预训练语言模型最新研究进展

RECENT ADVANCES IN PRE-TRAINED LANGUAGE MODELS

崔一鸣

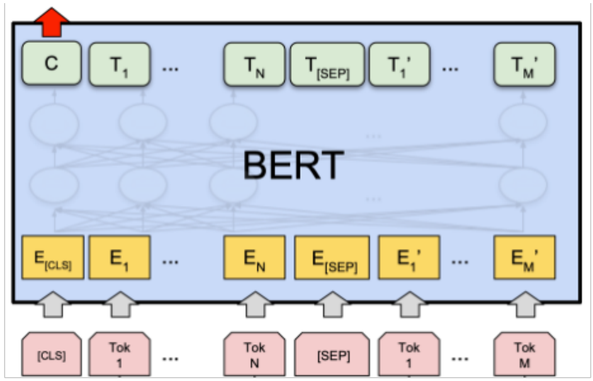
SMP 2022 Tutorial (Online)

2022年8月19日

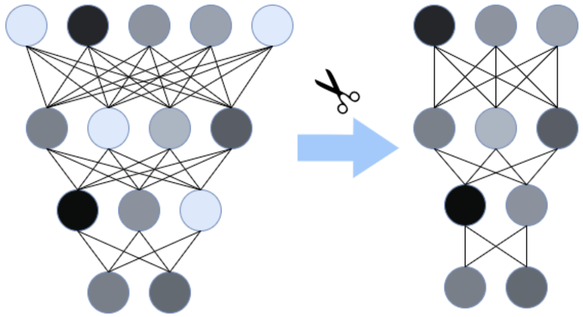
科大讯飞 iFLYTEK

报告内容

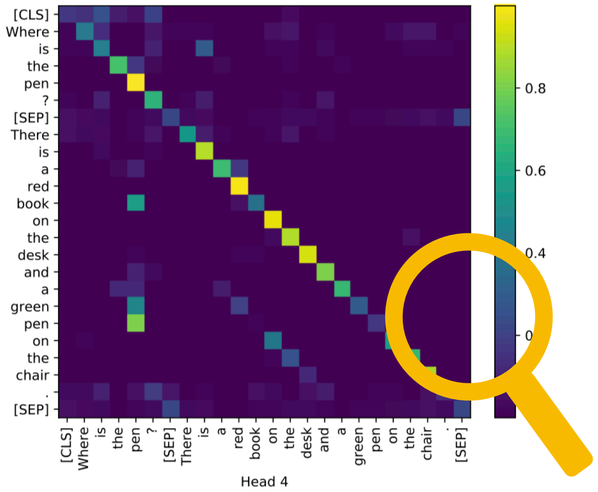
- 概述
- 通用预训练语言模型
  - 基于全词掩码的中文BERT-wwm
  - 基于纠错型掩码语言模型的MacBERT
  - 基于乱序语言模型的PERT
  - 面向少数民族语言的预训练模型CINO
- 预训练语言模型的蒸馏与裁剪
  - 知识蒸馏工具TextBrewer
  - 模型裁剪工具TextPruner
- 预训练语言模型的可解释性
- 总结



