循环智能
Recurrent



XLNet and Beyond

杨植麟

循环智能
Recurrent

Part I: XLNet

- NeurIPS 2019 Oral
- Acceptance Rate 0.5%

- Joint work with Zihang Dai et al.

循环智能 Recurrent

Learning from Unlabeled Data





Unlabeled data

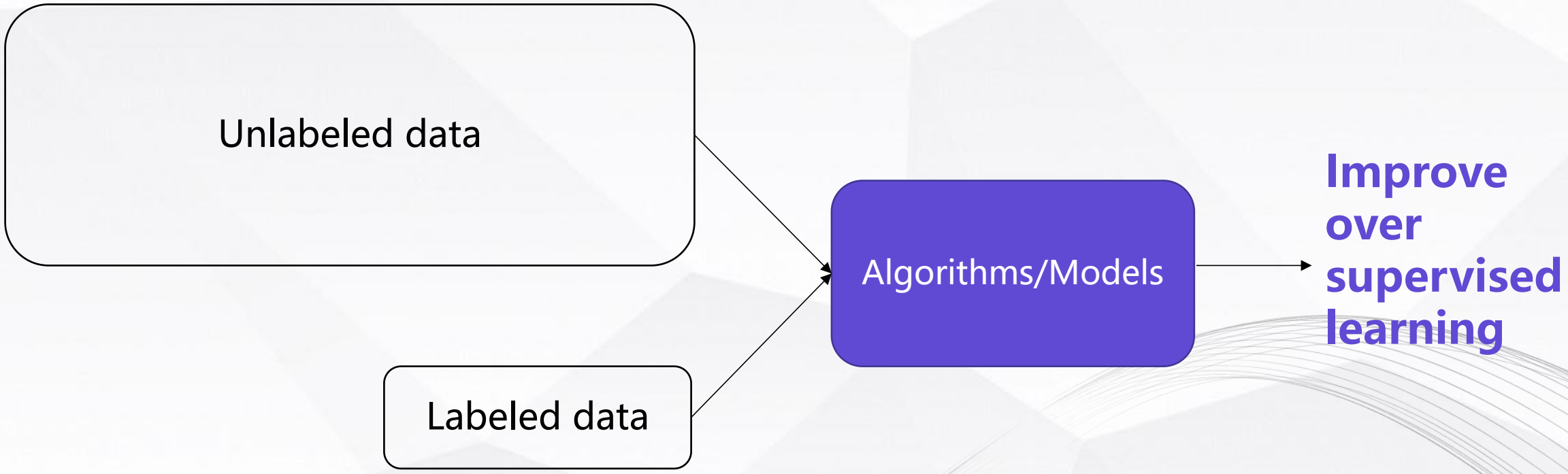
Abundant (1000x more), accessible

Labeled data

Scarce, expensive

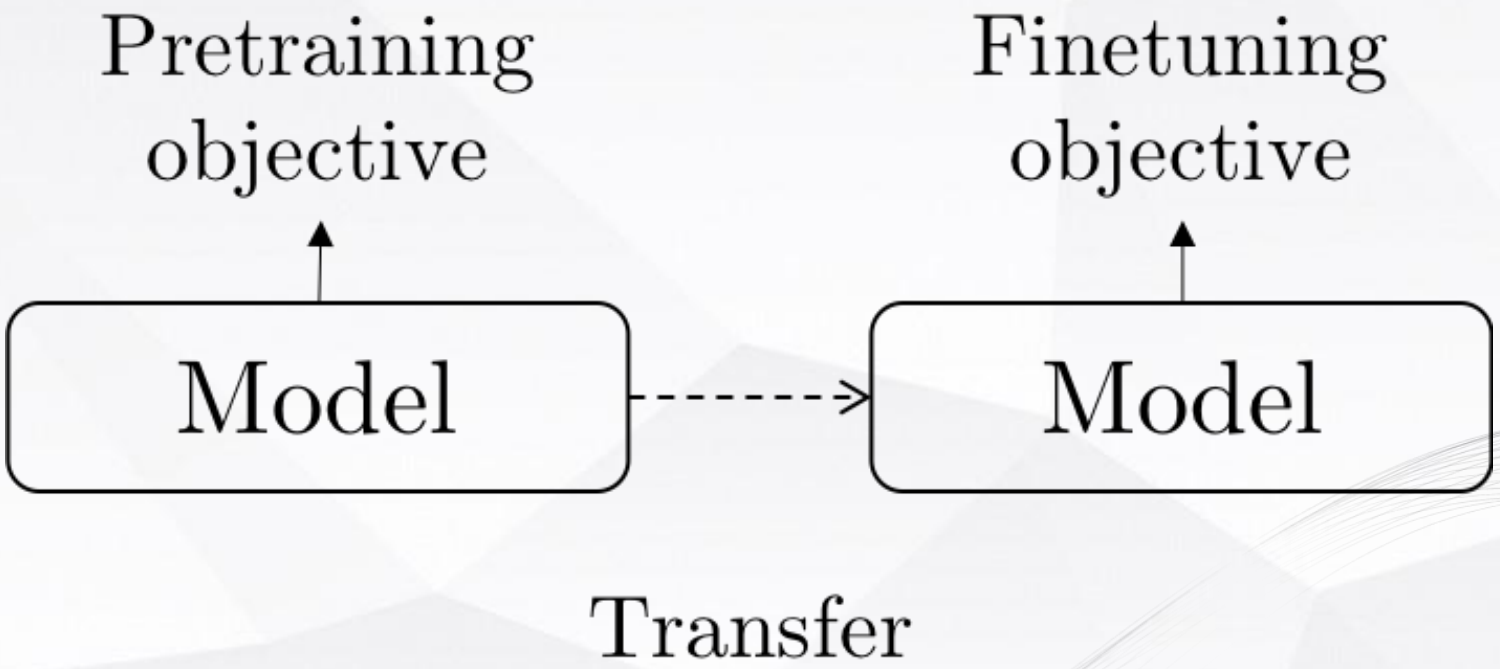
循环智能 Recurrent

Unsupervised Pretraining



循环智能 Recurrent

Unsupervised Pretraining



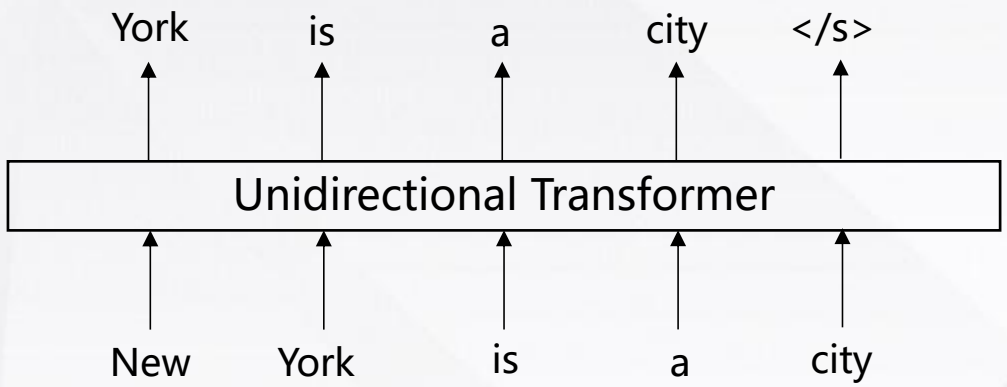
循环智能 Recurrent

Related Work

- RBMs (Salakhutdinov et al 2007), Autoencoders (Vincent et al 2008), Jigsaw (Noroozi and Favaro 2016), GANs (Donahue and Simonyan 2019)…
- word2vec (Mikolov et al 2013), GloVe (Pennington et al 2014)
- Semi-supervised sequence learning (Dai and Le 2015), ELMo (Peters et al 2017), CoVe (McCann et al 2017), GPT (Radford et al 2018), BERT (Devlin et al 2018)

