4.1 预训练语言模型

林洲汉

上海交通大学

2023年秋季学期

提纲

- 词向量回顾
- 上下文相关词向量早期发展
  - CoVe
  - ELMo
- 典型预训练语言模型及其分类
  - BERT
  - GPT
  - BART
  - 其他
- 预训练语言模型的发展和前景

词向量回顾

- 出发点：
  - 理解自然语言需要对单词进行建模
  - 单热向量维度爆炸且不能反映词间关系

词向量回顾

- 出发点：
  - 理解自然语言需要对单词进行建模
  - 单热向量维度爆炸且不能反映词间关系

- 经典词向量方法：
  - CBOW：邻域词预测中心词
  - Skip-gram：中心词预测邻域词

词向量回顾

- 水果包括苹果、香蕉、橙子、橘子等
  新出的苹果手机没有华为手机好用
- 一朝被蛇咬，十年怕井绳
  面朝大海，春暖花开
- 干一行行一行，一行行行行行

词向量回顾

- 水果包括苹果、香蕉、橙子、橘子等
  新出的苹果手机没有华为手机好用
- 一朝被蛇咬，十年怕井绳
  面朝大海，春暖花开
- 千一行行一行，一行行行行行

- 传统词向量的问题：
  - 静态——上下文无关
  - 使用时需要引入复杂结构来建模上下文信息
  - 对于不同任务有时需要分别设计上下文建模模型

上下文相关词向量

- 预训练向量 ⇒ 预训练上下文建模模型
- 大规模数据预训练复杂模型 + 有限的领域特定数据微调简单模型

- 意义：
  - 极大的开发和揭示了大规模预训练的潜力
  - 极大的简化和统一了上下文信息建模的模型结构

上下文相关词向量的早期发展

Context Vector (CoVe)

- 第一次提出上下文相关词向量
- 受启发于CV中在ImageNet上预训练模型后在其他小数据集上使用

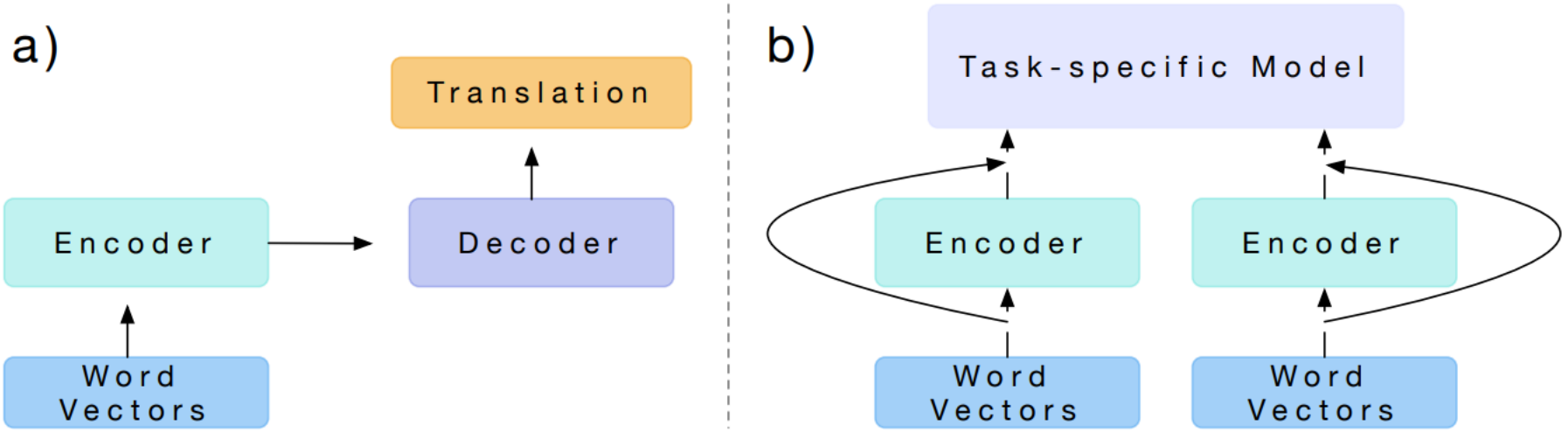


Figure 1: We a) train a two-layer, bidirectional LSTM as the encoder of an attentional sequence-to-sequence model for machine translation and b) use it to provide context for other NLP models.

McCann, Bryan, et al. "Learned in Translation: Contextualized Word Vectors." *NIPS*. 2017.

上下文相关词向量的早期发展

Context Vector (CoVe)

- 第一次提出上下文相关词向量
- 受启发于CV中在ImageNet上预训练模型后在其他小数据集上使用

- 问题：
  - 预训练过程使用的是有监督数据，需要大量额外人工标注

McCann, Bryan, et al. "Learned in Translation: Contextualized Word Vectors." *NIPS*. 2017.

语言模型回顾

基本任务：已知前文预测下一个字

- 输入：单词序列 $w_1, w_2, \cdots, w_{k-1}$
- 输出：对下一个单词预测的概率 $P(w_k|w_1, \cdots, w_{k-1})$

或者说：推测句子的合理程度

- $\prod_{k=1}^{K} P(w_k|w_1, \cdots, w_{k-1}) = P(w_1, w_2, \cdots, w_K)$

神经网络语言模型的训练任务

- 序列到序列建模，整句输入网络，每个词元（Token）的位置预测下一个词元

语言模型回顾

基本任务：已知前文预测下一个字

- 输入：单词序列 $w_1, w_2, \cdots, w_{k-1}$
- 输出：对下一个单词预测的概率 $P(w_k|w_1, \cdots, w_{k-1})$

或者说：推测句子的合理程度

- $\prod_{k=1}^{K} P(w_k|w_1, \cdots, w_{k-1}) = P(w_1, w_2, \cdots, w_K)$

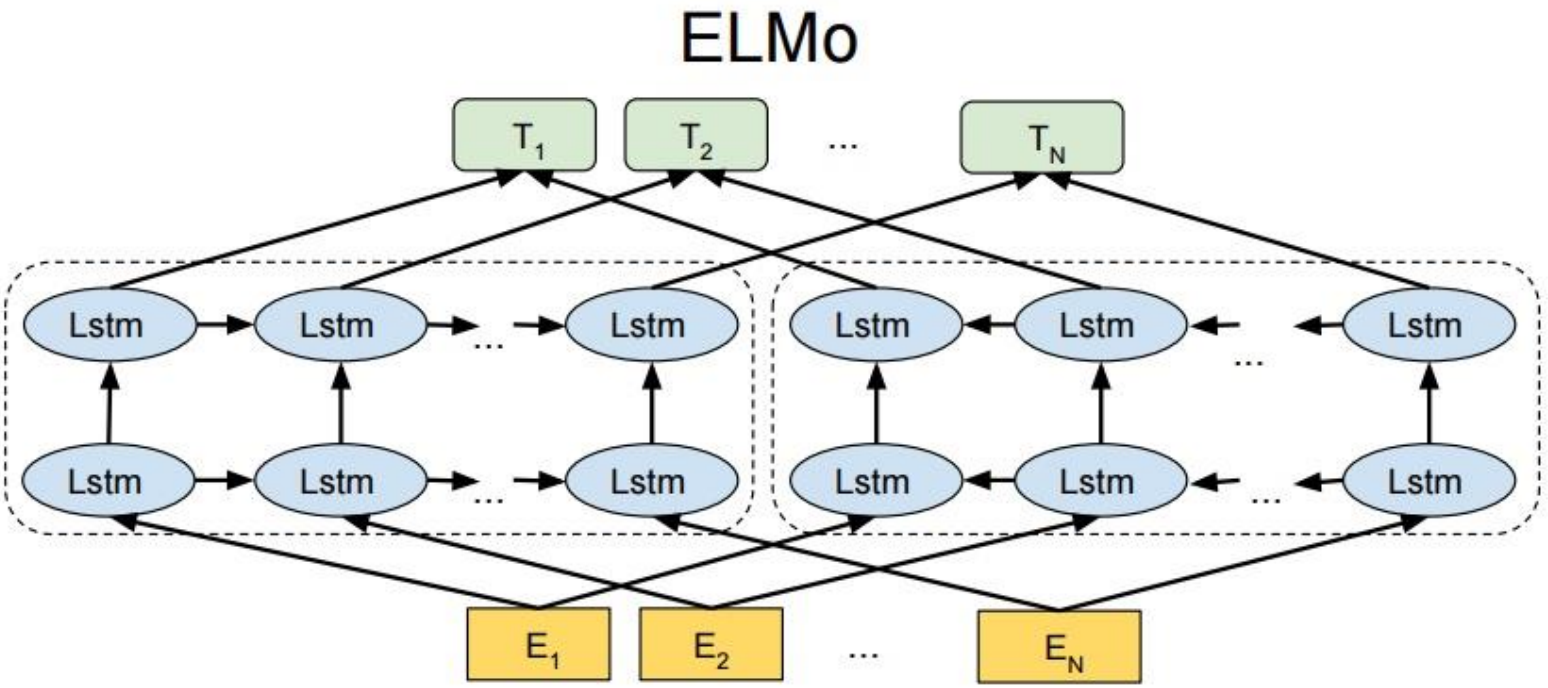
神经网络语言模型的训练任务

- 序列到序列建模，整句输入网络，每个词元（Token）的位置预测下一个词元
- 可以**自我监督**，不需要额外的标注！

上下文相关词向量的早期发展

Embeddings from Language Models (ELMo)

- 利用语言模型的自监督特性，不需要额外标注
- 使用双向语言模型，建模双向的依赖关系



Peters, Matthew E., et al. “Deep contextualized word representations”. *NAACL* 2018.

上下文相关词向量的早期发展

Embeddings from Language Models (ELMo)

- 利用语言模型的自监督特性，不需要额外标注
- 使用双向语言模型，建模双向的依赖关系

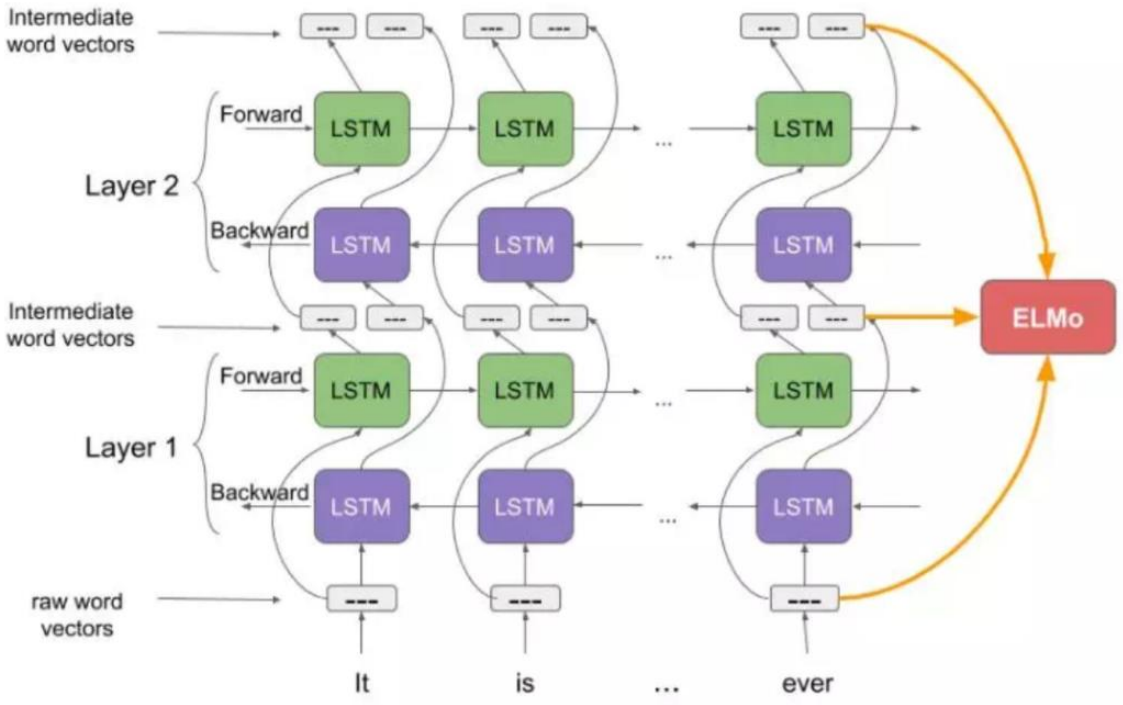
$$\sum_{k=1}^{N} \Big(\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \Big) .$$

Peters, Matthew E., et al. “Deep contextualized word representations”. *NAACL* 2018.