预训练语言模型



模型蒸馏与裁剪



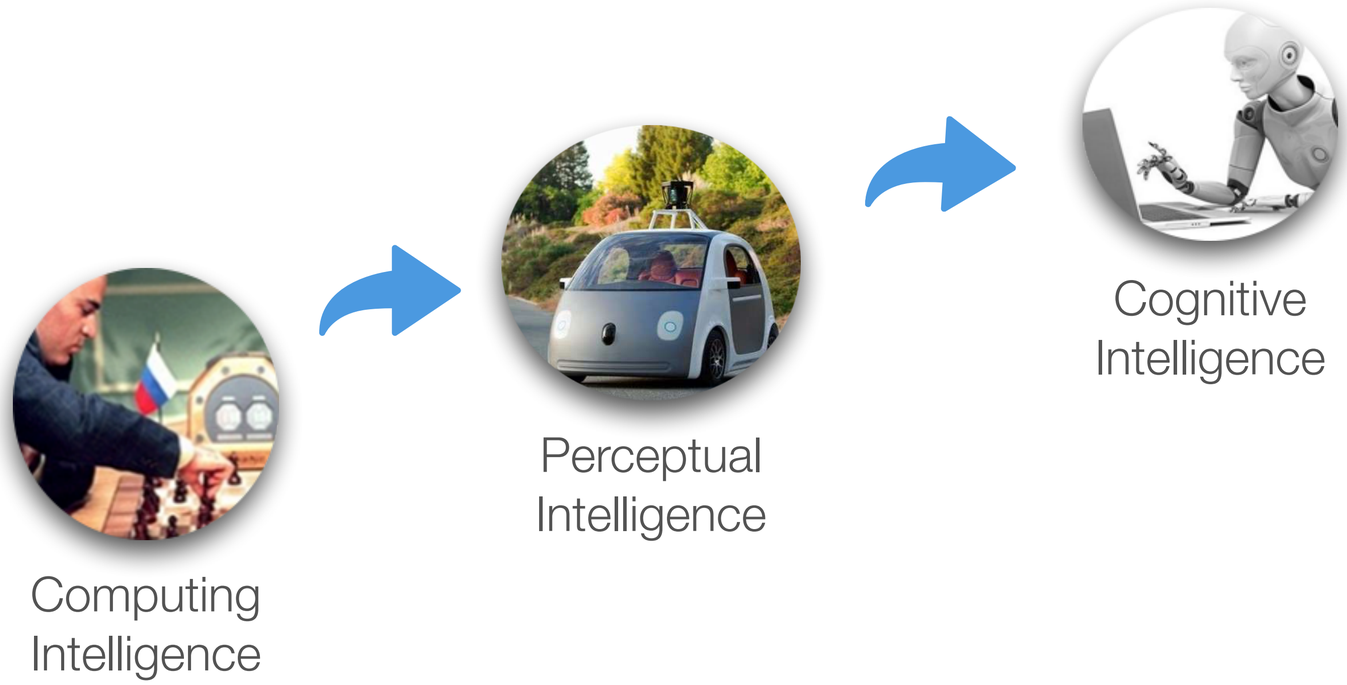
预训练模型可解释性

概述

INTRODUCTION

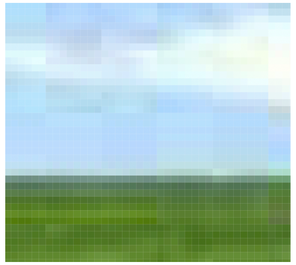
概述

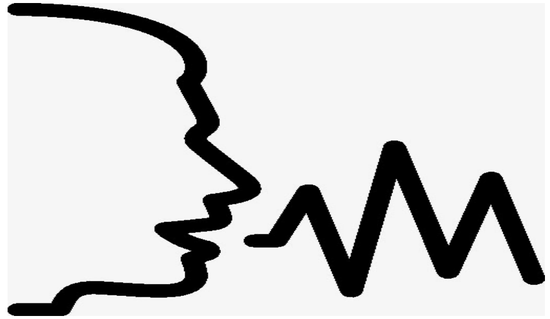
- **人工智能发展的三个阶段**
  - 人工智能的一个重要目标是**让机器能听会说，能理解会思考**
  - 目前人工智能技术正处在从感知智能到认知智能跨越的时间节点
  - 自然语言处理是认知智能中的重要内容，是通往强人工智能的必须攻克的难题



概述

- **自然语言处理为什么难?**
  - 与图像、语音不同，**语言是高度抽象的产物**，其基本组成单位并不是明确的物理量



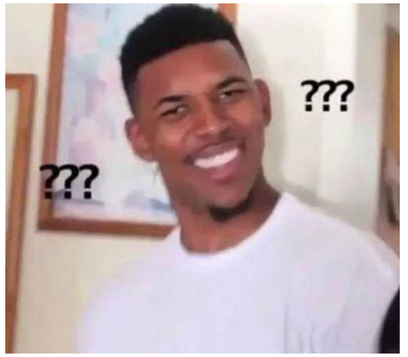


概述

- **自然语言处理为什么难?**
  - 自然语言存在更加错综复杂的现象，相比语音、图像存在更多不确定性
  - 语言是与时俱进的，并不是一成不变的，自然语言处理方法也需要进化

货拉拉拉不拉拉布拉多?

