深度之眼
deepshare.net

本期论文主题:Transformer-xl

导师：Yamada

《Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context》

超出固定长度上下文的注意力语言模型

作者：Zhilin Yang、Zihang Dai

单位：google brain && cmu

发表会议及时间：ACL，2019

深度之眼
deepshare.net

前期知识储备

Pre-knowledge reserve

概率论

了解基本的概率论知识，掌握条件概率的概念和公式

Transformer

了解Transformer的结构，掌握Transformer的基本工作原理

Vanilla Trans

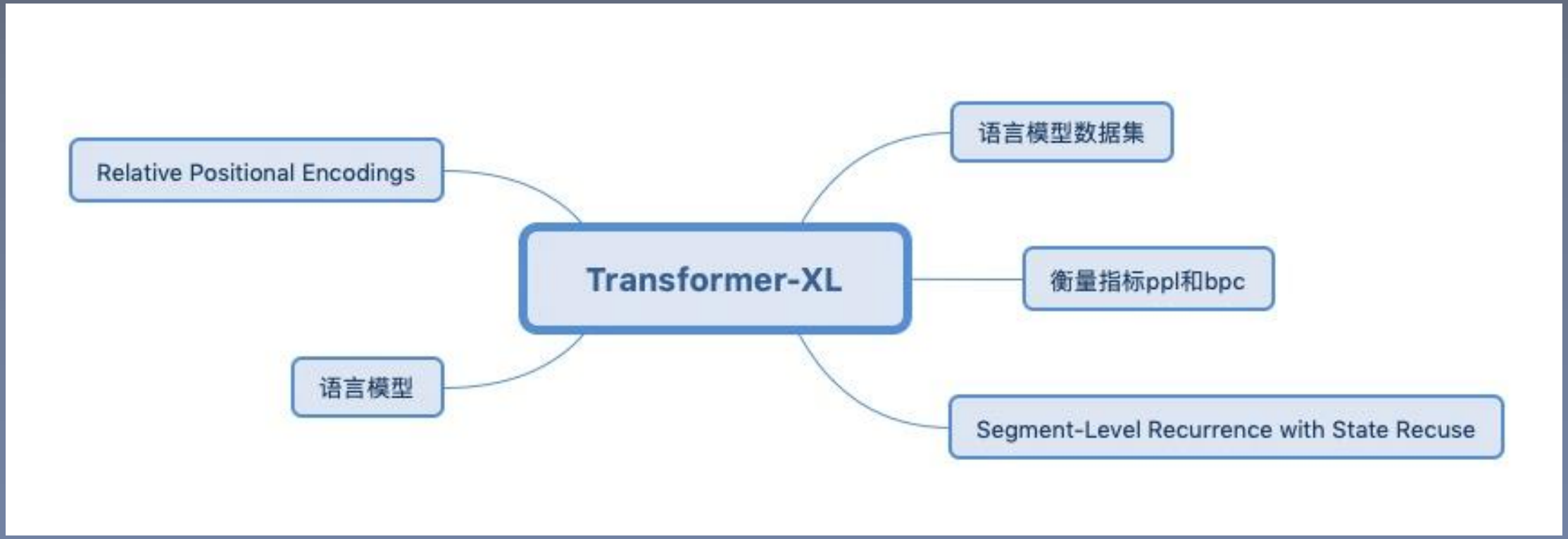
掌握Vanilla Model的基本工作原理。

注意力机制

了解注意力机制的思想，掌握注意力机制的分类和实现方式

学习目标

Learning objectives



深度之眼
deepshare.net

课程安排
The schedule of course

01 **Part 1** 论文导读 背景介绍

02 **Part2** Vanilla Transformer结构

03 **Part 3** Segment-Level Recurrence with State Reuse

04 **Part 4** **Relative Postional Encodings**