Two Objectives for Pretraining

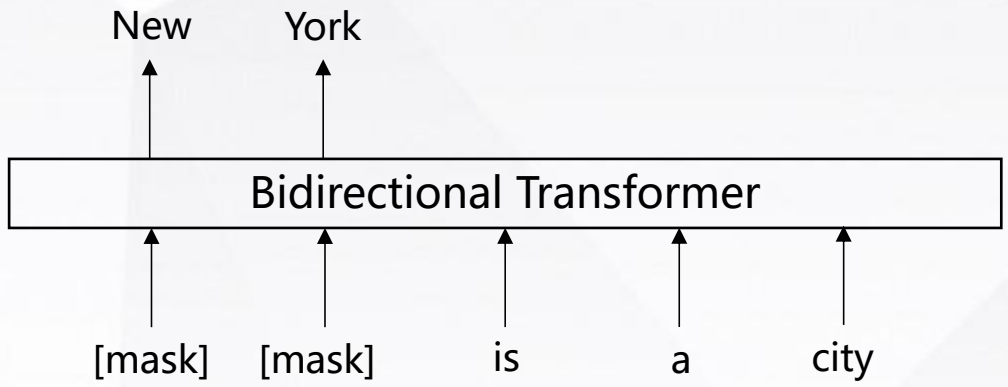
Auto-regressive (AR) language modeling



$$\log p(\mathbf{x}) = \sum_{t=1}^{T} \log p(x_t | \mathbf{x}_{<t})$$

Not able to model bidirectional context. ☹

(Denoising) Auto-encoding (AE)



$$\log p(\bar{\mathbf{x}} | \hat{\mathbf{x}}) = \sum_{t=1}^{T} \text{mask}_t \log p(x_t | \hat{\mathbf{x}})$$

Predicted tokens are independent of each other. ☹
[mask] is not used during finetuning. ☹

循环智能 Recurrent

New Objective: Permutation Language Modeling

- Sample a factorization order
- Determine the attention masks based on the order
- Optimize a standard language modeling objective

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{Z}_T}\left[\sum_{t=1}^{T}\log p(x_{z_t}|\mathbf{x}_{\mathbf{z}<t})\right]$$

- Benefits:
  - Autoregressive, avoiding disadvantages of AE
  - Able to model bidirectional context

循环智能 Recurrent

Examples

Factorization order: New York is a city

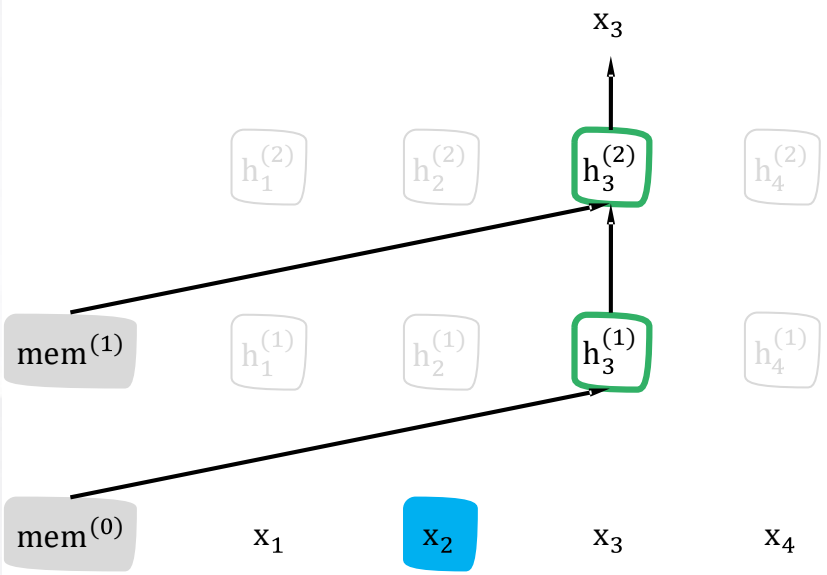
P(New York is a city)
= P(New) * P(York | New) * P(is | New York) * P(a | New York is) * P(city | New York is a)

Factorization order: city a is New York

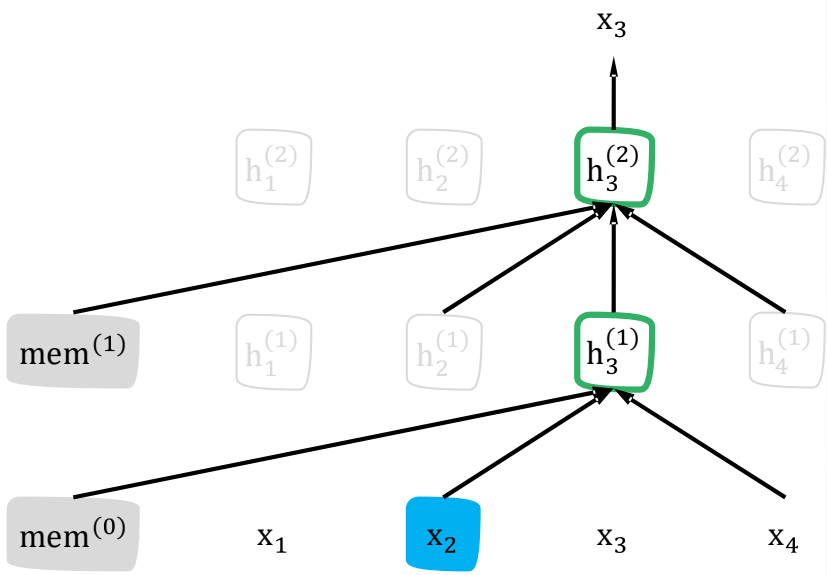
P(New York is a city)
= P(city) * P(a | city) * P(is | city a) * P(New | city a is) * P(York | city a is New)

Sequence order is not shuffled.
Attention masks are changed to reflect factorization order.