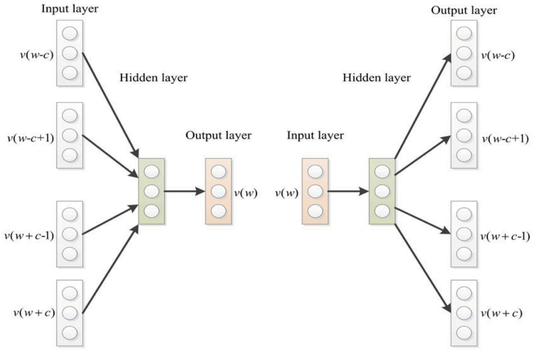
真好吃 → 真　好　吃



他抱不起他的儿子，因为他太 重/虚弱 了。谁 重/虚弱 ?

概述

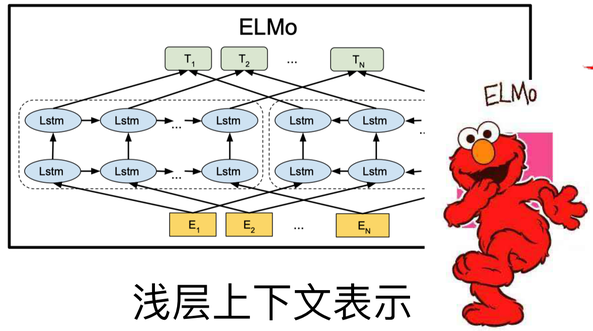
- **自然语言表示的发展一定程度上反映了自然语言处理的发展**
  - 自然语言表示的变迁很大程度影响着自然语言处理的范式
  - 从离散到连续，从上下文无关到上下文相关，从浅层到深层