上下文相关词向量的早期发展

Embeddings from Language Models (ELMo)

- 利用语言模型的自监督特性，不需要额外标注
- 使用双向语言模型，建模双向的依赖关系
- 加权平均所有中间层向量得到动态词向量

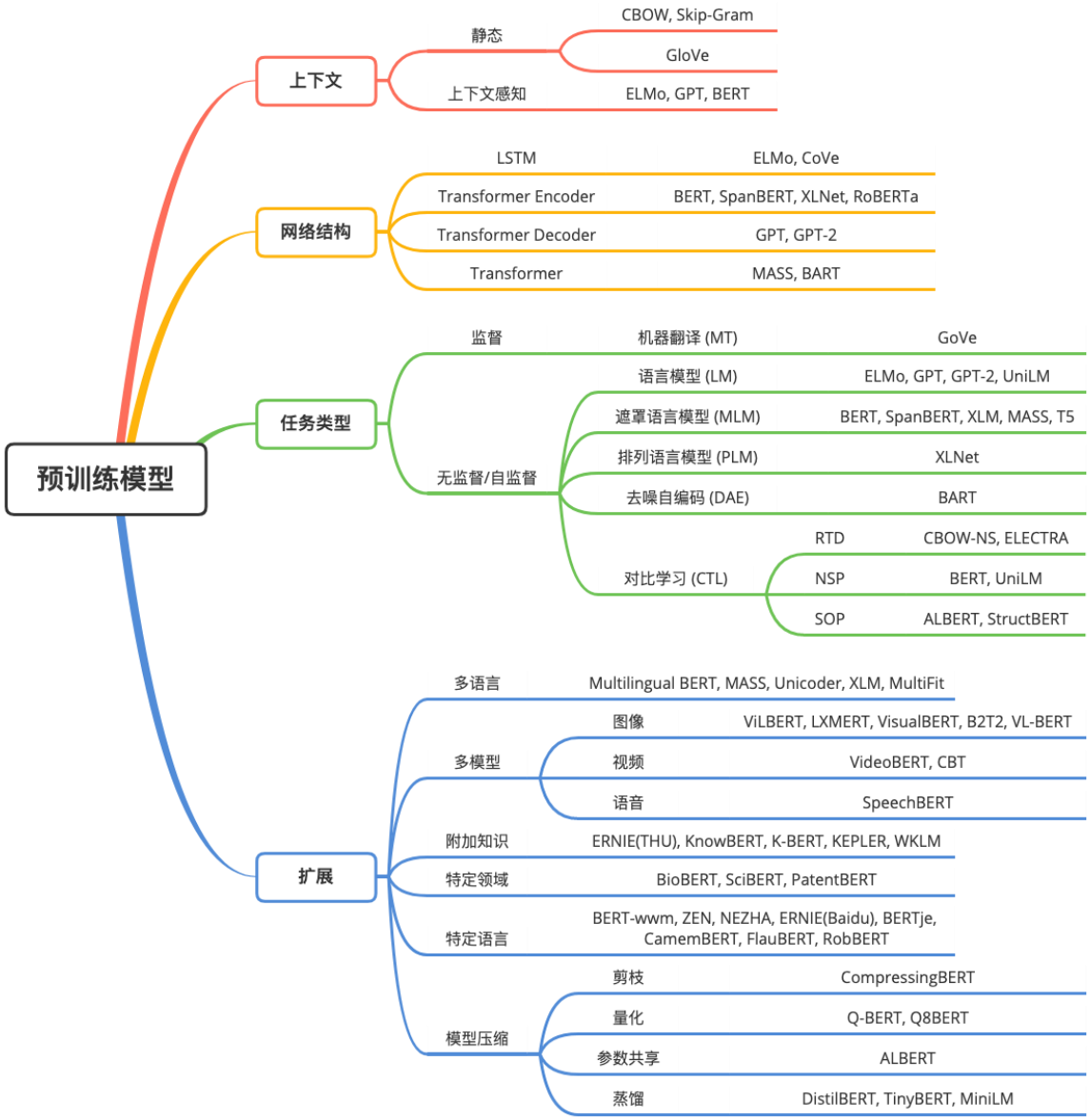


Peters, Matthew E., et al. “Deep contextualized word representations”. *NAACL* 2018.

上下文相关词向量和语言模型

- 上下文相关词向量需要大量的有监督数据
- 语言模型提供的自监督思想不需要人工标注
- 一拍即合！

典型预训练语言模型及其分类



Qiu, Xipeng, et al. "Pre-trained models for natural language processing: A survey." *Science China Technological Sciences* (2020): 1-26.

Outline

- 词向量回顾
- 上下文相关词向量早期发展
  - CoVe
  - ELMo
- 典型预训练语言模型及其分类
  - BERT
  - GPT
  - BART
  - 其他
- 预训练语言模型的发展和前景

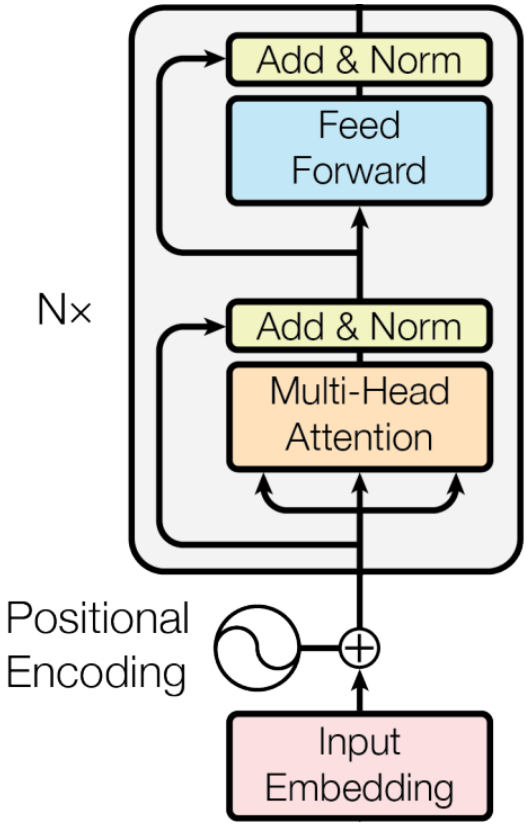
双向自编码模型：BERT

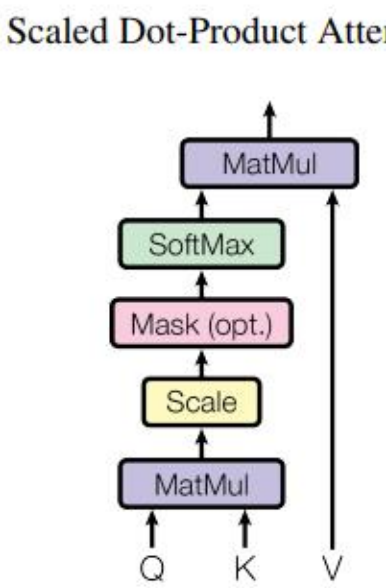
- 一个高效的预训练语言模型的要求：
  - 强大的建模能力
  - 有效建模双向的上下文信息
  - 能够利用大规模无监督数据

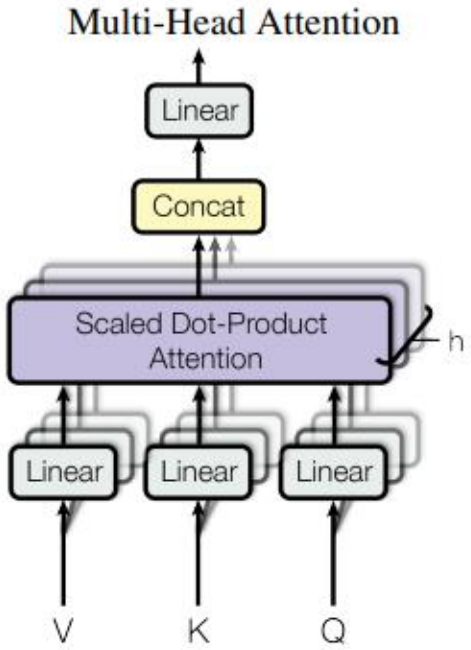
双向自编码模型：BERT

建模能力：LSTM v.s. Transformer

- 使用基于Attention的Transformer结构





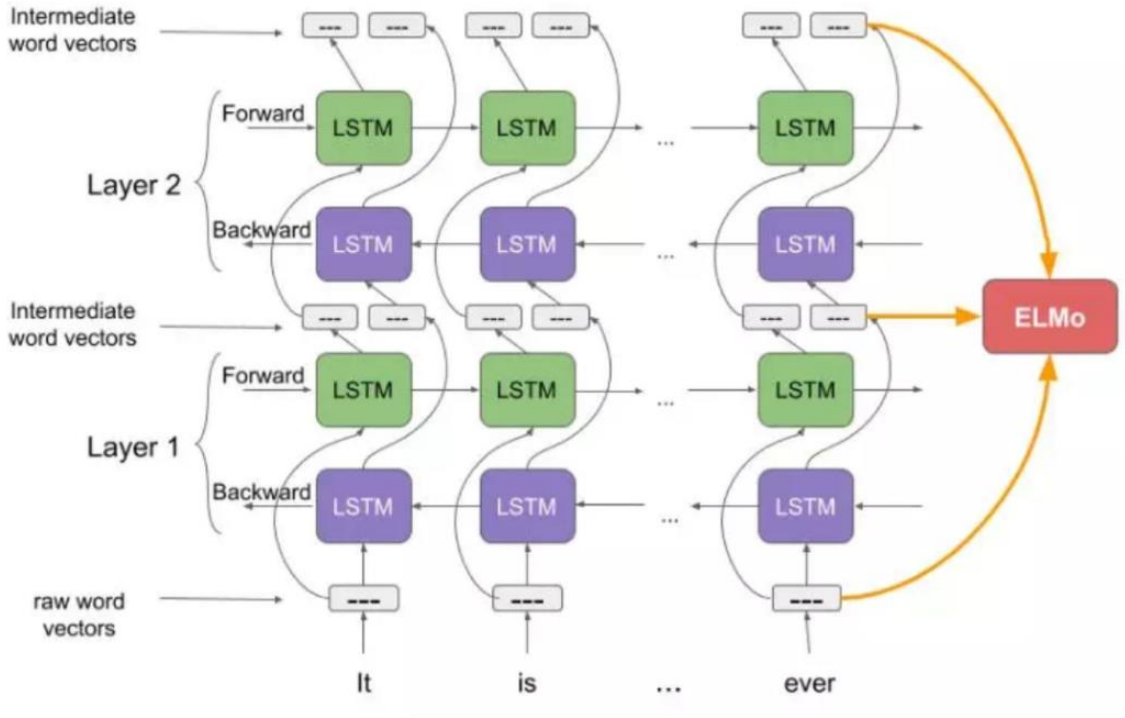


Vaswani, Ashish, et al. "Attention is all you need." Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017.