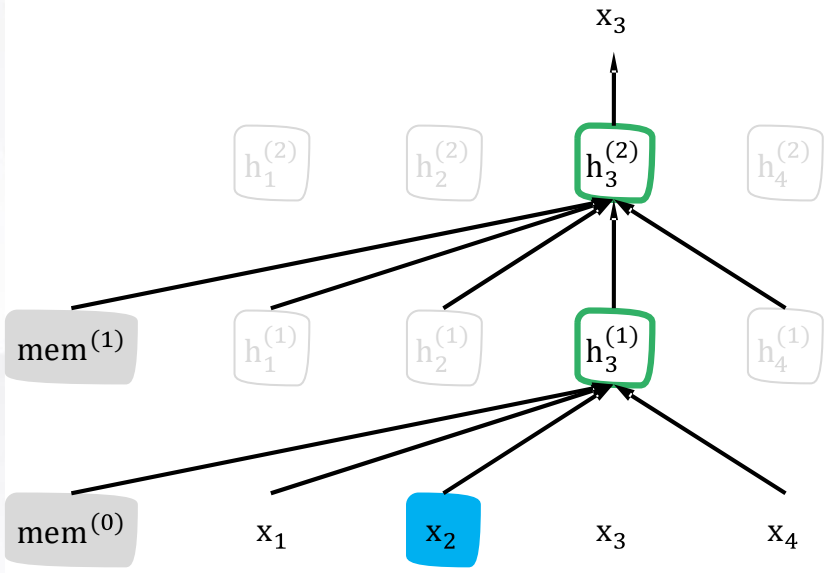
循环智能
Recurrent



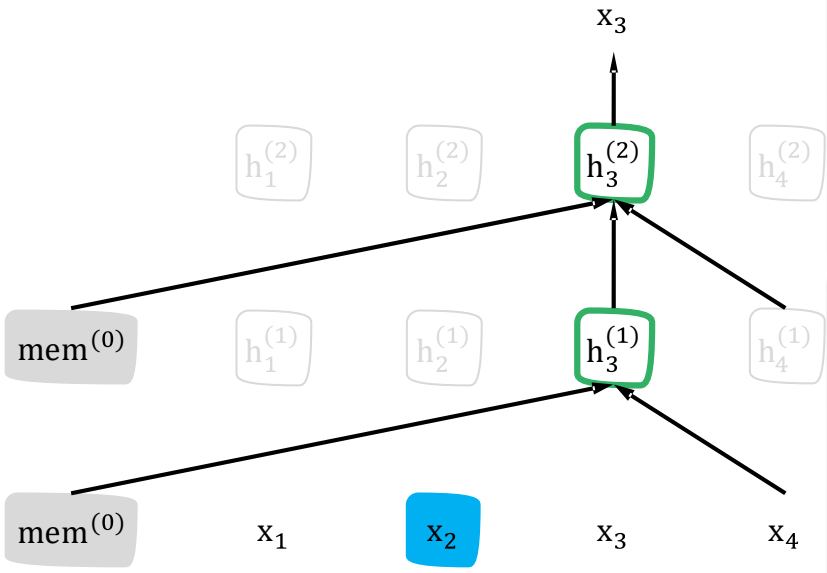
Factorization order: 3 → 2 → 4 → 1



Factorization order: 2 → 4 → 3 → 1



Factorization order: 1 → 4 → 2 → 3



Factorization order: 4 → 3 → 1 → 2

循环智能 Recurrent

Comparing XLNet and BERT objectives

BERT objective (auto-encoding)

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

New and York are independent. ☹

XLNet objective (auto-regressive)

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

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

Able to model dependency between New and York. ☺
Able to model bidirectional context. ☺
Factorize the joint probability using a product rule that holds universally.

循环智能 Recurrent

Reparameterization

Standard Parameterization

$$p_\theta(X_{z_t} = x \mid \mathbf{x}_{\mathbf{z}_{<t}}) = \frac{e(x)^\top h_\theta(\mathbf{x}_{\mathbf{z}_{<t}})}{\sum_{x'} e(x')^\top h_\theta(\mathbf{x}_{\mathbf{z}_{<t}})}$$

h does not contain the position of the target.

$$p(X_3 = \text{is} \mid \text{New York}) = p(X_4 = \text{is} \mid \text{New York}) = p(X_5 = \text{is} \mid \text{New York})$$

Reduced to predicting a bag of words.

Solution: condition the distribution on the position.

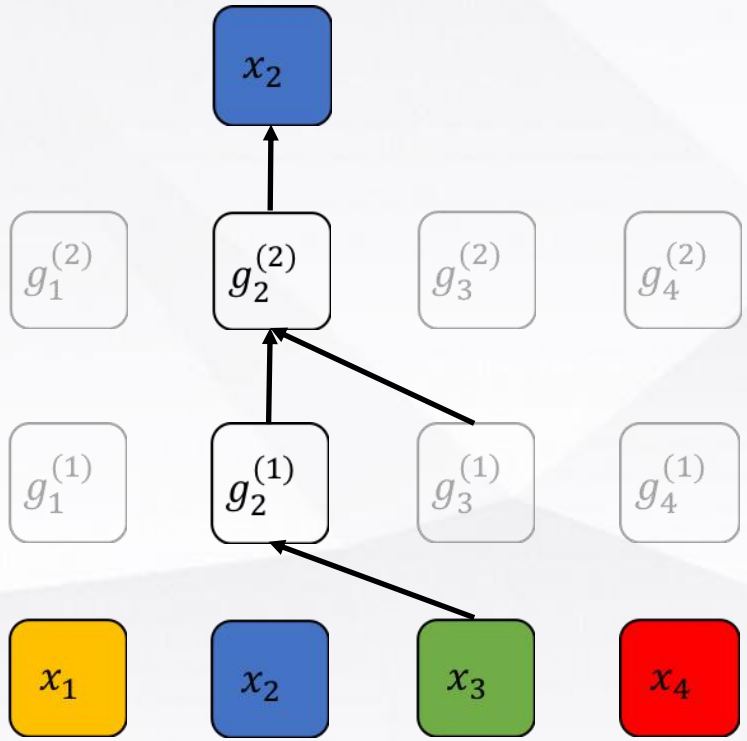
$$p_\theta(X_{z_t} = x \mid \mathbf{x}_{\mathbf{z}_{<t}}) = \frac{e(x)^\top g_\theta(\mathbf{x}_{\mathbf{z}_{<t}}, \boxed{z_t})}{\sum_{x'} e(x')^\top g_\theta(\mathbf{x}_{\mathbf{z}_{<t}}, \boxed{z_t})}$$

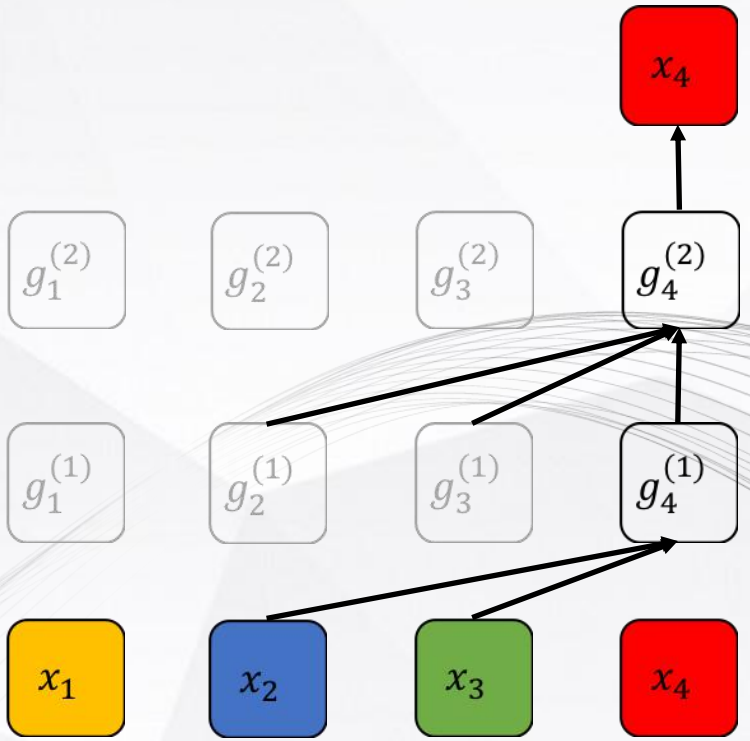
"Stand at" z_t and predict self

循环智能 Recurrent

How to Formulate Features g

Let $g_i^{(l)}$ denote the feature of the i-th token on layer l
Suppose the factorization order is 3 2 4 1