05 **Part 5** 小trick

06 **Part 6** 代码实现分析

深度之眼
deepshare.net

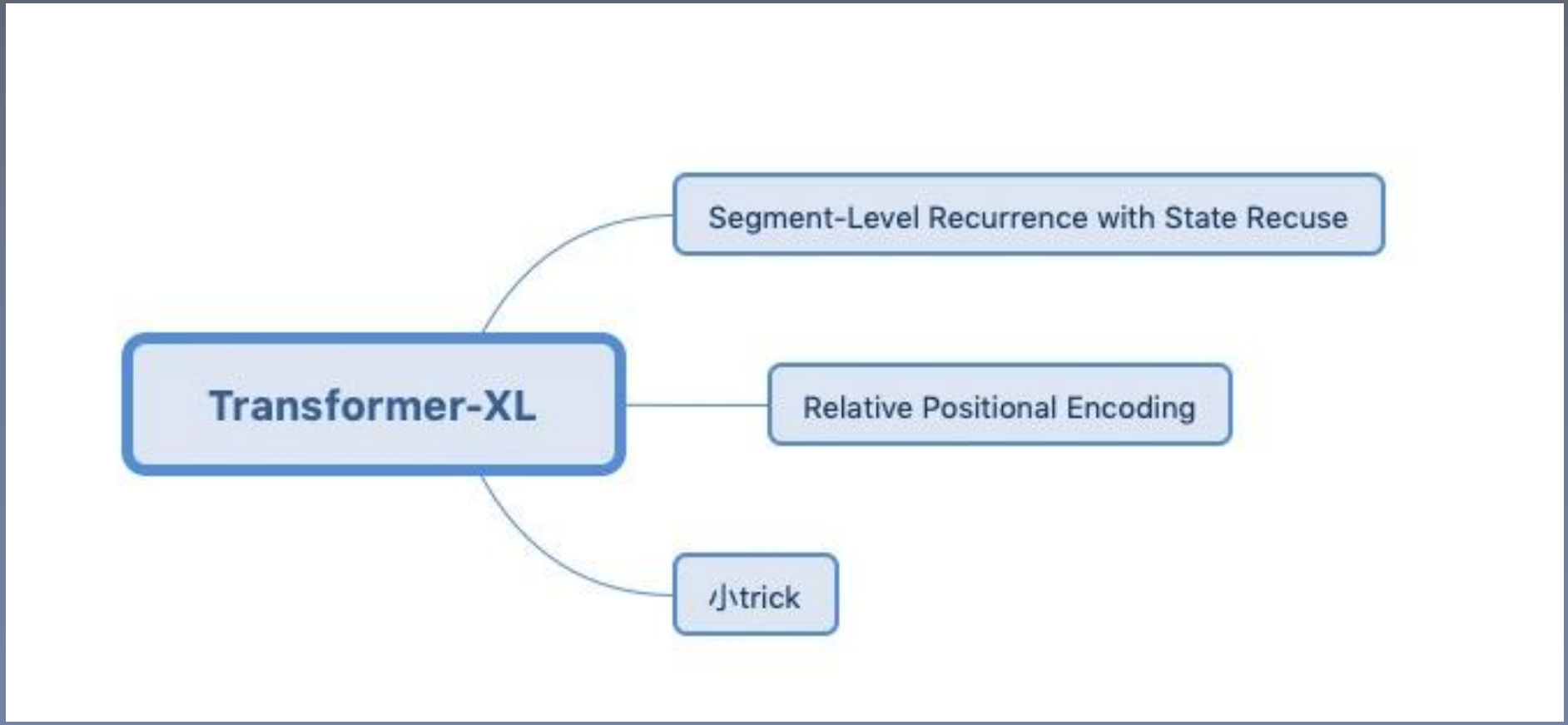
第一课：论文导读

The first lesson: the paper guide

深度之眼
deepshare.net

目 录

知识树



论文研究背景、成果及意义

研究背景
Research background


```
1    enwik8 （复杂格式）
2  '''Anarchism''' originated as a term of abuse first used against early [[working class]] [[radical]]
3
4    text8 （1行，10^8个字符）
5   anarchism originated as a term of abuse first used against early working class radicals including t
```

enwik8和text8数据集

	Tokens	Articles	clean
WikiText-103	103,227,021	28475	
enwiki8	100,000,000	243,426	
text8	100,000,000	243,426	true
One Billion Word	1,000,000,000		
Penn Treebank	1,000,000	2499	

