_T} \left[\sum_{t=c+1}^{|\mathbf{z}|} \log p_\theta(x_{z_t} \mid \mathbf{x}_{\mathbf{z}_{<t}}) \right]$$

Transformer XL: Extending Context

- Segment Recurrence



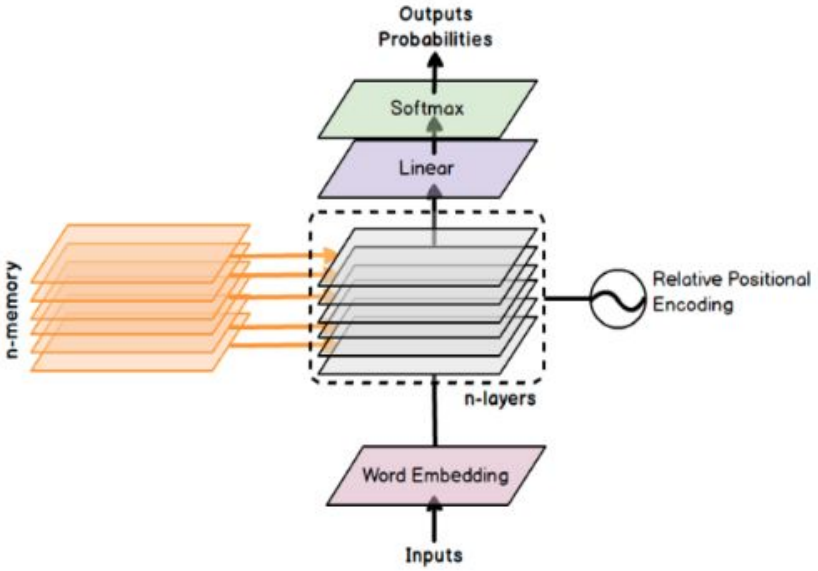
- Relative Positional Embedding (fixed)



Segment Recurrence

The n-th layer Key, Value are backed by the previous segment's n-1th hidden state

$$\widetilde{\mathbf{h}}_{\tau+1}^{n-1} = \left[\mathrm{SG}(\mathbf{h}_{\tau}^{n-1}) \circ \mathbf{h}_{\tau+1}^{n-1}\right],$$
$$\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n} = \mathbf{h}_{\tau+1}^{n-1}\mathbf{W}_{q}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{k}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{v}^{\top},$$
$$\mathbf{h}_{\tau+1}^{n} = \text{Transformer-Layer}\left(\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n}\right)$$



Two-stream Attention with Segment Recurrence

Initial represetation:

$$\forall t = 1, \ldots, T: \quad h_t = e(x_t) \quad \text{and} \quad g_t = w$$

Cached layer-m content represetation (memory) from previous segment: $\tilde{\mathbf{h}}^{(m)}$

For the Transformer-XL layer $m = 1, \cdots, M$, attention with relative positional encoding and position-wise feed-forward are consecutively employed to update the represetntations:

$$\begin{aligned}
\forall t = 1, \ldots, T: \quad \hat{h}_{z_t}^{(m)} &= \text{LayerNorm}\left(h_{z_t}^{(m-1)} + \text{RelAttn}\left(h_{z_t}^{(m-1)}, \left[\tilde{\mathbf{h}}^{(m-1)}, \mathbf{h}_{\mathbf{z}_{\le t}}^{(m-1)}\right]\right)\right) \\
h_{z_t}^{(m)} &= \text{LayerNorm}\left(\hat{h}_{z_t}^{(m)} + \text{PosFF}\left(\hat{h}_{z_t}^{(m)}\right)\right) \\
\hat{g}_{z_t}^{(m)} &= \text{LayerNorm}\left(g_{z_t}^{(m-1)} + \text{RelAttn}\left(g_{z_t}^{(m-1)}, \left[\tilde{\mathbf{h}}^{(m-1)}, \mathbf{h}_{\mathbf{z}_{\le t}}^{(m-1)}\right]\right)\right) \\
g_{z_t}^{(m)} &= \text{LayerNorm}\left(\hat{g}_{z_t}^{(m)} + \text{PosFF}\left(\hat{g}_{z_t}^{(m)}\right)\right)
\end{aligned}$$

Target-aware prediction distribution:

$$p_\theta(X_{z_t} = x \mid \mathbf{x}_{z<t}) = \frac{\exp\left(e(x)^\top g_{z_t}^{(M)}\right)}{\sum_{x'} \exp\left(e(x')^\top g_{z_t}^{(M)}\right)},$$

Input

[A, SEP, B, SEP, CLS]

- Sentences randomly sampled for A and B

  (50% consecutive, 50% non-consecutive)

- [A, B] concatenated & permuted

- SEP, CLS not counted for target

Relative segment embedding?