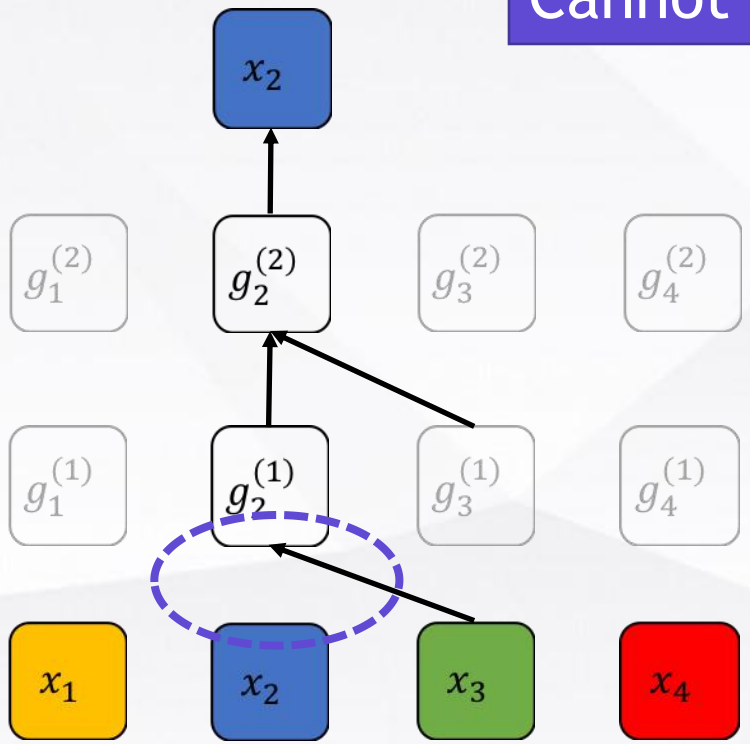


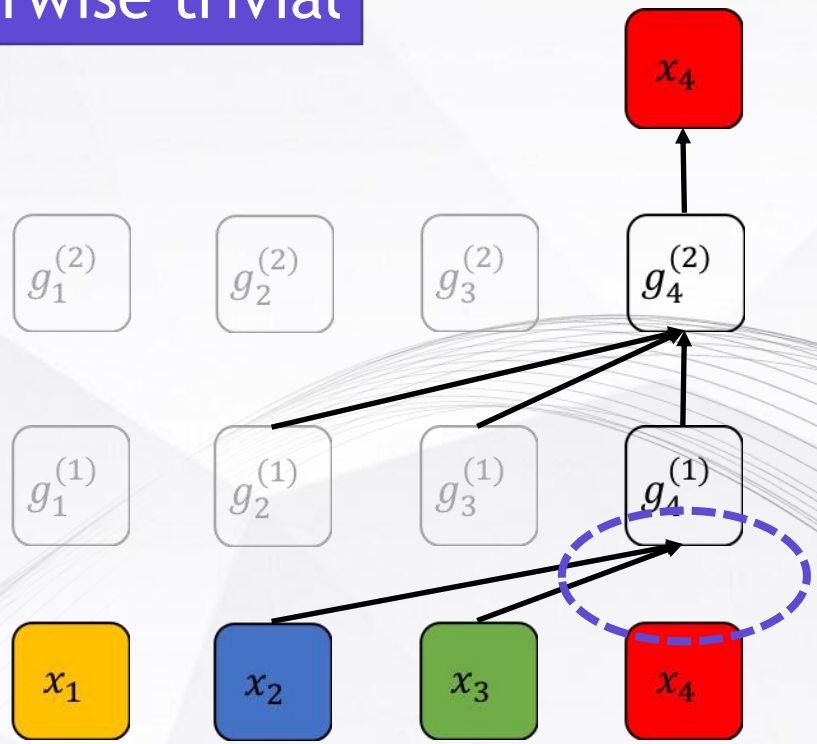
循环智能 Recurrent

How to Formulate Features g

Let $g_i^{(l)}$ denote the feature of the i-th token on layer l
Suppose the factorization order is 3 2 4 1