研究背景

Research background

深度之眼
deepshare.net

重点 重点来了！

语言模型

语言模型是用来计算一个句子的概率的模型，判断这句话是否合理。

给定句子(词语序列)： 今天早上我去食堂吃饭电视

$$S = W_1, W_2, ..., W_k$$

语言模型概率：

$$P(S) = P(W_1, W_2, ..., W_k) = p(W_1)p(W_2|W_1)...P(W_k|W_1, W_2, ..., W_{k-1})$$

研究背景

Research background

深度之眼
deepshare.net

重点 重点来了！

语言模型

为了解决参数空间过大的问题，引入了马尔可夫假设：随意一个词的出现只与它前面出现的有限的一个或者几个词有关。

unigram:

$$P(S) = P(w_1) * P(w_2) * P(w_3) * ... * P(w_n)$$

bigram:

$$P(S) \approx P(w_1)P(w_2|w_1)P(w_3|w_2)..P(w_n|w_{n-1})$$

trigram:

$$P(S) \approx P(w_1)P(w_2|w_1)P(w_3|w_2,w_1)..P(w_n|w_{n-1},w_{n-2})$$

研究背景

Research background

重点　重点来了！

语言模型

这些概率参数都是通过大规模语料来计算。

$c(w_{i-1}, w_i)$

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

$c(w_{i-1})$

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

$$P(want|i) = c(i, want)/c(i) = 827/2533 \approx 0.33$$

$$P(< s > I want food < /s >) = P(I| < s >) * P(want|I) * P(food|want) * P(< /s > |food) = 0.000031$$

研究背景

Research background

深度之眼
deepshare.net

困惑度(PPL)

困惑度(perplexity)的基本思想:给测试集的句子赋予较高概率值的语言模型，当语言模型训练好之后，测试集中的句子都是正常句子，那么训练好的模型就是在测试集上的概率越高越好，公式如下：

$$PP(W) = P(w_1 w_2 ... w_N)^{-1/N} = \sqrt[N]{1/P(w_1 w_2 ... w_N)}$$

由公式可知：**语言模型越好，困惑度越小**

研究背景

Research background

重点 重点来了！

bits-per-character(BPC)

$$bpc(string) = 1/T \sum_{t=1}^{T} H(P_t, \overline{P_t})$$

$$= -1/T \sum_{t=1}^{T} \sum_{c=1}^{n} P_t(c) log_2 \overline{P_t}(c) = -1/T \sum_{t=1}^{T} log_2 \overline{P_t}(x_t)$$

当以每个单词为一个字符计算bpc时，存在：$2^{bpc} = ppl$

研究成果

Research Results

Model	#Param	PPL
Grave et al. (2016b) - LSTM	-	48.7
Bai et al. (2018) - TCN	-	45.2
Dauphin et al. (2016) - GCNN-8	-	44.9
Grave et al. (2016b) - LSTM + Neural cache	-	40.8
Dauphin et al. (2016) - GCNN-14	-	37.2
Merity et al. (2018) - QRNN	151M	33.0
Rae et al. (2018) - Hebbian + Cache	-	29.9
Ours - Transformer-XL Standard	151M	**24.0**
Baevski and Auli (2018) - Adaptive Input$^{\diamond}$	247M	20.5
Ours - Transformer-XL Large	257M	**18.3**

Model	#Param	bpc
Ha et al. (2016) - LN HyperNetworks	27M	1.34
Chung et al. (2016) - LN HM-LSTM	35M	1.32
Zilly et al. (2016) - RHN	46M	1.27
Mujika et al. (2017) - FS-LSTM-4	47M	1.25
Krause et al. (2016) - Large mLSTM	46M	1.24
Knol (2017) - cmix v13	-	1.23
Al-Rfou et al. (2018) - 12L Transformer	44M	1.11
Ours - 12L Transformer-XL	41M	**1.06**
Al-Rfou et al. (2018) - 64L Transformer	235M	1.06
Ours - 18L Transformer-XL	88M	1.03
Ours - 24L Transformer-XL	277M	**0.99**