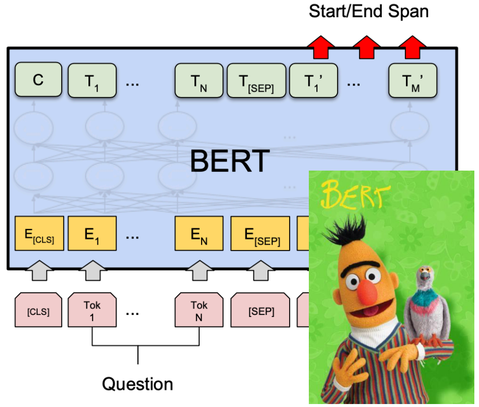
词向量表示

[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

One-hot表示

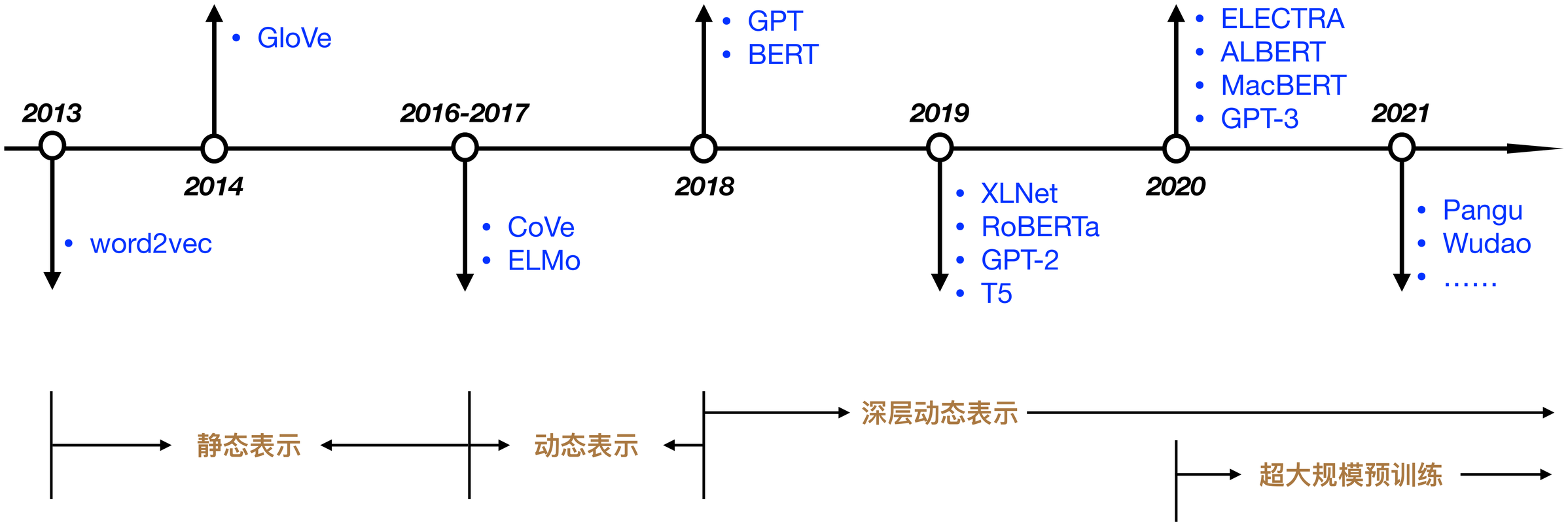


浅层上下文表示



深层上下文表示

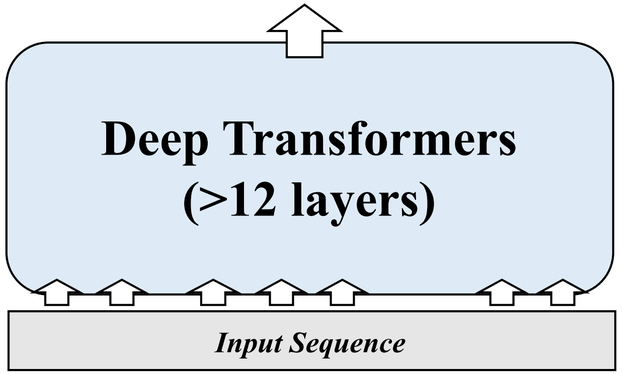
概述


2013
2014
2016-2017
2018
2019
2020
2021
• word2vec
• GloVe
• CoVe
• ELMo
• GPT
• BERT
• XLNet
• RoBERTa
• GPT-2
• T5
• ELECTRA
• ALBERT
• MacBERT
• GPT-3
• Pangu
• Wudao
• ……
静态表示
动态表示
深层动态表示
超大规模预训练