双向自编码模型：BERT

双向上下文信息：自回归 v.s. 自编码

- 使用基于Attention的Transformer结构
- 自回归模型难以有效建模双向上下文信息

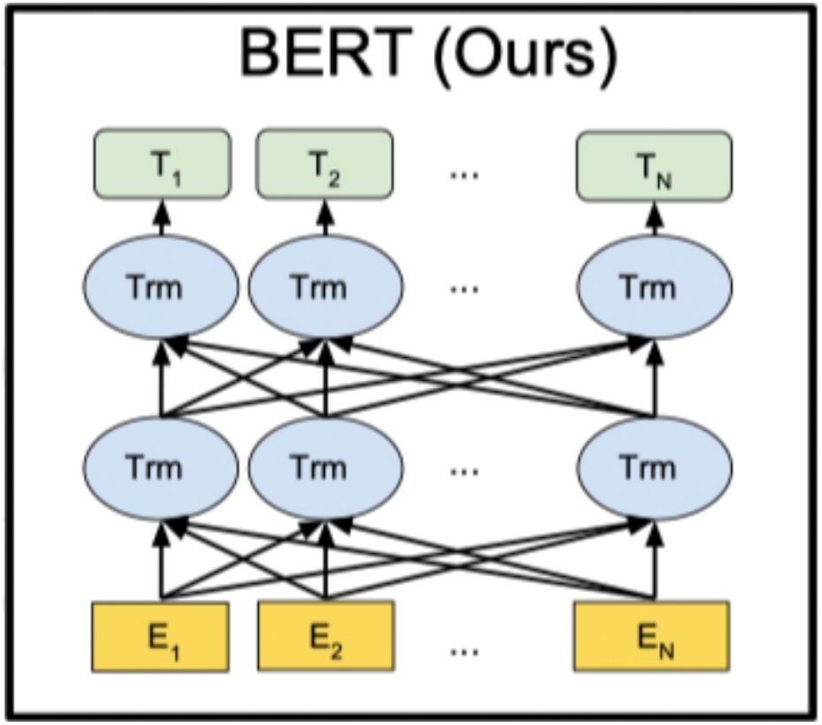


Peters, Matthew E., et al. “Deep contextualized word representations”. *NAACL* 2018.

双向自编码模型：BERT

双向上下文信息：自回归 v.s. 自编码

- 使用基于Attention的Transformer结构
- 仅使用Transformer Encoder结构（即自编码结构）

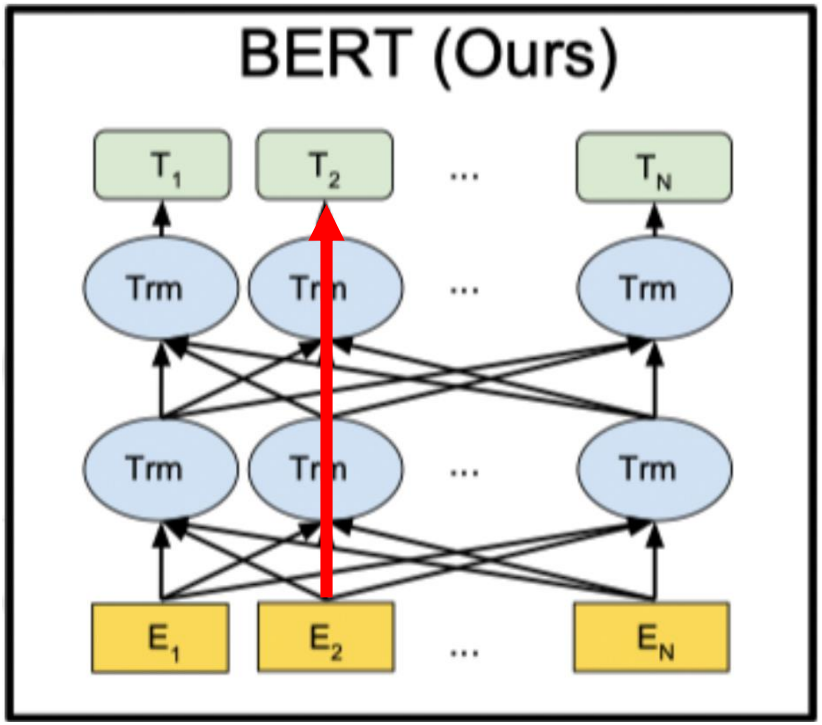


Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

双向自编码模型：BERT

自监督任务：LM v.s. MLM

- 使用基于Attention的Transformer结构
- 仅使用Transformer Encoder结构（即自编码结构）
- 双向建模会导致信息泄露

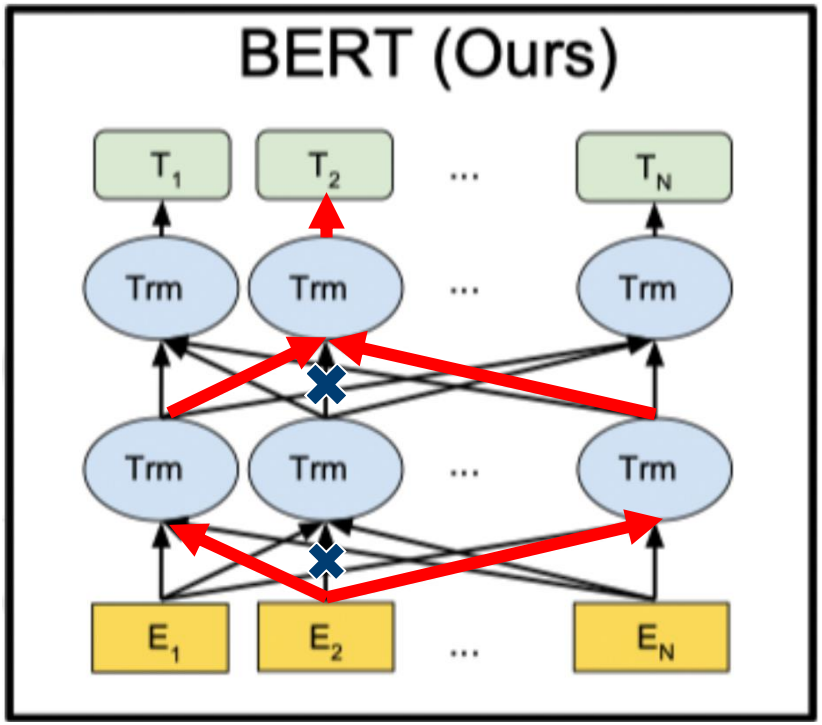


Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

双向自编码模型：BERT

自监督任务：LM v.s. MLM

- 使用基于Attention的Transformer结构
- 仅使用Transformer Encoder结构（即自编码结构）
- 双向建模会导致信息泄露



Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

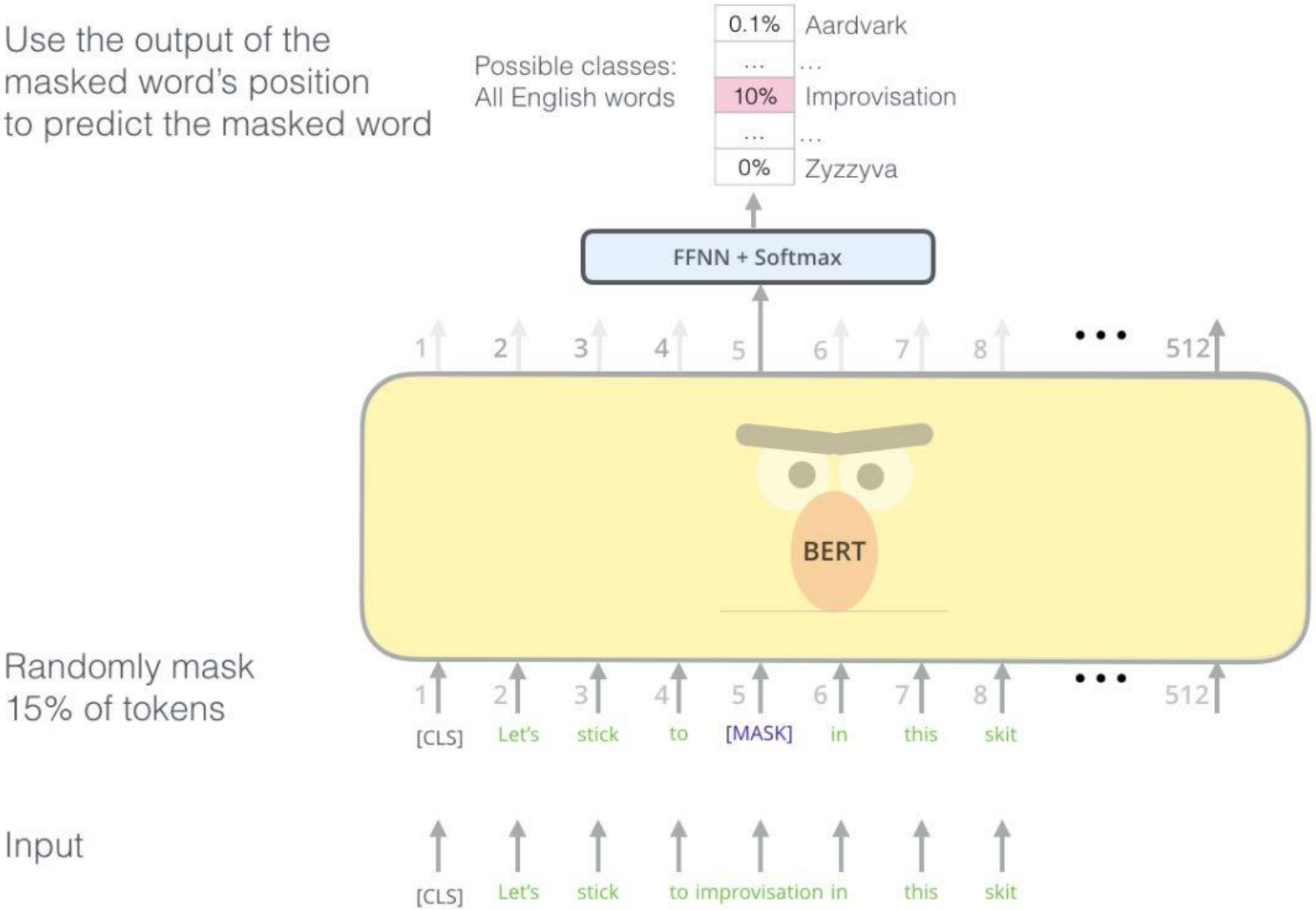
双向自编码模型：BERT

自监督任务：LM v.s. MLM

- 使用基于Attention的Transformer结构
- 仅使用Transformer Encoder结构（即自编码结构）
- 新的预训练任务：掩码语言模型（Masked Language Model, MLM）

双向自编码模型：BERT

自监督任务：LM v.s. MLM



By Jay Alammar's Blog "The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)"