1 在WikiText-103数据集上的ppl为18.3，start-of-the-art为20.5。

2 在enwik8数据集上的bpc为1.06，start-of-the-art为1.11，12层layer的效果达到了start-of-the-art的64层的效果并且只用到了后者17%的参数。

研究成果

Research Results

Model	#Param	bpc
Cooijmans et al. (2016) - BN-LSTM	-	1.36
Chung et al. (2016) - LN HM-LSTM	35M	1.29
Zilly et al. (2016) - RHN	45M	1.27
Krause et al. (2016) - Large mLSTM	45M	1.27
Al-Rfou et al. (2018) - 12L Transformer	44M	1.18
Al-Rfou et al. (2018) - 64L Transformer	235M	1.13
Ours - 24L Transformer-XL	277M	**1.08**

Model	#Param	PPL
Shazeer et al. (2014) - Sparse Non-Negative	33B	52.9
Chelba et al. (2013) - RNN-1024 + 9 Gram	20B	51.3
Kuchaiev and Ginsburg (2017) - G-LSTM-2	-	36.0
Dauphin et al. (2016) - GCNN-14 bottleneck	-	31.9
Jozefowicz et al. (2016) - LSTM	1.8B	30.6
Jozefowicz et al. (2016) - LSTM + CNN Input	1.04B	30.0
Shazeer et al. (2017) - Low-Budget MoE	~5B	34.1
Shazeer et al. (2017) - High-Budget MoE	~5B	28.0
Shazeer et al. (2018) - Mesh Tensorflow	4.9B	24.0
Baevski and Auli (2018) - Adaptive Input$^{\diamond}$	0.46B	24.1
Baevski and Auli (2018) - Adaptive Input$^{\diamond}$	1.0B	23.7
Ours - Transformer-XL Base	0.46B	23.5
Ours - Transformer-XL Large	0.8B	**21.8**

Model	#Param	PPL
Inan et al. (2016) - Tied Variational LSTM	24M	73.2
Zilly et al. (2016) - Variational RHN	23M	65.4
Zoph and Le (2016) - NAS Cell	25M	64.0
Merity et al. (2017) - AWD-LSTM	24M	58.8
Pham et al. (2018) - Efficient NAS	24M	58.6
Liu et al. (2018) - Differentiable NAS	23M	56.1
Yang et al. (2017) - AWD-LSTM-MoS	22M	55.97
Melis et al. (2018) - Dropout tuning	24M	55.3
Ours - Transformer-XL	24M	**54.52**
Merity et al. (2017) - AWD-LSTM+Finetune†	24M	57.3
Yang et al. (2017) - MoS+Finetune†	22M	**54.44**