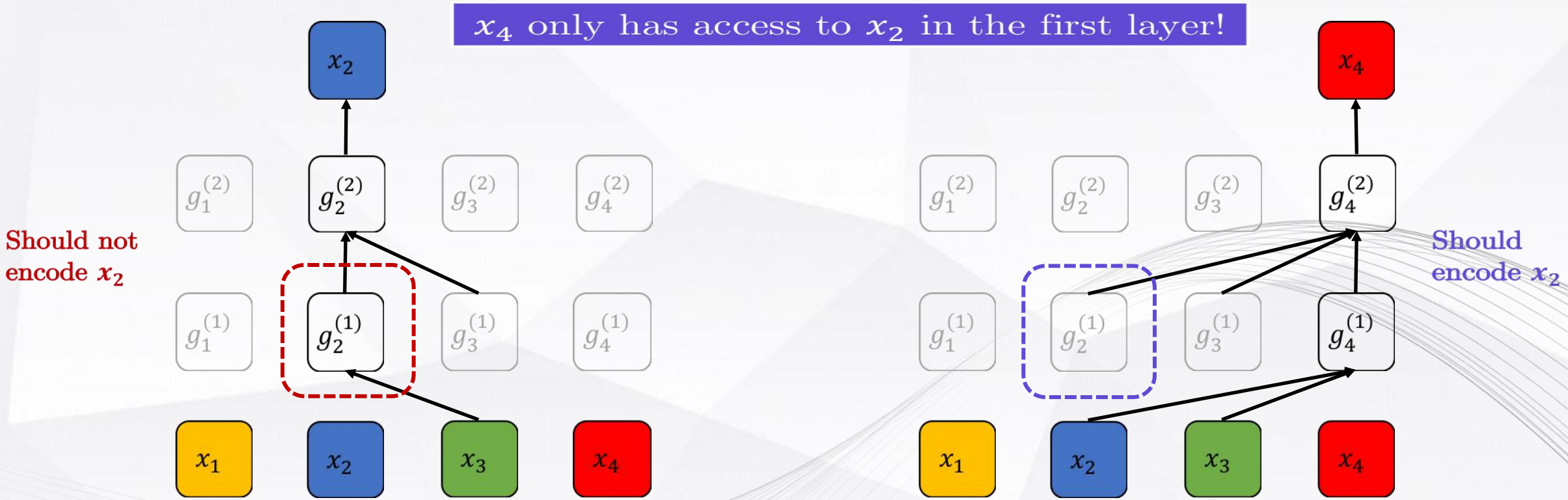
Cannot see self, otherwise trivial





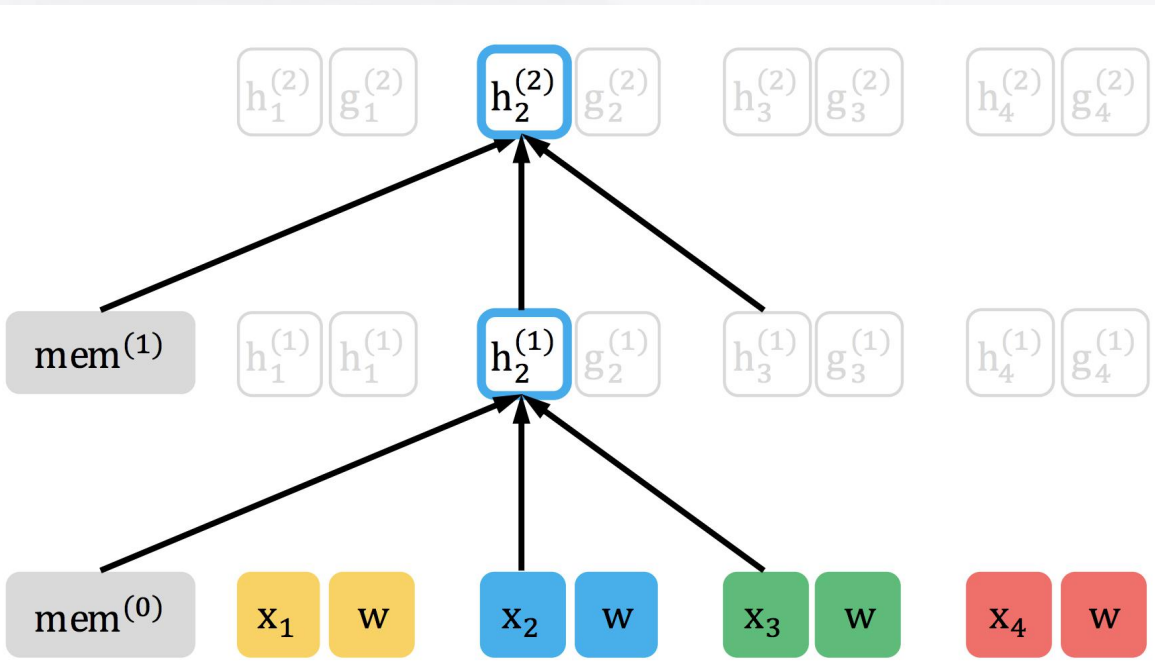
How to Formulate Features g

Let $g_i^{(l)}$ denote the feature of the i-th token on layer l
Suppose the factorization order is 3 2 4 1