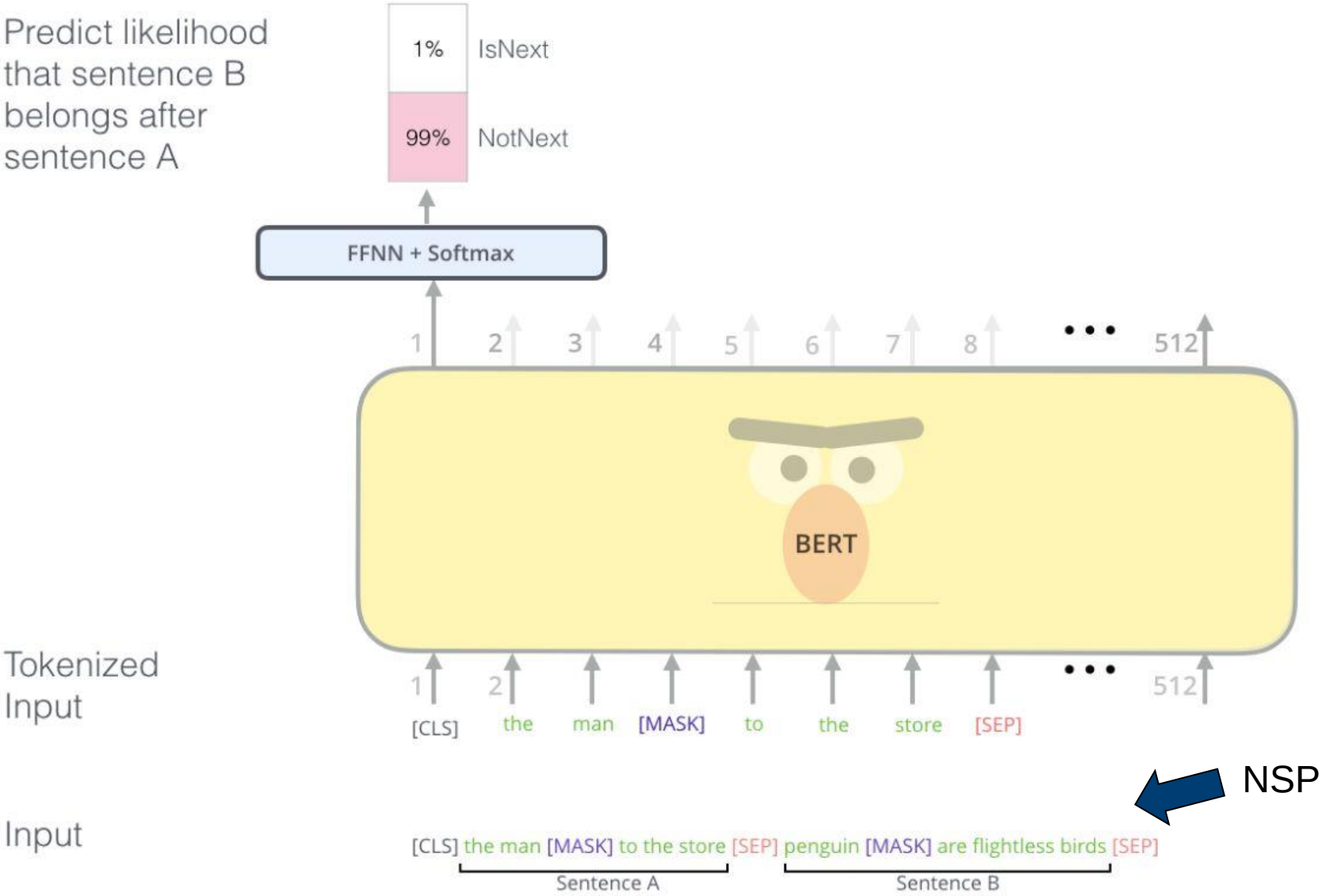
双向自编码模型：BERT

自监督任务：NSP

- 使用基于Attention的Transformer结构
- 仅使用Transformer Encoder结构（即自编码结构）
- 新的预训练任务：掩码语言模型（Masked Language Model, MLM）
- 另外，还引入了NSP(Next Sentence Prediction)任务来引入句间关系的建模

双向自编码模型：BERT

自监督任务：NSP

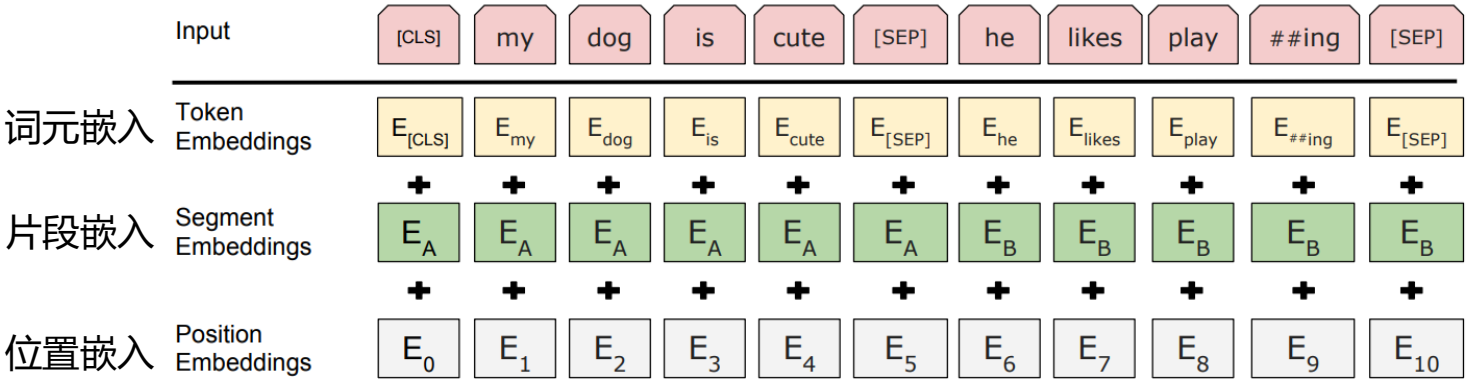


By Jay Alammar's Blog "The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)"

双向自编码模型：BERT

模型设计总结

- 使用Transformer Encoder的自编码模型进行建模
- 输入组成：
  - 词元嵌入 + 片段嵌入 + 位置嵌入
  - [CLS] & [SEP]
  - BPE（Token）



Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

双向自编码模型：BERT

模型设计总结

- 使用Transformer Encoder的自编码模型进行建模
- 输入组成：
  - 词元嵌入 + 片段嵌入 + 位置嵌入
  - [CLS] & [SEP]
  - BPE（Token）
- 预训练任务：
  - MLM
    - 掩蔽15%的词元：10%保留、10%随机替换、80%[MASK]
  - NSP
    - 50%正例+50%负例

双向自编码模型：BERT

参数量及训练量

- $BERT_{BASE}$
  - $L = 12, H = 768, A = 12$
  - 参数总量100M
- $BERT_{LARGE}$
  - $L = 24, H = 1024, A = 16$
  - 参数总量340M

- 在总词数3.3B的数据集上训练了40轮
- $BERT_{BASE}$使用16张TPU，$BERT_{LARGE}$使用64张TPU
- 总训练时间为4天

Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

双向自编码模型：BERT

预训练微调使用范式

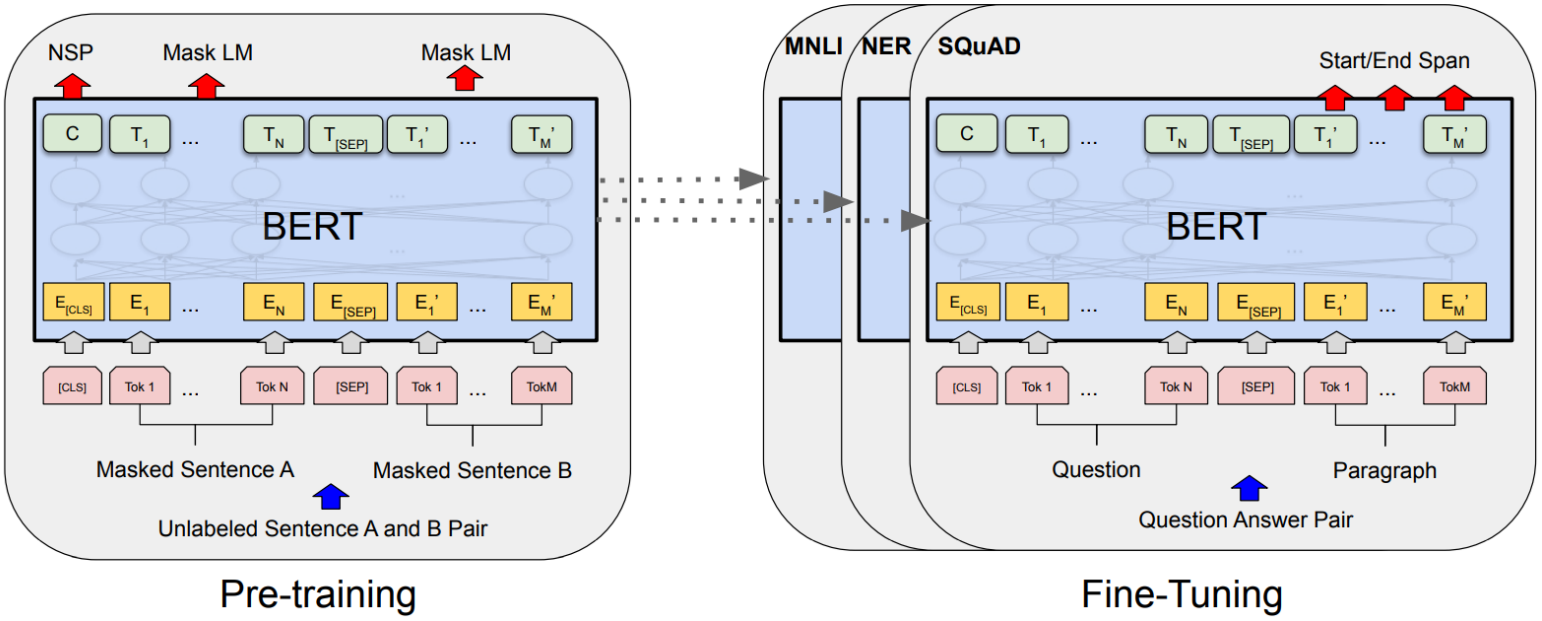


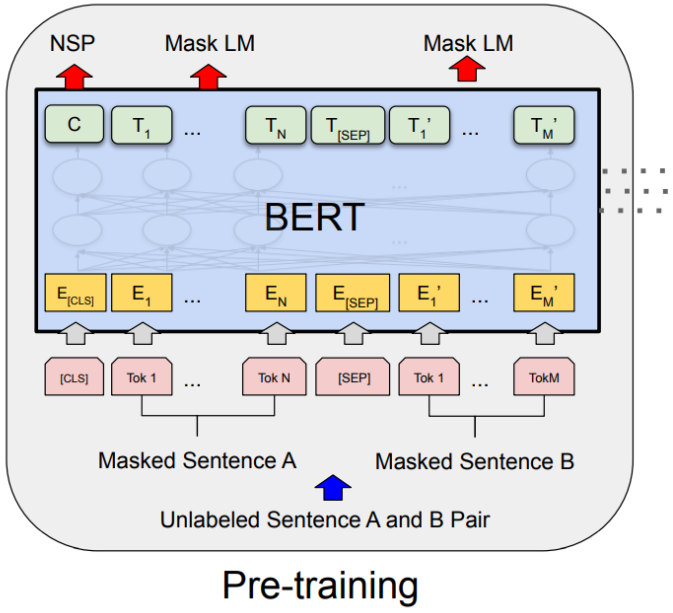
Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

双向自编码模型：BERT

预训练微调使用范式

- 词向量在哪？



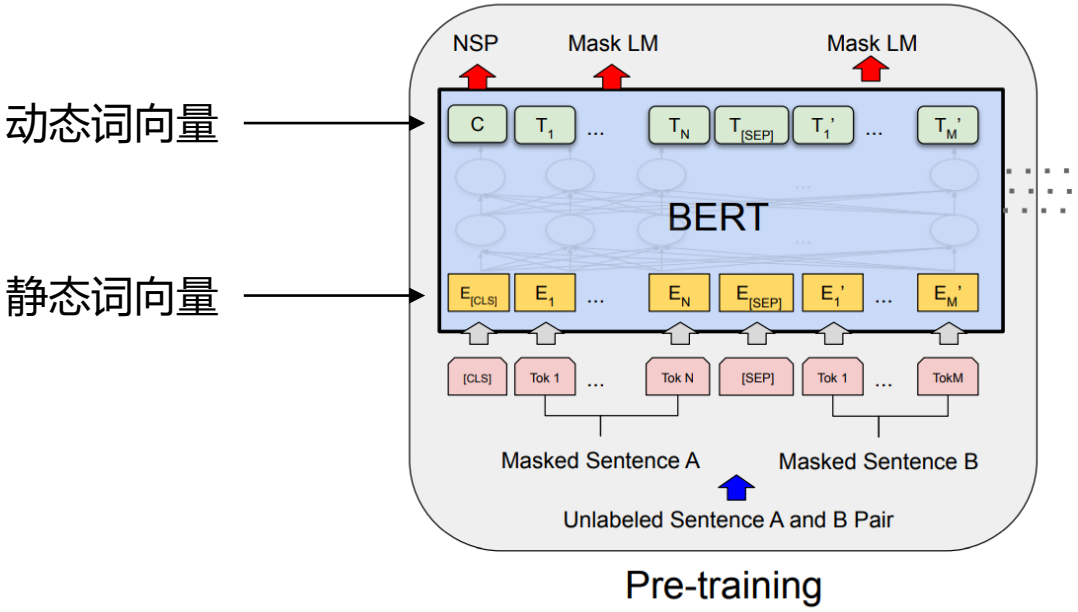
Pre-training

Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

双向自编码模型：BERT

预训练微调使用范式

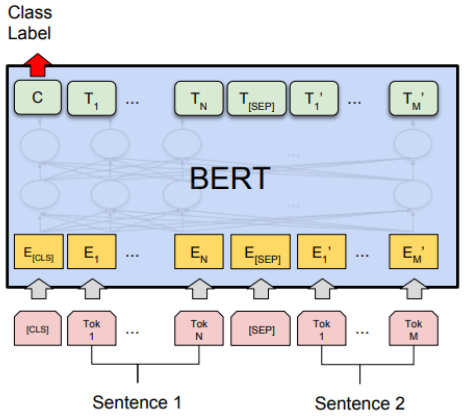
- 词向量在哪？
  - 静态词向量：输入中的Token Embedding
  - 动态词向量：线性分类层前的隐向量
- 下游任务可以直接改变线性分类层结构即可