3 在text8数据集上的bpc为1.08，start-of-the-art为1.13。

4 在One Billion Word数据集上的ppl为21.8，start-of-the art为23.7。

5 在Penn Treebank数据集上的ppl为54.5，是在没有经过fine-tuning的情况下。

研究意义

Research Meaning

重点 重点来了！

Transformer-xl历史意义

- **提出segment-level recurrence mechanism机制，以及相对位置编码机制。**
- **为XLNet的到来做好了铺垫**

研究意义

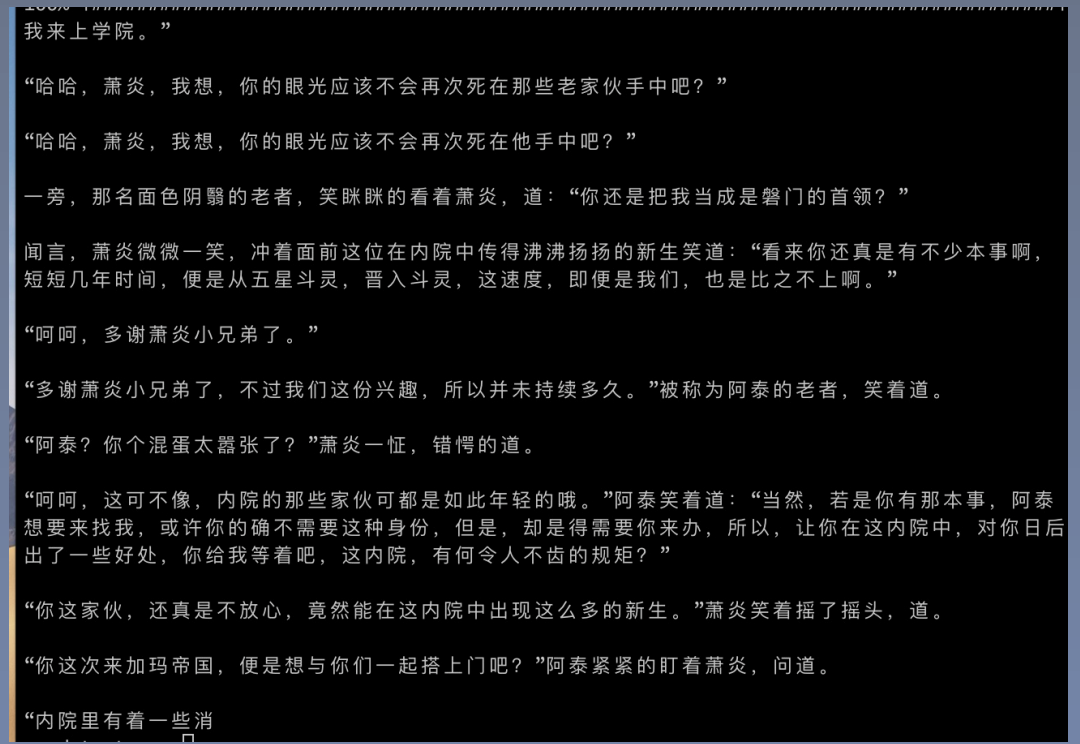
Research Meaning

重点 重点来了！

Transformer-xl历史意义

- **提出segment-level recurrence mechanism机制，以及相对位置编码机制。**
- **为XLNet的到来做好了铺垫**

Transformer-xl采用了片段级递归机制，从而提高了语言模型的长距离依赖，对长文本的编码更加有效。Transformer-xl在文本生成任务上具有不错的效果。



我来上学院。”

“哈哈，萧炎，我想，你的眼光应该不会再次死在那些老家伙手中吧？”

“哈哈，萧炎，我想，你的眼光应该不会再次死在他手中吧？”

一旁，那名面色阴翳的老者，笑眯眯的看着萧炎，道：“你还是把我当成是磐门的首领？”

闻言，萧炎微微一笑，冲着面前这位在内院中传得沸沸扬扬的新生笑道：“看来你还真是有不少本事啊，短短几年时间，便是从五星斗灵，晋入斗灵，这速度，即便是我们，也是比之不上啊。”

“呵呵，多谢萧炎小兄弟了。”

“多谢萧炎小兄弟了，不过我们这份兴趣，所以并未持续多久。”被称为阿泰的老者，笑着道。

“阿泰？你个混蛋太嚣张了？”萧炎一怔，错愕的道。

“呵呵，这可不像，内院的那些家伙可都是如此年轻的哦。”阿泰笑着道：“当然，若是你有那本事，阿泰想要来找我，或许你的确不需要这种身份，但是，却是得需要你来办，所以，让你在这内院中，对你日后出了一些好处，你给我等着吧，这内院，有何令人不齿的规矩？”

“你这家伙，还真是不放心，竟然能在这内院中出现这么多的新生。”萧炎笑着摇了摇头，道。

“你这次来加玛帝国，便是想与你们一起搭上门吧？”阿泰紧紧的盯着萧炎，问道。

“内院里有着一些消

深度之眼
deepshare.net

论文泛读

Strcuture of Paper

论文结构

Structure of Papers

深度之眼
deepshare.net

摘要 Abstruct

介绍背景及提出Transformer-xl模型，能获取长文本的长距离依赖，并且在多个数据集上的效果都表现优异

1 Introduction

Vailla Transformer语言模型在获取长距离依赖的时候存在缺少segemnt之间的信息传递的问题，因此提出本篇论文模型

2 Related Work

在获取长度依赖的问题上，之前很多工作都是在rnn机制上采用memory结构。

3 The Architecture

Transformer-xl网络结构及其内部细节：Segment-Level Recurrence with State Reuse、Relative Positional Encodings