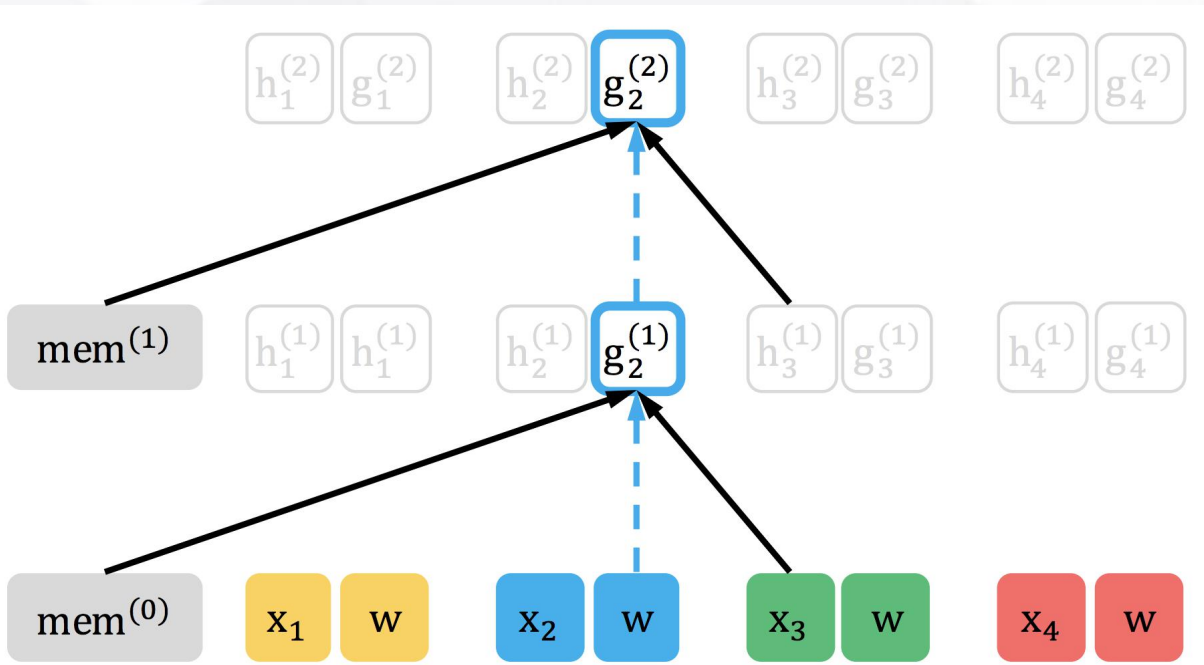


Two-Stream Attention

- Factorization order: 3, 2, 4, 1

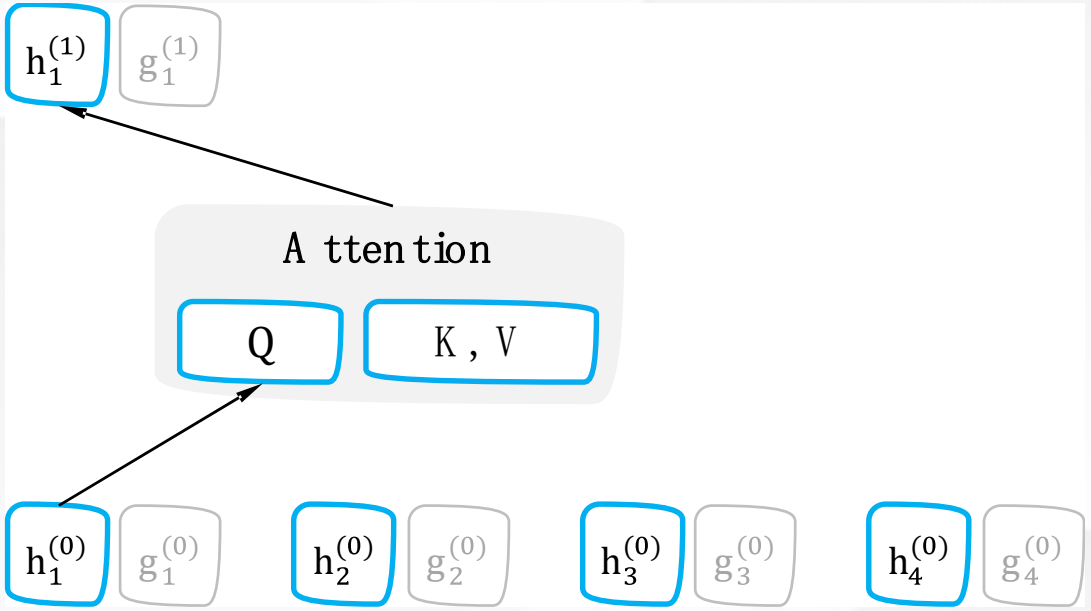


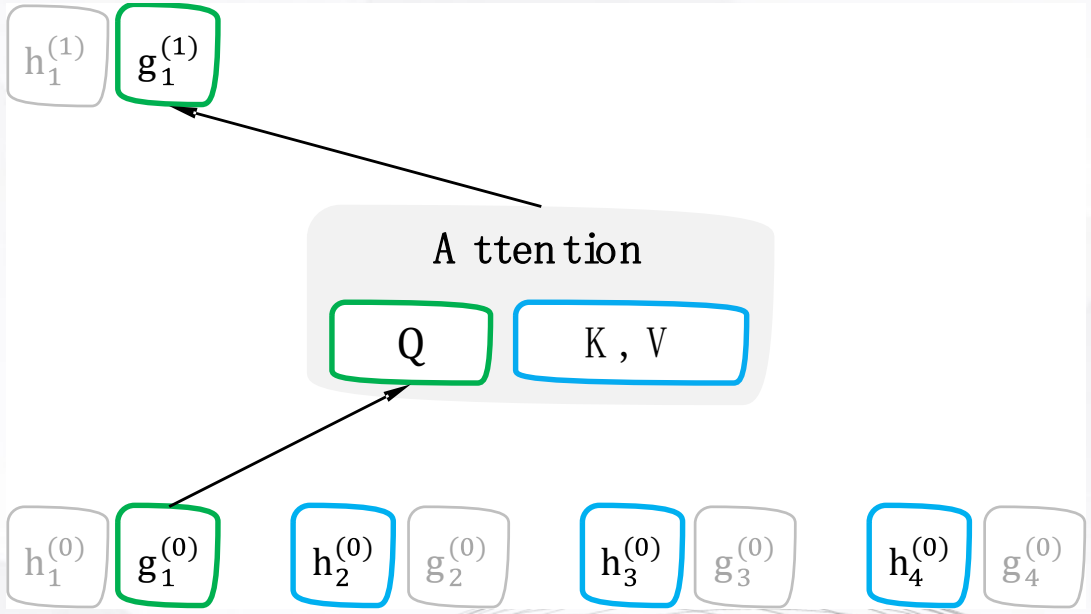
Content stream



Query stream

Two-Stream Attention





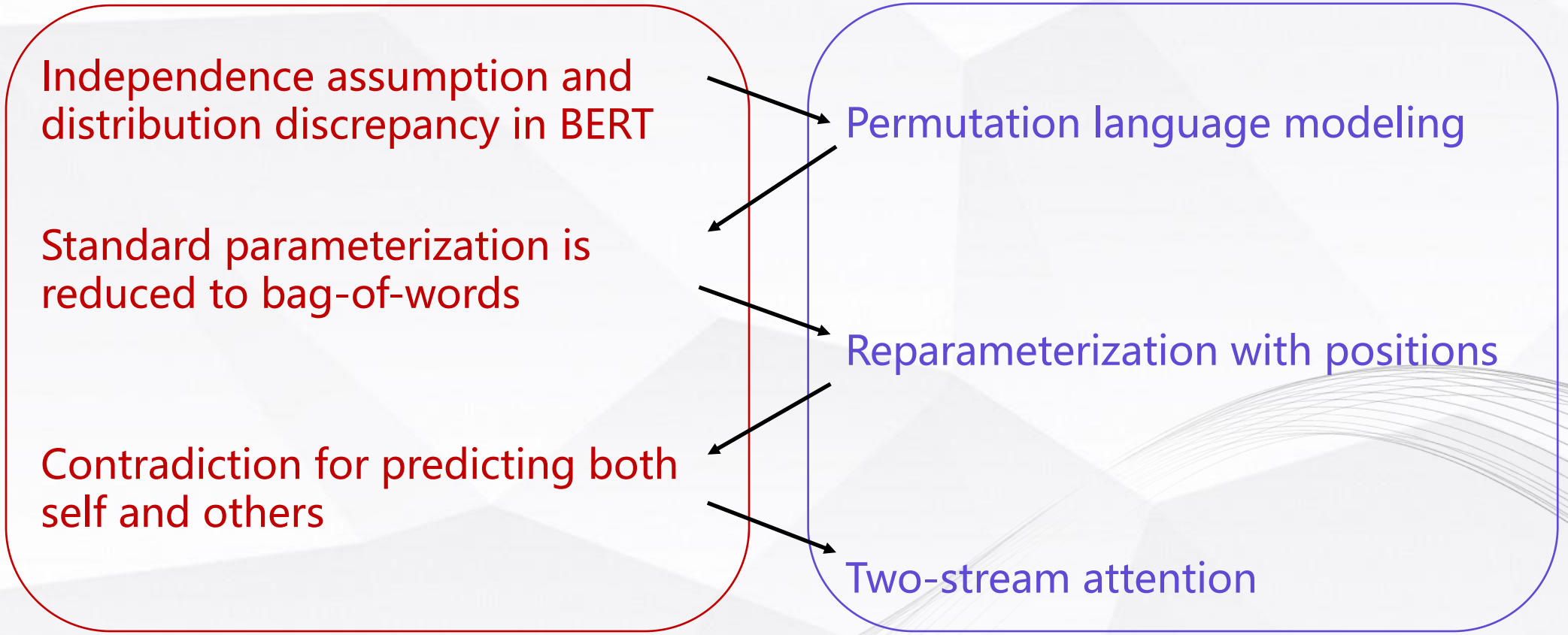
$$g_{z_t}^{(m)} \leftarrow \text{Attention}(\mathrm{Q} = g_{z_t}^{(m-1)}, \mathrm{KV} = \mathbf{h}_{\mathbf{z}_{<t}}^{(m-1)}; \theta), \quad \text{(query stream: use } z_t \text{ but cannot see } x_{z_t})$$

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

At first layer, h is the word embeddings, and g is a trainable parameter.
Only h is used during finetuning. The last g is used for optimizing the LM loss.

循环智能 Recurrent

Summarizing XLNet



循环智能 Recurrent

Bridging the Gap Between LM and Pretraining

- XLNet connects LM and pretraining
- Benefits
  - Taking the advantages of LM, while being able to model bidirectional contexts
  - Progress in LM directly connected to target tasks
  - LM research relies on less computational resources
- Incorporating Transformer-XL into pretraining

循环智能 Recurrent

Experiment 1: Fair Comparison with BERT

Model	SQuAD1.1	SQuAD2.0	RACE	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B
BERT-Large (Best of 3)	86.7/92.8	82.8/85.5	75.1	87.3	93.0	91.4	74.0	94.0	88.7	63.7	90.2
XLNet-Large-wikibooks	88.2/94.0	85.1/87.8	77.4	88.4	93.9	91.8	81.2	94.4	90.0	65.2	91.1

- XLNet trained with the same data and hyperparameters as in BERT.
- We use the best of 3 BERT variants for comparison; i.e., the original BERT, BERT with whole word masking, and BERT without NSP.

XLNet outperforms BERT on all considered tasks, often by a large margin.

循环智能 Recurrent

Experiment 2: Fair Comparison with RoBERTa

RACE	Accuracy	Middle	High
GPT [28]	59.0	62.9	57.4
BERT [25]	72.0	76.6	70.1
BERT+DCMN* [38]	74.1	79.5	71.8
RoBERTa [21]	83.2	86.5	81.8
XLNet	**85.4**	**88.6**	**84.0**

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
RoBERTa [21]	88.9	94.6	RoBERTa [21]	86.5	89.4
XLNet	**89.7**	**95.1**	XLNet	**87.9**	**90.6**

- XLNet trained with the same data and hyperparameters as in RoBERTa.

XLNet outperforms RoBERTa on all considered tasks, often by a large margin.

循环智能 Recurrent

Ablation Study with Base Models

#	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

循环智能 Recurrent

Ablation Study with Base Models

#	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

Permutation language modeling improves performance.

循环智能 Recurrent

Ablation Study with Base Models

#	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

Backbone model Transformer-XL improves performance on long-sequence tasks.

循环智能
Recurrent

Part II: Rethinking

Some thoughts about NLP and industrialization.

循环智能 Recurrent

Why Transformers and Pretraining?

Belief: Transformers are good because they are more efficient!
Fact: Transformers are slower than LSTMs given the same number of parameters.

Belief: Transformers are good because they are better on machine translation!
Fact: Transformers, RNNs, CNNs are equally good for machine translation. [Chen et al 2018, Wu et al 2018]

Belief: Transformers are good because they use nothing but attention!
Fact: well...