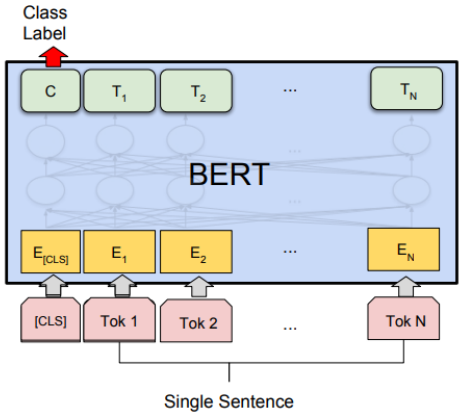
Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

双向自编码模型：BERT

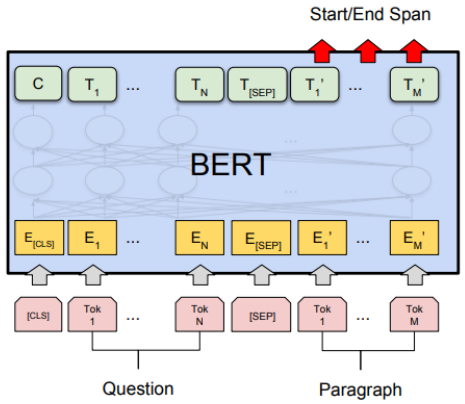
在下游任务上的微调：示例



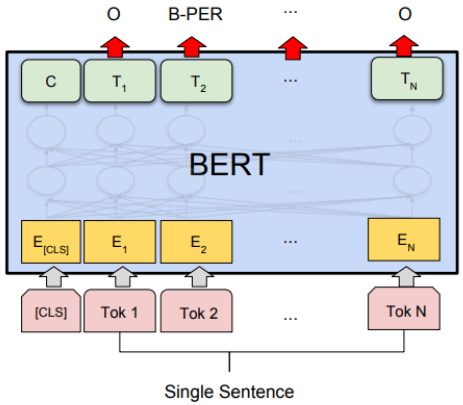
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



(c) Question Answering Tasks: SQuAD v1.1



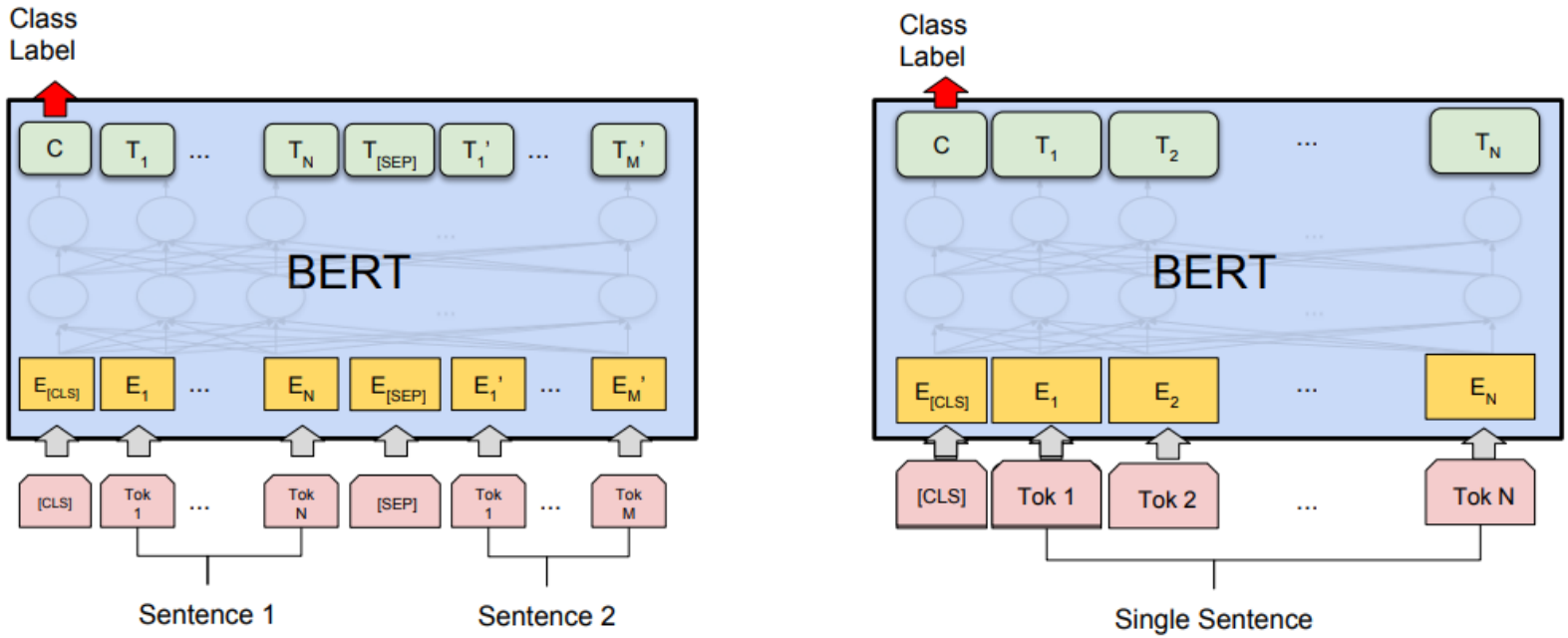
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

双向自编码模型：BERT

在下游任务上的微调：示例

- 只需要将最后的线性分类层换成相应的维度即可



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

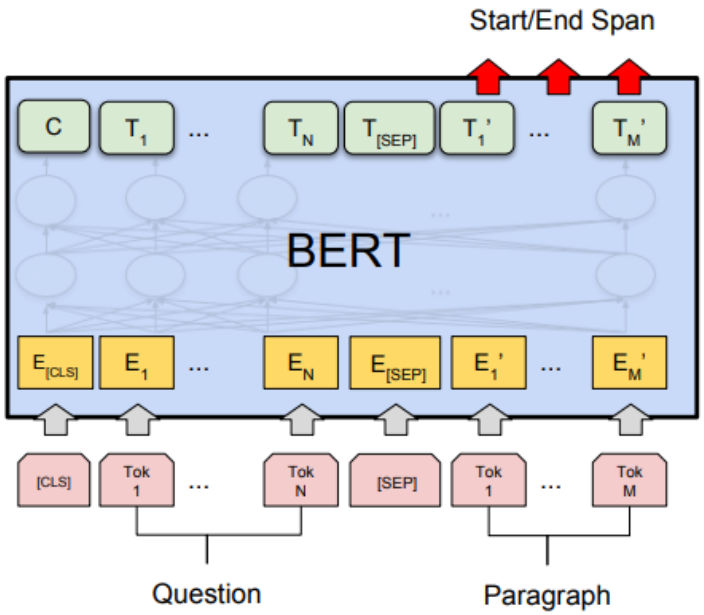
(b) Single Sentence Classification Tasks: SST-2, CoLA

Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

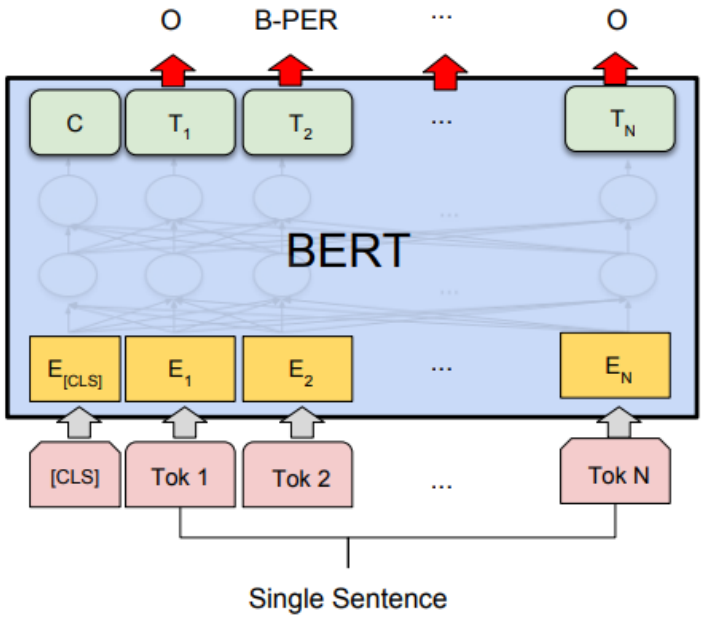
双向自编码模型：BERT

在下游任务上的微调：示例

- 只需要将最后的线性分类层换成相应的维度即可



(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

双向自编码模型：BERT

在下游任务上的微调：GLUE结果

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

双向自编码模型：BERT

在下游任务上的微调：SQuAD结果

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

Table 3: SQuAD 2.0 results. We exclude entries that use BERT as one of their components.

Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

双向自编码模型：BERT

- 优缺点：
  - 在自然语言理解任务上表现突出
  - 难以解决自然语言生成问题
  - [MASK]
    - 预训练和微调阶段的输入空间分布不匹配
    - 预训练过程收敛较慢，不能掩蔽过多词元
    - 假设被掩蔽词元是独立的，只能进行联合概率的有偏估计

Jacob Devlin

双向自编码模型：BERT

- 优缺点：
  - 在自然语言理解任务上表现突出
  - 难以解决自然语言生成问题
  - [MASK]
    - 预训练和微调阶段的输入空间分布不匹配
    - 预训练过程收敛较慢，不能掩蔽过多词元
    - 假设被掩蔽词元是独立的，只能进行联合概率的有偏估计

- 直观理解独立性假设的问题：词组

  New York is a city.