Relative Segment Encoding

BERT는 absolute segment embedding을 사용하지만 XLNet은 relative position encoding과 비슷한 원리로 relative segment encoding을 적용하였습니다. 전체 sequence에서 주어진 position i, j 가 같은 segment라면 $s_{ij} = s_+$, 아니면 $s_{ij} = s_-$ 를 사용하며, s_+ 와 s_- 는 각 attention head에 존재하는 학습가능한(learnable) paramenters입니다.

이를 통해 2가지 이득이 있는데, 첫 번째는 relative encoding의 inductive bias가 generalization을 향상시킨다는 것이고, 두 번째는 둘 이상의 segment를 갖는 fine-tuning 테스크에 대한 가능성을 열어줬다는 것입니다.

Training

(1) BooksCorpus, (2) English Wikipedia +
(3) Giga5, (4) Clue Web2012-B, (5) Common Crawl dataset

512 TPU v3,
24 Transformer XL layers,
bs 2048,
500k (underfitting)
2.5 days

XLNet vs the World: RACE

- 100k sets (Q&A)
- Long passages
- Multiple-choice questions
- Middle & High

Passage:

In a small village in England about 150 years ago, a mail coach was standing on the street. It didn't come to that village often. People had to pay a lot to get a letter. The person who sent the letter didn't have to pay the postage, while the receiver had to.
"Here's a letter for Miss Alice Brown," said the mailman.
" I'm Alice Brown," a girl of about 18 said in a low voice.
Alice looked at the envelope for a minute, and then handed it back to the mailman.
"I'm sorry I can't take it, I don't have enough money to pay it", she said.
A gentleman standing around were very sorry for her. Then he came up and paid the postage for her.
When the gentleman gave the letter to her, she said with a smile, " Thank you very much, This letter is from Tom. I'm going to marry him. He went to London to look for work. I've waited a long time for this letter, but now I don't need it, there is nothing in it."
"Really? How do you know that?" the gentleman said in surprise.
"He told me that he would put some signs on the envelope. Look, sir, this cross in the corner means that he is well and this circle means he has found work. That's good news."
The gentleman was Sir Rowland Hill. He didn't forgot Alice and her letter.
"The postage to be paid by the receiver has to be changed," he said to himself and had a good plan.
"The postage has to be much lower, what about a penny? And the person who sends the letter pays the postage. He has to buy a stamp and put it on the envelope." he said . The government accepted his plan. Then the first stamp was put out in 1840. It was called the "Penny Black". It had a picture of the Queen on it.

Questions:

1): The first postage stamp was made _.
A. in England B. in America C. by Alice D. in 1910

2): The girl handed the letter back to the mailman because _ .
A. she didn't know whose letter it was
B. she had no money to pay the postage
C. she received the letter but she didn't want to open it
D. she had already known what was written in the letter

3): We can know from Alice's words that _ .
A. Tom had told her what the signs meant before leaving
B. Alice was clever and could guess the meaning of the signs
C. Alice had put the signs on the envelope herself
D. Tom had put the signs as Alice had told him to

4): The idea of using stamps was thought of by _ .
A. the government
B. Sir Rowland Hill
C. Alice Brown
D. Tom

5): From the passage we know the high postage made _ .
A. people never send each other letters
B. lovers almost lose every touch with each other
C. people try their best to avoid paying it
D. receivers refuse to pay the coming letters

Answer: ADABC

XLNet vs the World: RACE

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.

- Segment-recurrence catching longer term dependency?

XLNet vs the World: SQuAD 1.1

- 100k sets (Q&A)
- Answer (A segment of text)

Example

Question — "To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?"

Context — "Architecturally, the school has a Catholic character. Atop the Main Building\'s gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend "Venite Ad Me Omnes". Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. **It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858**. At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary."

Text — "Saint Bernadette Soubirous"

XLNet vs the World: SQuAD 1.1

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.

XLNet vs the World: Text Classification

Movie, Restaurant, Product etc.

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).

XLNet vs the World: 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.

XLNet vs the World: ClueWeb09-B

- Reranking top 100 search results retrieved from 50M

Model	NDCG@20	ERR@20
DRMM [12]	24.3	13.8
KNRM [8]	26.9	14.9
Conv [8]	28.7	18.1
BERT†	30.53	18.67
XLNet	**31.10**	**20.28**

Table 5: Comparison with state-of-the-art results on the test set of ClueWeb09-B, a document ranking task. † indicates our implementations.

Ablation Study

- Permutation **O**
  - XLNet > BERT & Transformer-XL
- Segment-level recurrence **O**
  - XLNet > Transformer-XL > BERT in RACE and SQuAD 2.0
- Span-based prediction **O**
- Bidirectional input **O**
- Next sentence prediction **X**

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: 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.