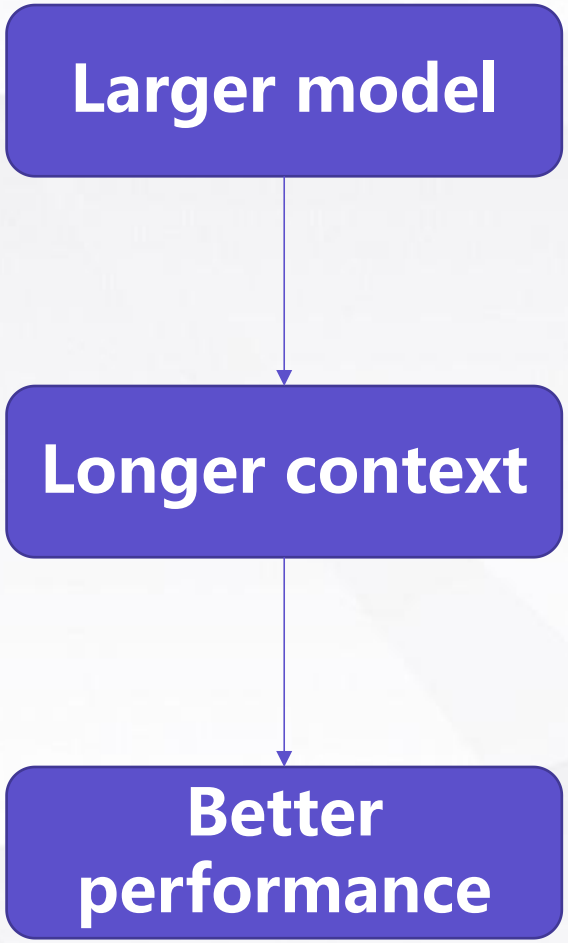
Belief: Pretraining is awesome because it learns contextualized embeddings!
Question: Do ordinary RNNs also learn contextualized embeddings?

What Did Transformers and Pretraining Do?



循环智能 Recurrent

Transformers Enable “Big Model”



#1: Transformers scale much better with more parameters.

“Exploring the Limits of Language Modeling
Jozefowicz et al 2016

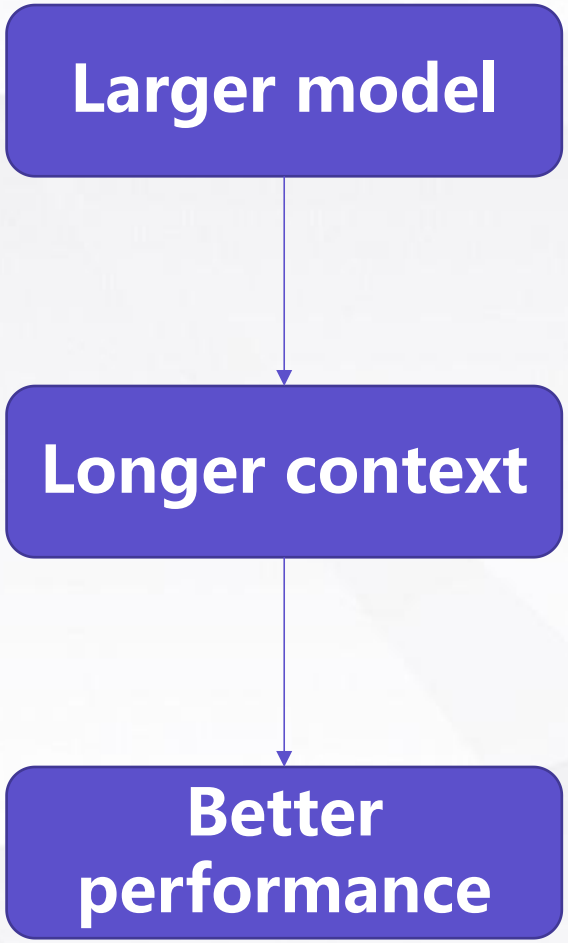
LSTM-8192-1024, 1.8 billion params, ppl 30.6
LSTM-8192-2048, 3.3 billion params, ppl 32.2

Dai, Yang et al 2016
Transformer-XL Base, 0.46 billion params, ppl 23.5
Transformer-XL Large, 0.8 billion params, ppl 21.8

ppl=perplexity, the lower the better

循环智能 Recurrent

Transformers Enable “Big Model”



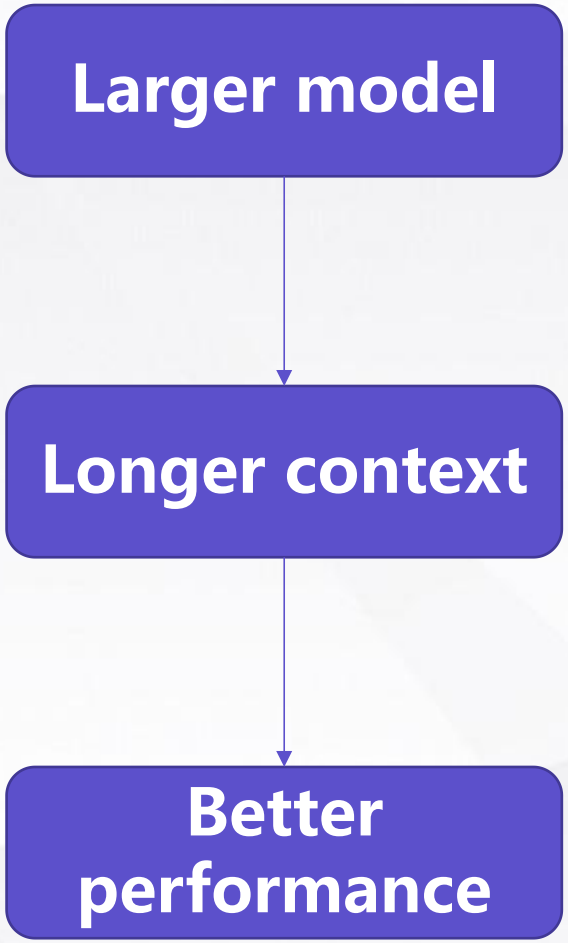
#2: Transformers can be much deeper

Al Rfou et al develop a 64-layer Transformer that was state-of-the-art for char-level LM.

Few (if any) deep LSTMs have been trained to obtain state-of-the-art results in NLP.

循环智能 Recurrent

Transformers Enable “Big Model”



#3: Transformers learn much longer dependency

Transformer-XL [Dai Yang et al] is able to generate relatively coherent text of **a few thousand** tokens.
Nothing similar has been reported with LSTMs.

Transformer-XL learns much longer dependency than LSTMs using the metrics of RECL. [Dai Yang et al]

循环智能 Recurrent

Pretraining Enables “Big Data”



1. NLP tasks are extremely diverse (think about GLUE, question answering, document ranking, text classification, information extraction, and 100 more tasks and subtasks)
2. As a result, it is hard to have a huge benchmark (at the scale of ImageNet)
3. But it does not matter, because unlabeled data rock!

循环智能 Recurrent

“The Bitter Lesson” – Richard Sutton

- “The biggest lesson that can be learned from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin.”