4 Experiments

Transformer-xl的实验结果对比，并且比较了预测时的速度，展示了在文本生成上的效果。

5 Conclusions

Transformer-xl在获取长度依赖的比rnn和Transformer上都有优势，并且在文本生成上也有好的效果。

摘要
abstract

摘要核心

1. Transformer在获取长度依赖上受制于固定长度，本文提出了Transformer-xl模型能解决该问题。

2. Transformer-xl模型提出了片段级递归机制和相对位置编码，并且能够解决片段之间联系丢失的问题。

3. Transformer-xl模型在学习长度依赖的问题上比rnn要长80%、比vallina transformer要长450%，和start-of-the-art的bpc/ppl结果相比，在enwiki8上的结果0.99、在text8上的结果为1.08、在WikiText-103上的结果为18.3、在one-billion上的结果为21.8、在Penn Treebank上的结果为54.5。

论文小标题

Paper title

深度之眼
deepshare.net

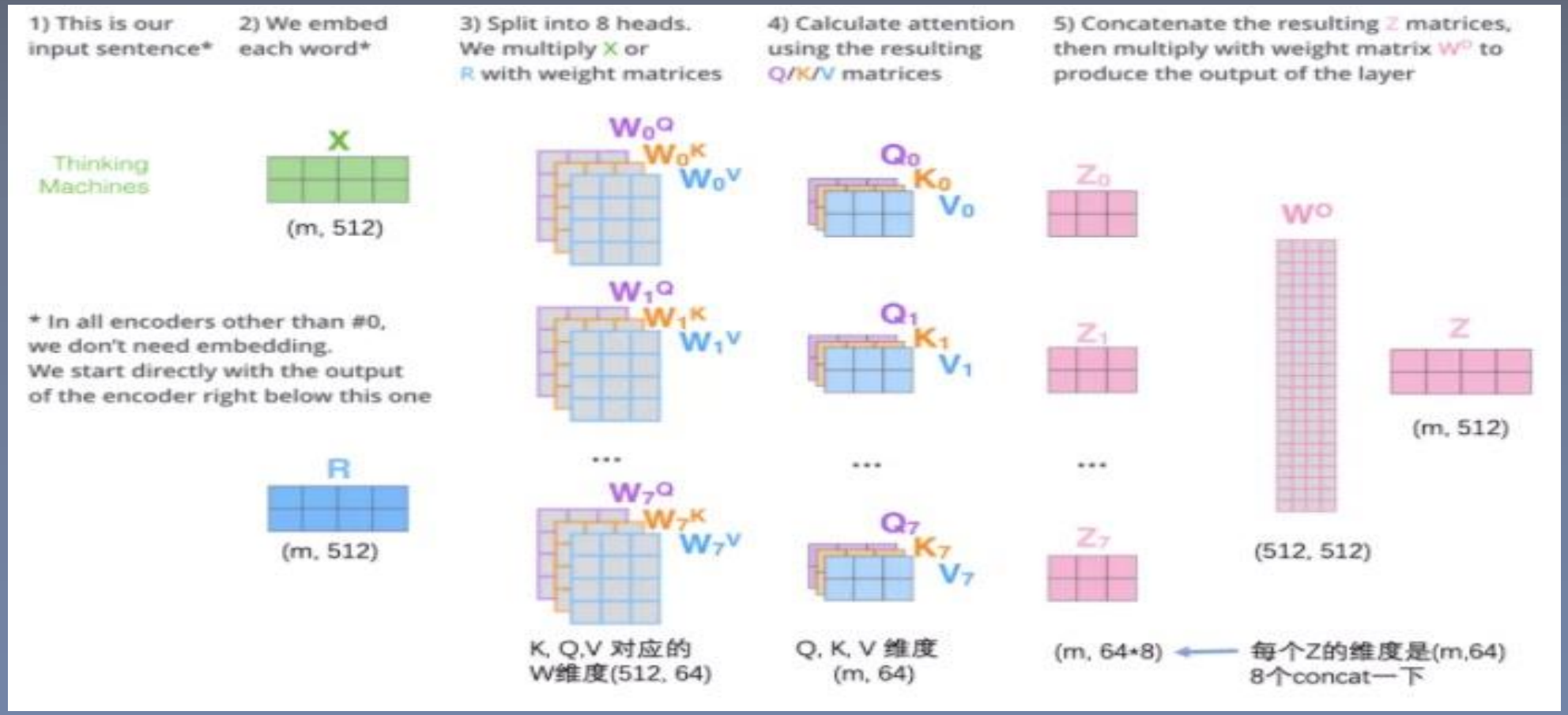
Transformer以及Vallina Transformer的回顾