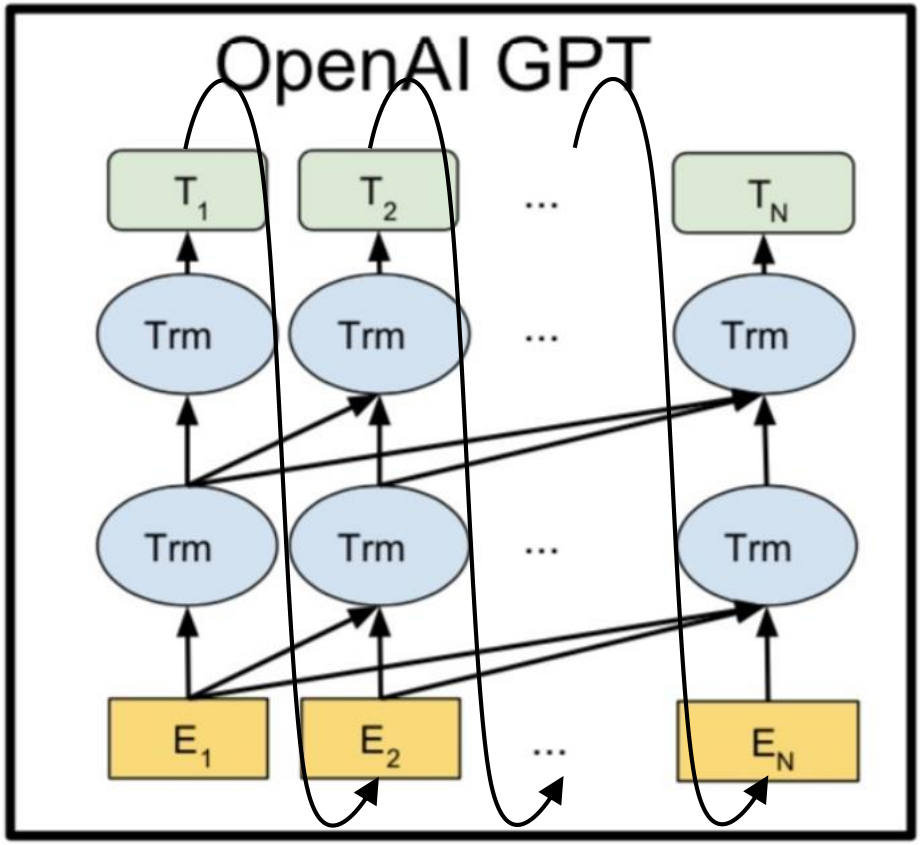
  - 当New和York同时被掩蔽时

$$p(\text{New}, \text{York}) = p(\text{New}) \times p(\text{York}|\text{New}) \neq p(\text{New}) \times p(\text{York})$$

单向自回归模型：GPT

结构

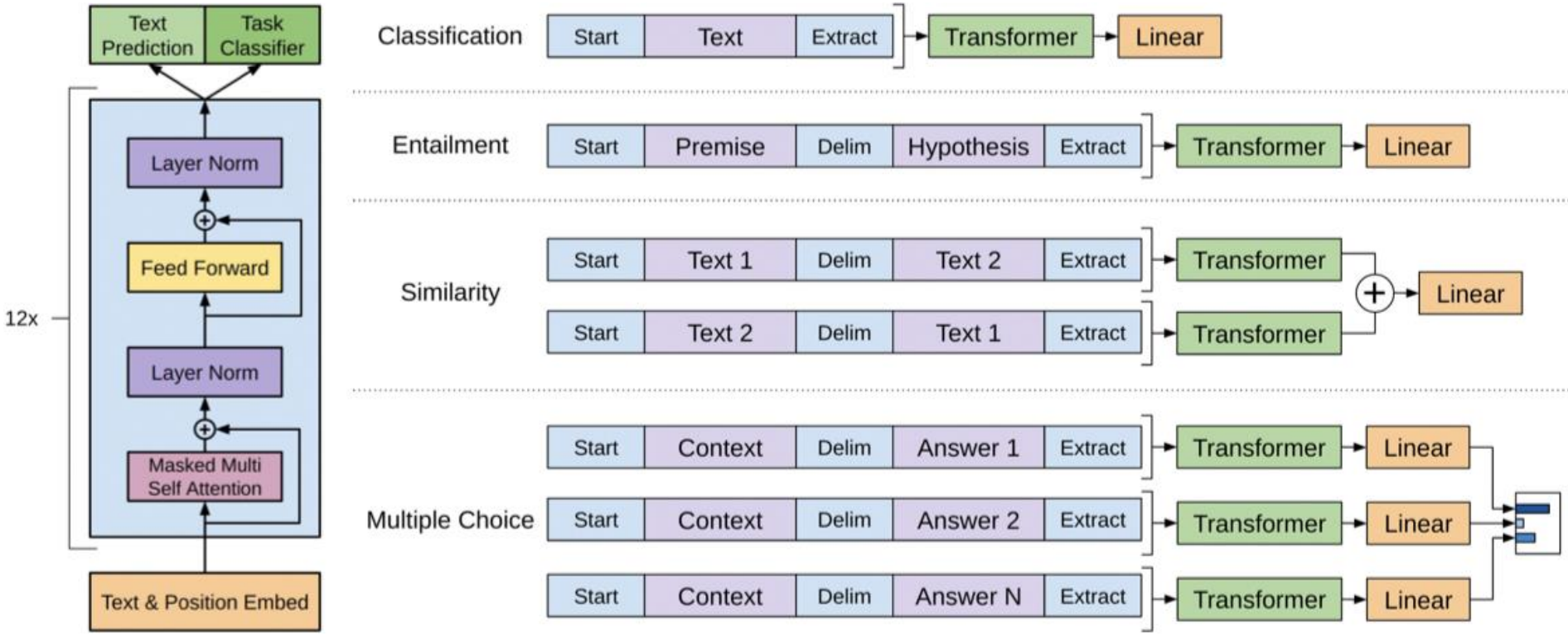
- 直接使用语言模型任务进行预训练
- 采用类Transformer Decoder结构（即自回归结构）



Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

单向自回归模型：GPT

下游任务使用方法



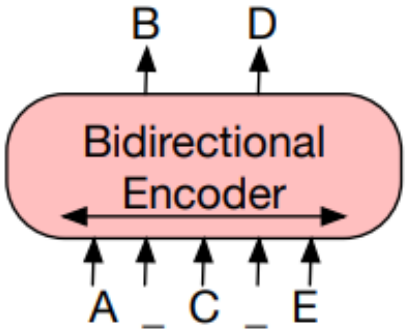
Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

单向自回归模型：GPT

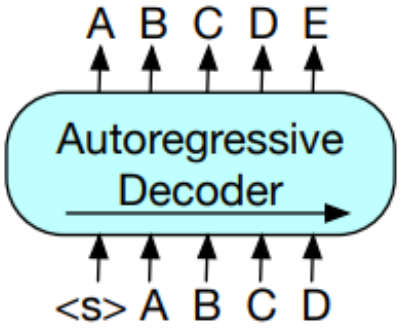
- 直接采用LM任务预训练，不存在[MASK]引入的偏差
- 在自然语言生成的任务上表现突出
- 由于LM任务的限制，只能建模上文信息，不能得到严格意义上的上下文相关词向量
- 不适合自然语言理解相关任务

编解码模型：BART

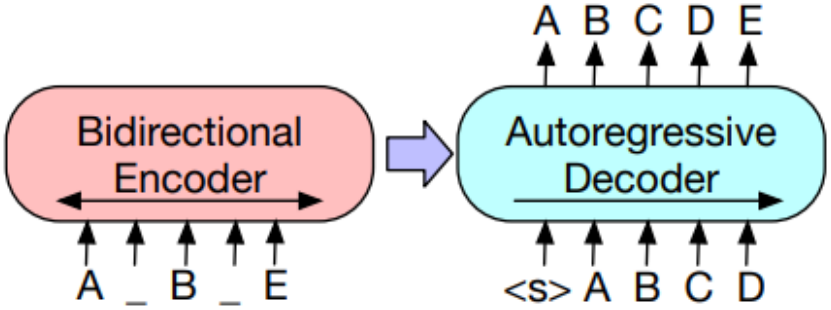
- 采用完整的Transformer结构
- 采用MLM来解决Encoder端信息泄露问题



BERT



GPT

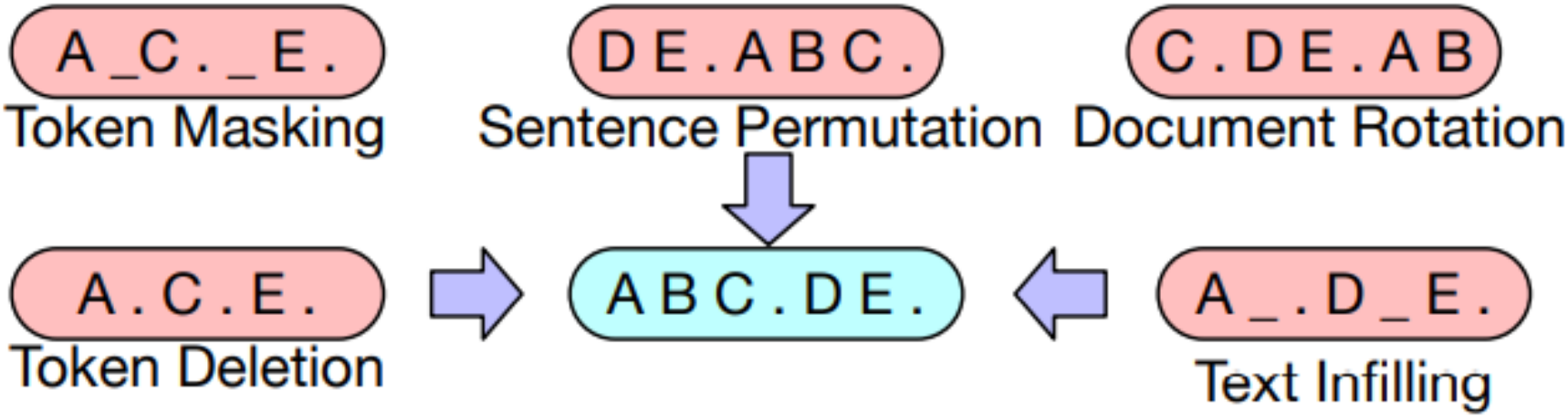


BART

Lewis, Mike, et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.

编解码模型：BART

- 采用完整的Transformer结构
- 采用MLM来解决Encoder端信息泄露问题
- 引入多种噪音来加强模型泛化性能



Lewis, Mike, et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.

其他典型结构

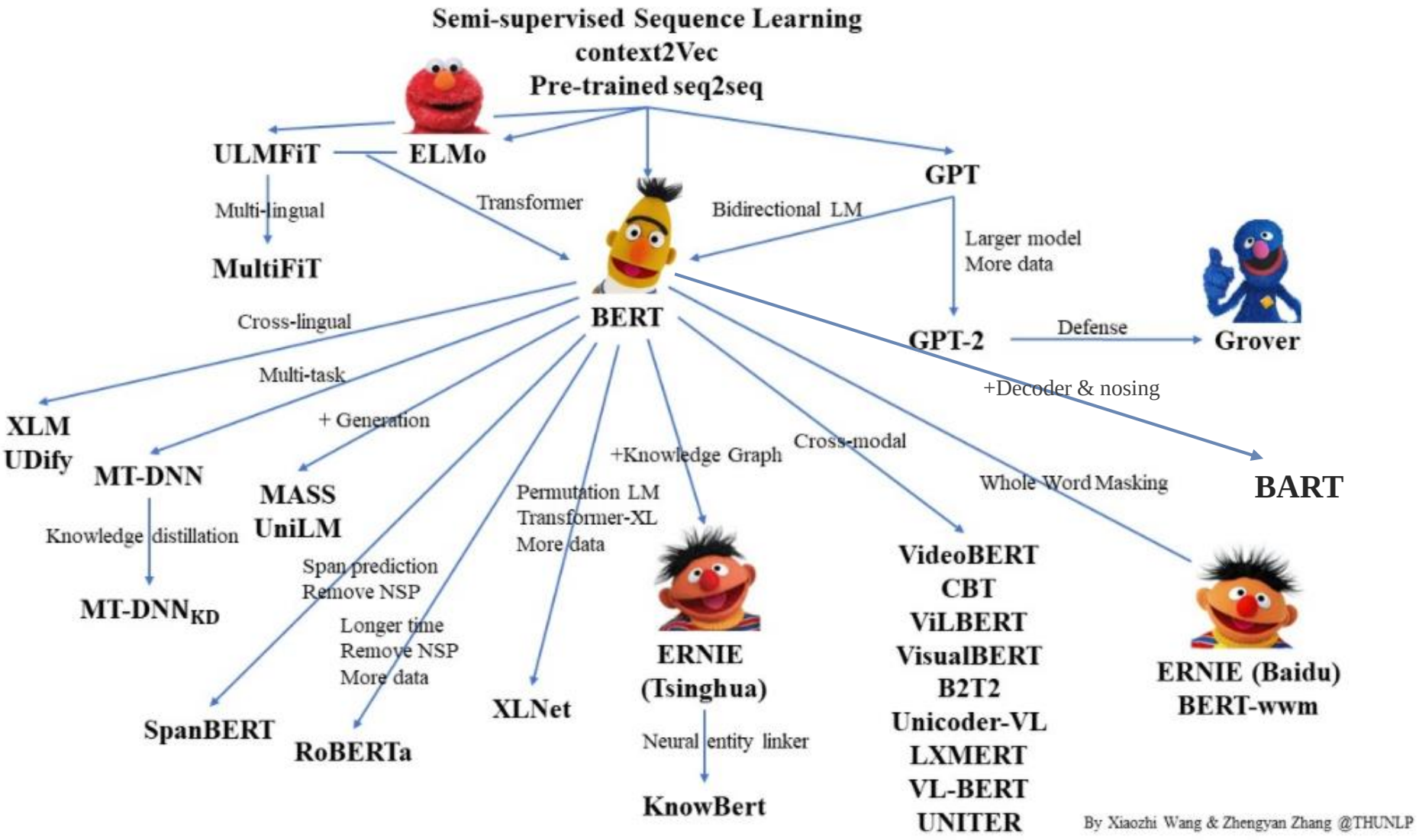
- XLNet
  - Permutated LM

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. "Xlnet: Generalized autoregressive pretraining for language understanding." *Advances in neural information processing systems* 32 (2019).