Pretraining

Big data

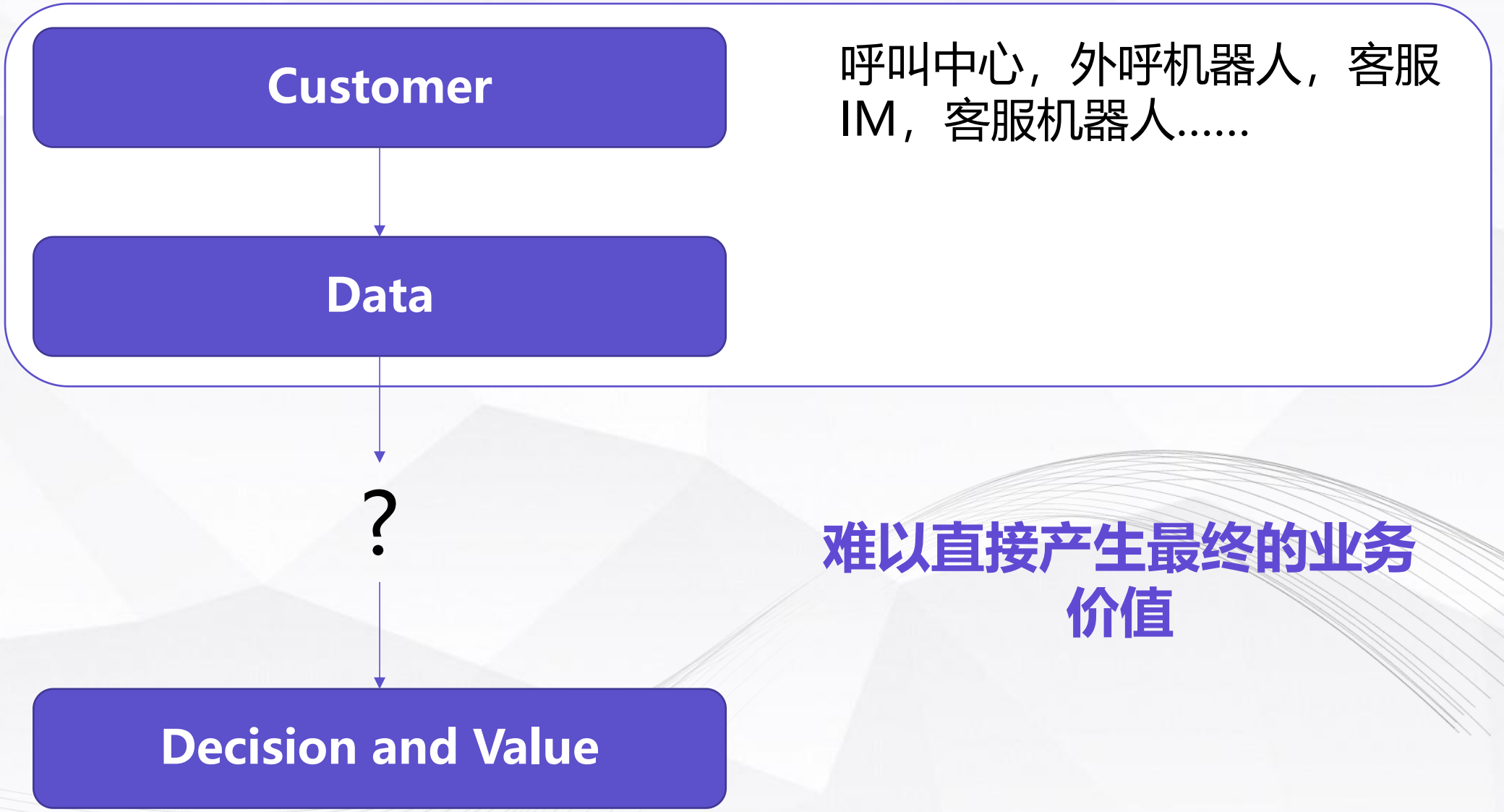
Transformers

Big model

江湖传说：水多加面，面多加水

循环智能 Recurrent

What NLP SaaS looked like before

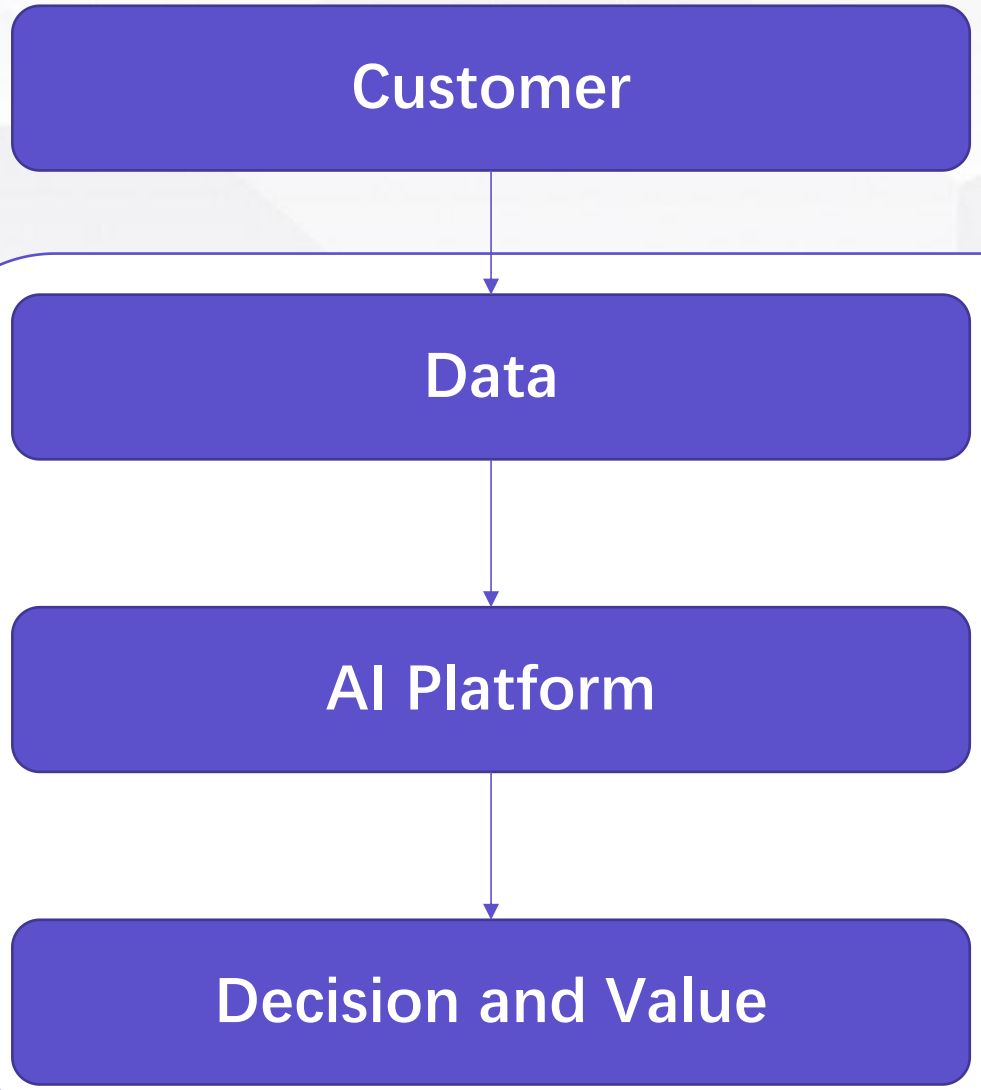


呼叫中心，外呼机器人，客服IM，客服机器人……

难以直接产生最终的业务价值

循环智能 Recurrent

What we do at Recurrent AI



AI销售中台

跨越对话数据和业务价值的鸿沟

Transformer(-XL) &
Pretraining (XLNet)

循环智能 Recurrent

落地效果

某数百亿市值公司的测试结果

	语音识别错误率	说话人分离错误率
Transformer-XL	-	-
某大厂A	+3.96%	+27.87%
某大厂B	+19.68%	+46.54%

大厂A（国内公认语音巨头）的错误率比我们高了3.96%和27.87%
大厂B（国内互联网巨头）的错误率比我们高了19.68%和46.54%

某教育独角兽公司的测试结果

	线索成单率
XLNet	5.37x
客户NLP模型	2.92x
人工运营	1.00x

客户NLP模型比人工运营提升2.92倍
XLNet比人工运营提升5.37倍

When, who, channel, message

循环智能 Recurrent

What is next?

Pretraining

Big data → Tradeoff between big data and high-quality data

Transformers

Big model → Tradeoff between model scale and model efficiency

循环智能
Recurrent

Thanks!

杨植麟