Strcuture of Paper

论文结构
Structure of Papers

深度之眼
deepshare.net

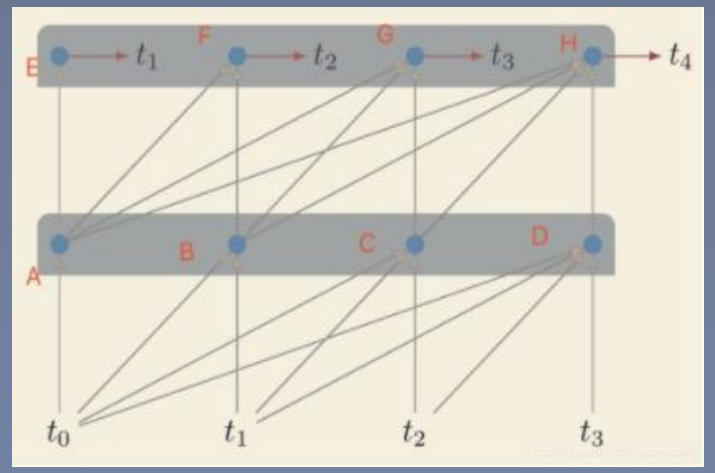
Transformer的多头机制流程



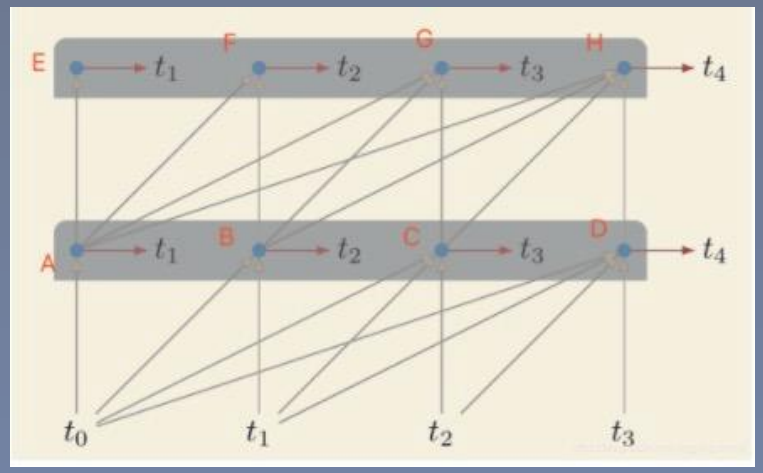
论文结构
Structure of Papers

Vanilla Transformer

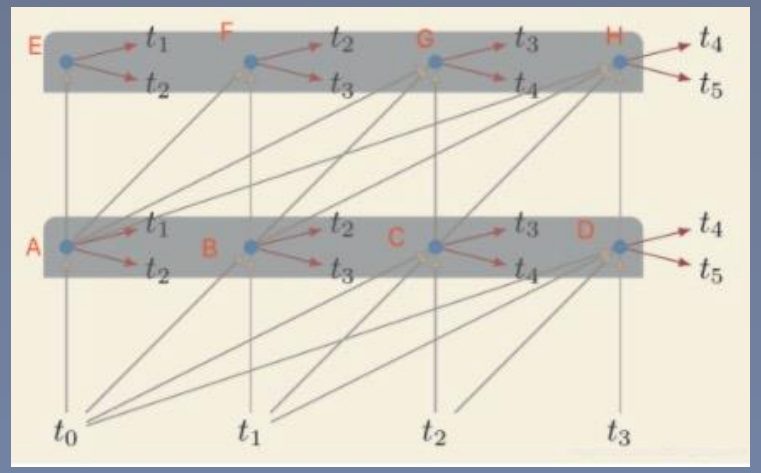
以2层layer来展示,且每个segement的length=4，根据t0-t3的输入预测t4=========>[我，今，天，上] 预测[学]



Multiple Postions



Intermediate Layer Losses



Multiple Targets

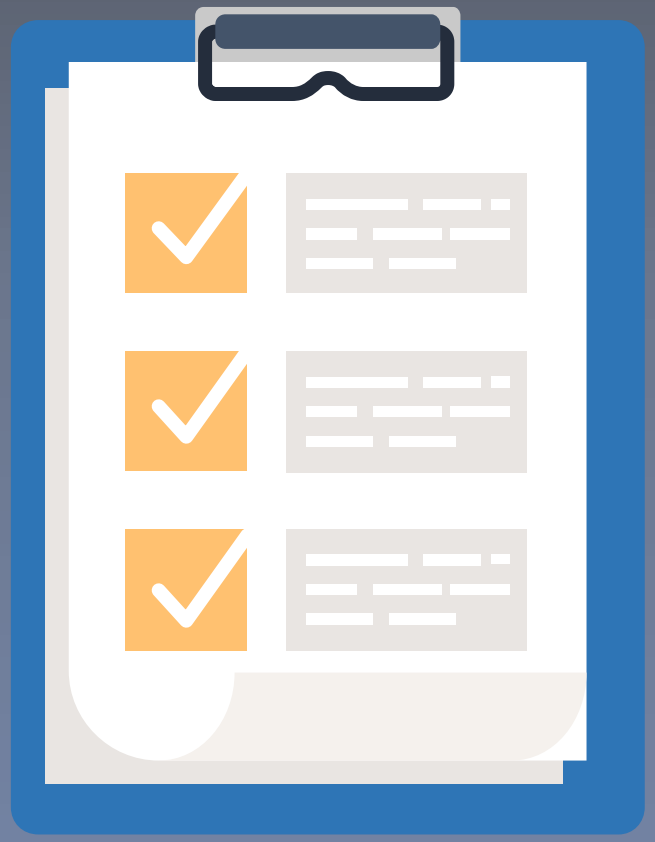
深度之眼
deepshare.net

本课回顾及下节预告

Review in the lesson and Preview of next lesson

本课回顾

Review in the lesson



01 研究背景及成果意义

学习了训练数据以及衡量指标ppl和bpc、了解了论文的实验结果。

02 论文总览

论文总共包含5个部分，论文主要介绍片段级递归机制和相对位置编码。

03 回顾Transformer以及Vanilla Transformer

回顾了self-attention的流程以及学习了Vanilla Transformer的几种loss计算。

下节预告

Preview of next lesson

01 Vanilla Transformer

回顾Vanilla Transformer的结构，分析该模型存在的问题，并提出Transformer-xl模型

02 Segment-Level Recurrence with State Recuse

提出片段级递归机制解决Vallina Model的segment之间联系丢失的问题，提出相对编码机制。

03 实验设置及结果分析

比较了模型在几个数据集上的表现情况，并且展示了模型在文本生成任务上的表现情况。

04 论文总结

总结论文中创新点、关键点及启发点

下节课前准备

Preview of next lesson

- 下载论文
- 泛读论文
- 筛选出自己不懂的部分，带着问题进入下一课时

结 语

循循而进，欲速则不达也。

深度之眼
deepshare.net

联系我们：
电话：18001992849
邮箱：service@deepshare.net
Q Q： 2677693114

公众号

客服微信