- UniLM
  - Prefix LM

Dong, Li, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. "Unified language model pre-training for natural language understanding and generation." *Advances in Neural Information Processing Systems* 32 (2019).

典型预训练语言模型总结

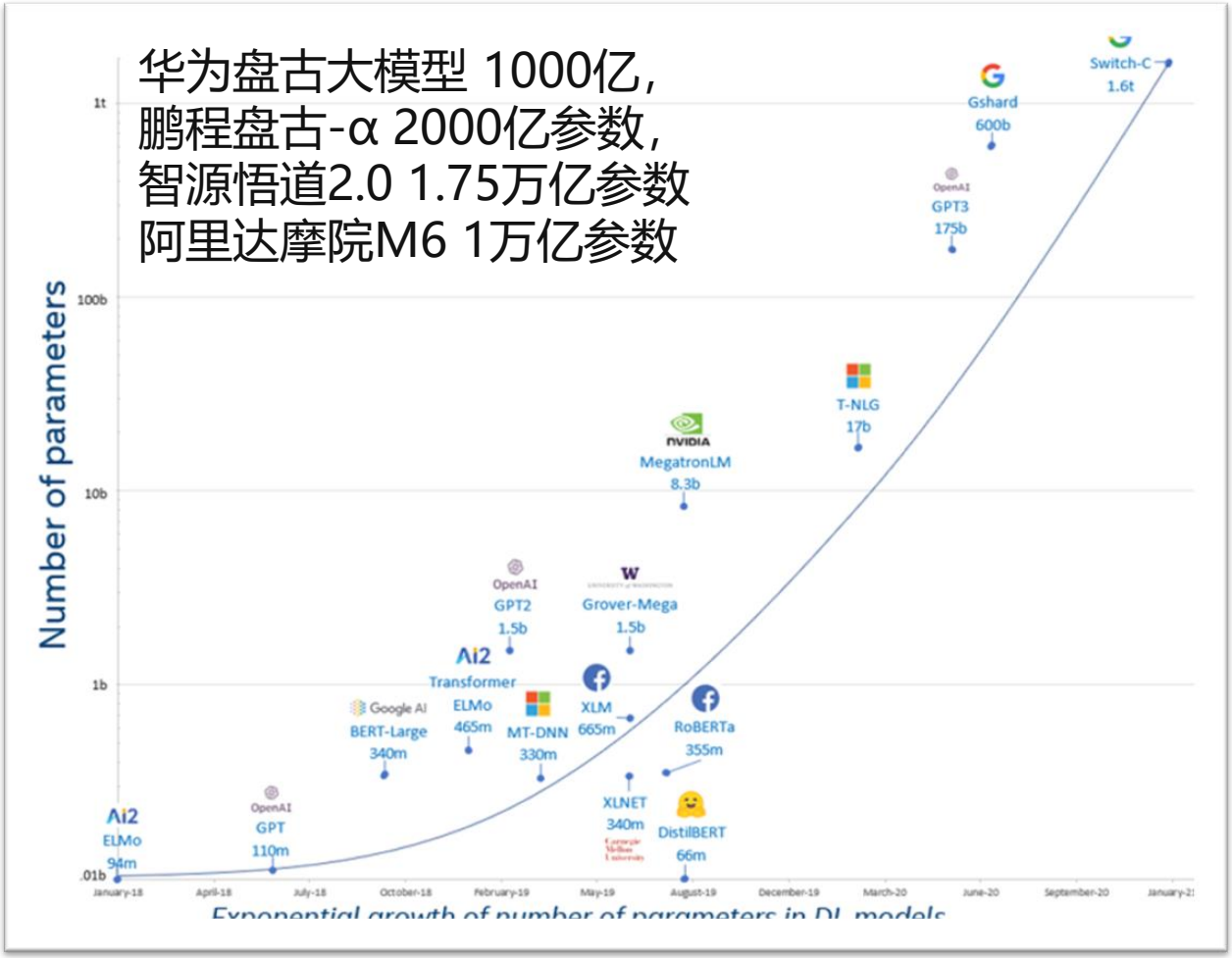


预训练语言模型的发展

- 模型自身结构的改进
- 优化预训练任务
- 预训练语料来源及其组成
- 微调方法

预训练语言模型的发展

- 不断扩大的参数量与计算量



From Daxin Jiang's PPT

- ALBERT
  - 核心思想
    - 嵌入分解：将嵌入层分解为两个更小的矩阵的乘积
    - 参数共享：将所有层分成多个组，每组的参数完全共享
  - 极大的减少了参数量，但是计算量同时增加

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	0.3x

Table 2: Dev set results for models pretrained over BOOKCORPUS and Wikipedia for 125k steps. Here and everywhere else, the Avg column is computed by averaging the scores of the downstream tasks to its left (the two numbers of F1 and EM for each SQuAD are first averaged).

Lan, Zhenzhong, et al. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations." International Conference on Learning Representations. 2019.

预训练语言模型的发展

模型自身结构的改进：模型的压缩与简化

- 知识蒸馏(+剪枝)：DistillBERT、TinyBERT等
- 量化：Q-BERT、Q8BERT等

Table 3: Comparison of Compressed PTMs

Method	Type	#Layer	Loss Function*	Speed Up	Params	Source PTM	GLUE‡
$\text{BERT}_{\text{BASE}}$ [36]	Baseline	12	$\mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{NSP}}$		110M		79.6
$\text{BERT}_{\text{LARGE}}$ [36]		24	$\mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{NSP}}$		340M		81.9
Q-BERT [156]	Quantization	12	HAWQ + GWQ	-		$\text{BERT}_{\text{BASE}}$	≈ 99% BERT◇
Q8BERT [211]		12	DQ + QAT	-		$\text{BERT}_{\text{BASE}}$	≈ 99% BERT
ALBERT§ [93]	Param. Sharing	12	$\mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{SOP}}$	×5.6 ~ 0.3	12 ~ 235M		89.4 (ensemble)
DistilBERT [152]	Distillation	6	$\mathcal{L}_{\text{KD-CE}}$+$\text{Cos}_{\text{KD}}$+ $\mathcal{L}_{\text{MLM}}$	×1.63	66M	$\text{BERT}_{\text{BASE}}$	77.0 (dev)
TinyBERT§ † [75]		4	$\text{MSE}_{\text{embed}}$+$\text{MSE}_{\text{attn}}$+ MSE_{hidn}+$\mathcal{L}_{\text{KD-CE}}$	×9.4	14.5M	$\text{BERT}_{\text{BASE}}$	76.5
BERT-PKD [169]		3 ~ 6	$\mathcal{L}_{\text{KD-CE}}$+$\text{PT}_{\text{KD}}$+ $\mathcal{L}_{\text{Task}}$	×3.73 ~ 1.64	45.7 ~ 67 M	$\text{BERT}_{\text{BASE}}$	76.0 ~ 80.6♯
PD [183]		6	$\mathcal{L}_{\text{KD-CE}}$+$\mathcal{L}_{\text{Task}}$+ $\mathcal{L}_{\text{MLM}}$	×2.0	67.5M	$\text{BERT}_{\text{BASE}}$	81.2♯
MobileBERT§[172]		24	FMT+AT+PKT+ $\mathcal{L}_{\text{KD-CE}}$+$\mathcal{L}_{\text{MLM}}$	×4.0	25.3M	$\text{BERT}_{\text{LARGE}}$	79.7
MiniLM [194]		6	AT+AR	×1.99	66M	$\text{BERT}_{\text{BASE}}$	81.0♭
DualTrain§ †[216]		12	Dual Projection+$\mathcal{L}_{\text{MLM}}$	-	1.8 ~ 19.2M	$\text{BERT}_{\text{BASE}}$	75.8 ~ 81.9♮
BERT-of-Theseus [203]	Module Replacing	6	$\mathcal{L}_{\text{Task}}$	×1.94	66M	$\text{BERT}_{\text{BASE}}$	78.6

Qiu, Xipeng, et al. "Pre-trained models for natural language processing: A survey." Science China Technological Sciences (2020): 1-26.

预训练语言模型的发展

模型自身结构的改进：处理长文本问题

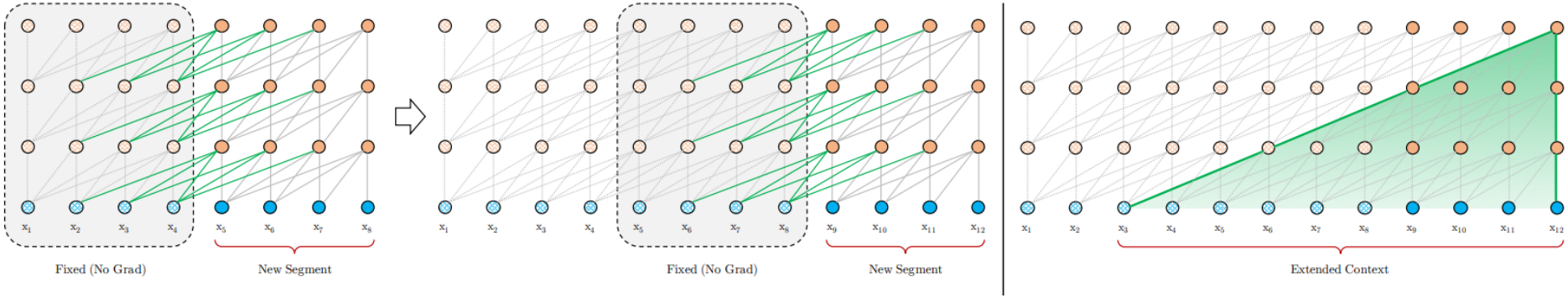
- 由于显存限制和训练效率要求，预训练阶段输入句子的长度不能过长；
- 由于Transformer对于顺序的不敏感，需要特别通过Position Embedding来引入位置信息。

- 使用时输入长度受限于预训练文本长度（一般为512）

预训练语言模型的发展

模型自身结构的改进：处理长文本问题

- Transformer-XL
  - Segment-level Recurrent

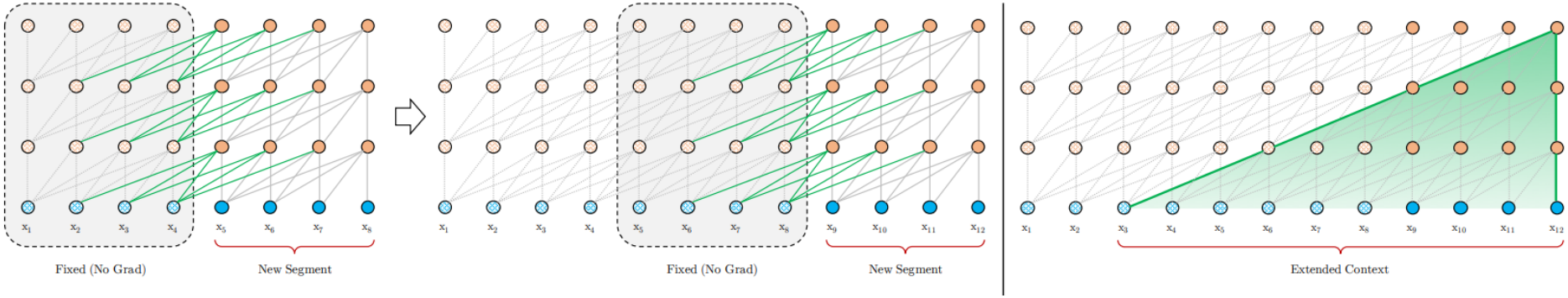


Dai, Zihang, et al. "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019.

预训练语言模型的发展

模型自身结构的改进：处理长文本问题

- Transformer-XL
  - Segment-level Recurrent
    - 将上一个Segment计算时的隐状态缓存下来
    - 计算后一个Segment时使用缓存隐状态来增大上下文范围
    - 缓存隐状态不参与梯度反传



Dai, Zihang, et al. "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019.

预训练语言模型的发展

预训练语料来源及其组成

- 多语言
- 多模态
- 领域特定与知识增强

Qiu, Xipeng, et al. "Pre-trained models for natural language processing: A survey." Science China Technological Sciences (2020): 1-26.

预训练语言模型的发展

预训练语料来源及其组成：多语言

- 核心问题：
  - 使用统一的模型对多种语言建模——多语言理解
  - 多语言的翻译和对齐——多语言生成

预训练语言模型的发展

预训练语料来源及其组成：多语言

- 核心问题：
  - 使用统一的模型对多种语言建模——多语言理解
  - 多语言的翻译和对齐——多语言生成

- 多语言理解：
  - mBERT——简单的使用多种语言语料进行预训练
  - XLM——在mBERT基础上，引入了机器翻译这个监督任务辅助
  - Unicoder——在XLM基础上，引入不同粒度的对齐任务（跨语言词语恢复，跨语言句义匹配判别，跨语言MLM），仍需要监督数据

Qiu, Xipeng, et al. "Pre-trained models for natural language processing: A survey." Science China Technological Sciences (2020): 1-26.

预训练语言模型的发展

预训练语料来源及其组成：多语言

- 核心问题：
  - 使用统一的模型对多种语言建模——多语言理解
  - 多语言的翻译和对齐——多语言生成

- 多语言理解：
  - mBERT、XLM、Unicoder等
- 多语言生成：
  - MASS——模型结构类似BART，使用多种语言的单语言语料进行训练
  - XNLG——2×2，两阶段预训练（encoder-MLM，decoder-DAE），单语言和跨语言两种语料

Qiu, Xipeng, et al. "Pre-trained models for natural language processing: A survey." Science China Technological Sciences (2020): 1-26.

预训练语言模型的发展

预训练语料来源及其组成：多模态

- 图像+文本：按照两个模态的交互位置分类
  - 双流（分别编码后交互再编码）
    - ViBERT(MMM, MMA), LXMBERT(MLM, RoIFR, DLC, CMM, IQA)等
  - 单流（直接统一进行编码）
    - VisualBERT(MLM, SIP), VL-BERT(MLM, MRoIC)等
- 视频+文本
  - VideoBERT(MLM, MFM), UniVL(VTJ, CMLM, CMFM, VTA, LR)等
- 音频+文本
  - SpeechBERT(After Initial Phonetic-Semantic Joint Embedding, MLM & MAL)等

Qiu, Xipeng, et al. "Pre-trained models for natural language processing: A survey." Science China Technological Sciences (2020): 1-26.

预训练语言模型的发展

微调方法

- 提升在特定下游任务上的性能
  - 两阶段微调（Two-Stage）
    - 先在其他相同领域的数据上预微调
  - 多任务学习（Multi Task）
    - 多个下游任务同时学习
  - 解冻方法（Unfreezing）
    - 由少到多的微调参数，例如逐层解冻

Qiu, Xipeng, et al. "Pre-trained models for natural language processing: A survey." Science China Technological Sciences (2020): 1-26.

预训练语言模型的发展

微调方法

- 提升在特定下游任务上的性能
  - 两阶段微调（Two-Stage）
    - 先在其他相同领域的数据上预微调
  - 多任务学习（Multi Task）
    - 多个下游任务同时学习
  - 解冻方法（Unfreezing）
    - 由少到多的微调参数，例如逐层解冻

- 统一所有的自然语言处理任务
  - In-context learning

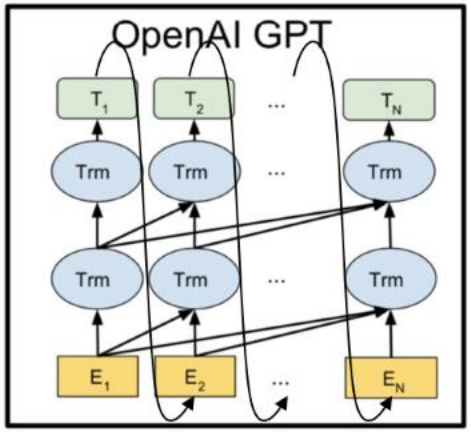
甚至不用微调

- 思路形成：GPT-2

- GPT回顾

单向自回归模型：GPT
结构

- 直接使用语言模型任务进行预训练
- 采用类Transformer Decoder结构（即自回归结构）



Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

预训练语言模型的发展

In-Context Learning

- GPT-2与GPT的区别
  - 更大的模型

参数量	层数	词元嵌入维度
177M (GPT-1)	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

  - 更多的数据
    - 约800万篇文章、总计40G

Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.

预训练语言模型的发展

In-Context Learning

- GPT-2与GPT的本质区别
  - 使用无监督的预训练模型做下游任务

Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.

预训练语言模型的发展

In-Context Learning

- GPT-2与GPT的本质区别
  - 使用无监督的预训练模型不微调做下游任务
  - 更多的无监督数据、更少的有监督数据

Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.

预训练语言模型的发展

In-Context Learning

- GPT-2的核心思想
  - 使用无监督的预训练模型不微调做下游任务
  - 更多的无监督数据、更少的有监督数据

- 原理
  - 语言模型本质上是在建模

$$P(w_{k+1}, w_{k+2}, \cdots, w_N | w_1, w_2, \cdots, w_k)$$

  - 而有监督任务实际上实在建模

$$P(\text{Output}|\text{Input})$$

  - 考虑到每个具体任务会使用不同的模型，可以将任务也作为条件，即

$$P(\text{Output}|\text{Input}, \text{Task})$$

Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.

预训练语言模型的发展
In-Context Learning

- GPT-2的核心思想
  - 使用无监督的预训练模型不微调做下游任务
  - 更多的无监督数据、更少的有监督数据

- 原理
  - 语言模型本质上是在建模

$$P(w_{k+1}, w_{k+2}, \cdots, w_N | w_1, w_2, \cdots, w_k)$$

  - 而有监督任务实际上实在建模

$$P(\text{Output} | \text{Input}, \text{Task})$$

  - 因此有监督任务实际上是无监督语言模型的一种特例
  - 只要模型容量足够大，可以直接通过无监督训练学习到有监督任务

Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.

预训练语言模型的发展

In-Context Learning

- 例如在GPT-2的训练集中存在如下数据

这种语料让模型自动会做机器翻译任务

”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”

“I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’

If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

Input

Output

Task

Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.

预训练语言模型的发展

In-Context Learning

- GPT-3：进一步明确上述思想
  - 更更大的模型
    - GPT-2：48层、1600维、参数量1.5B
    - GPT-3：96层、12888维、序列长度2048、参数量175B
  - 更更多的数据

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

  - 明确概念：In-Context Learning

Brown, Tom B., et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).

- In-Context Learning
  - Zero-Shot场景

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

Brown, Tom B., et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).

预训练语言模型的发展
In-Context Learning

- In-Context Learning
  - Zero-Shot场景
  - One-Shot场景

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

Brown, Tom B., et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).

预训练语言模型的发展
In-Context Learning

- In-Context Learning
  - Zero-Shot场景
  - One-Shot场景
  - Few-Shot场景

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

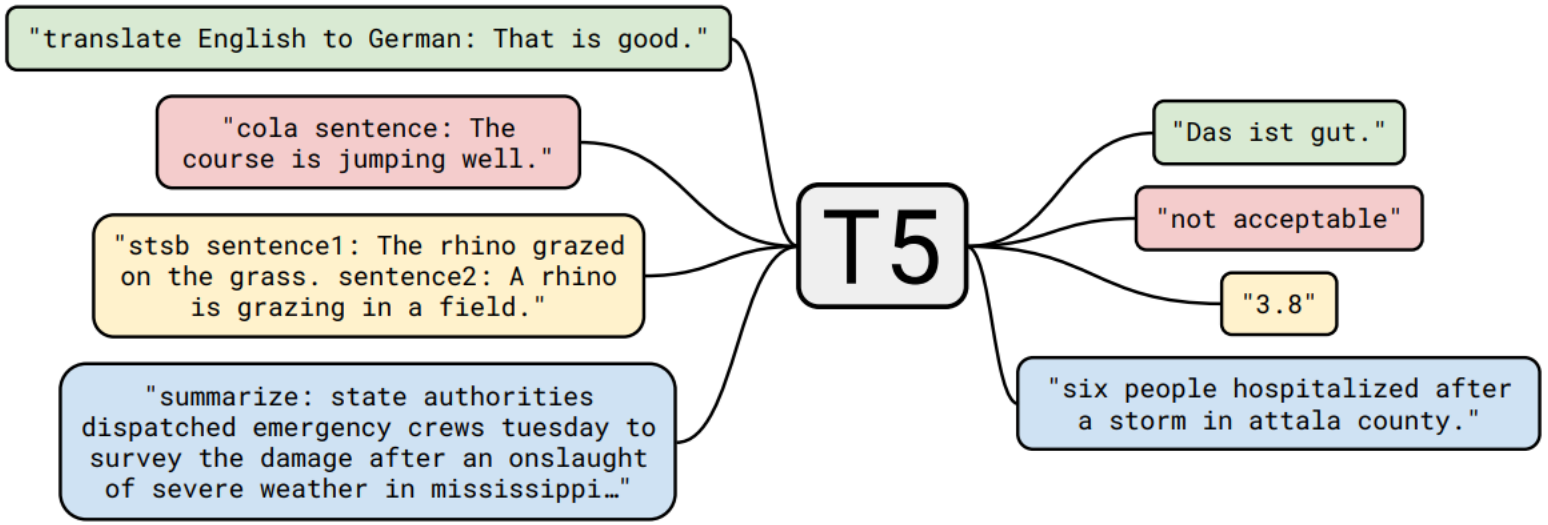
```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée        ←
4  plush girafe => girafe peluche      ←
5  cheese =>                           ← prompt
```

Brown, Tom B., et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).

预训练语言模型的发展

In-Context Learning

- 类似的思想：T5



- 进一步的发展：带提示的微调（Prompt）

Raffel, Colin, et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." Journal of Machine Learning Research 21 (2020): 1-67.

预训练语言模型的发展

ChatGPT

- https://openai.com/blog/chatgpt/

USER **this code is not working like i expect — how do i fix it?**

```
resultWorkerErr := make(chan error)
defer close(resultWorkerErr)
go func() {
        defer cancel()
        resultWorkerErr <- b.resultWorker(ctx)
}()

err := b.worker(ctx)
cancel()
if err == nil {
        return <-resultWorkerErr
}
return multierror.Append(err, <-resultWorkerErr)
```

CHATGPT It's difficult to say what's wrong with the code without more context. Can you provide more information about what the code is supposed to do and what isn't working as expected? Also, is this the entire code or just a part of it?

USER **just part of it — the error never surfaces. I think there is something wrong with the channel**