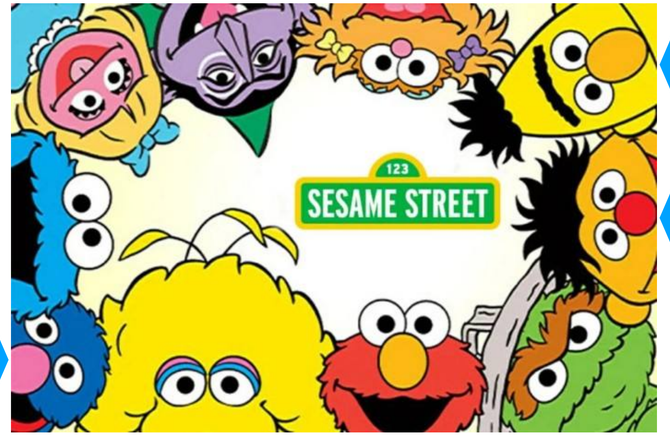
预训练模型三要素

大数据
（无标注文本）

大模型
（深度神经网络）



大算力
（并行计算集群）



BERT

ERNIE

Grover

BigBird

ELMo

Oscar

GPT

OpenAI

GPT-2

GPT-3

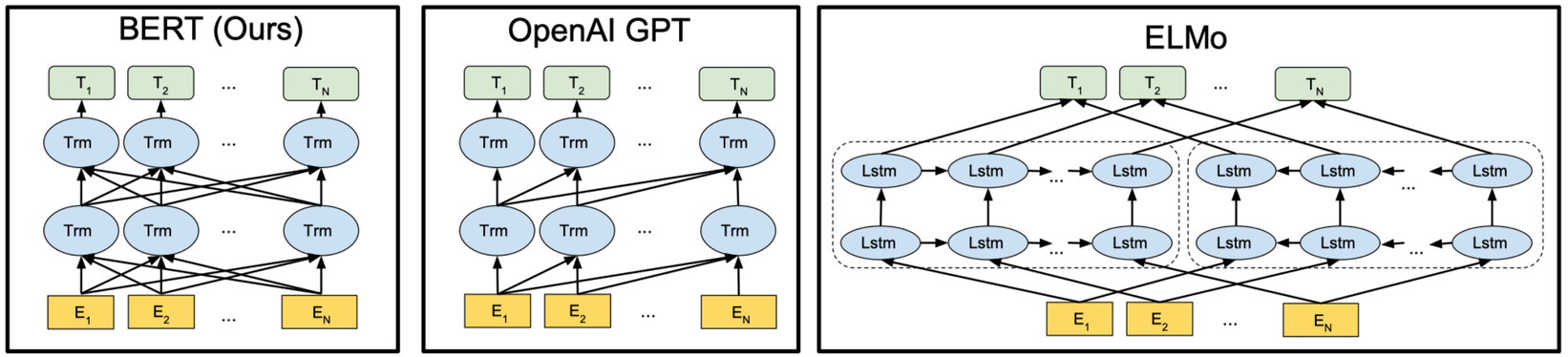
以及更多钱

通用预训练语言模型

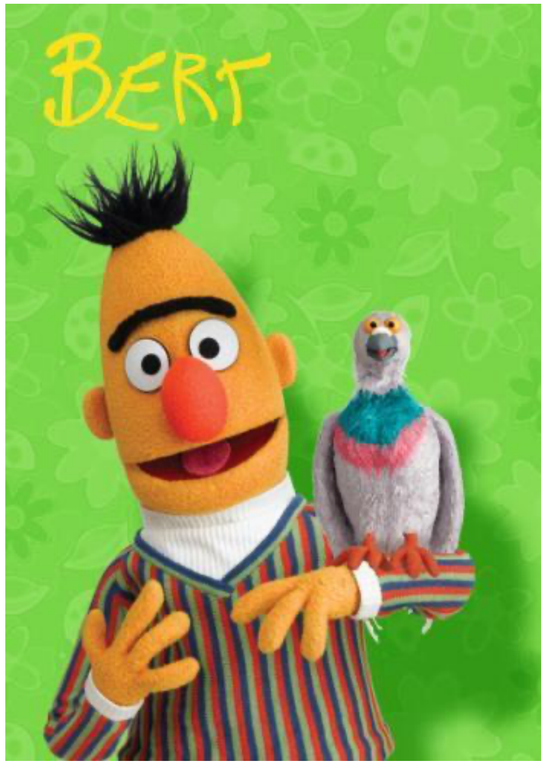
General Pre-trained Language Model

BERT

- **BERT: B**idirectional **E**ncoder **R**epresentations from **T**ransformers
  - 提出了一种双向预训练语言模型方法，利用大规模自由文本训练两个无监督预训练任务
  - BERT在众多NLP任务中获得了显著性能提升
  - 进一步强调了使用通用预训练取代繁杂的任务特定的模型设计



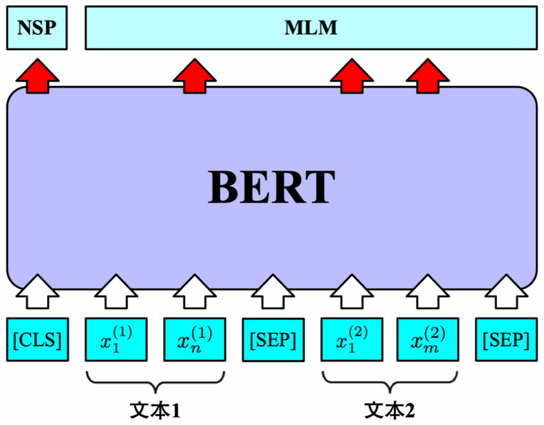
- BERT: **双向**Transformer语言模型
- ELMo: 将**独立的**前向和后向的LSTM语言模型拼接所得
- GPT: **单向**从左至右的Transformer语言模型

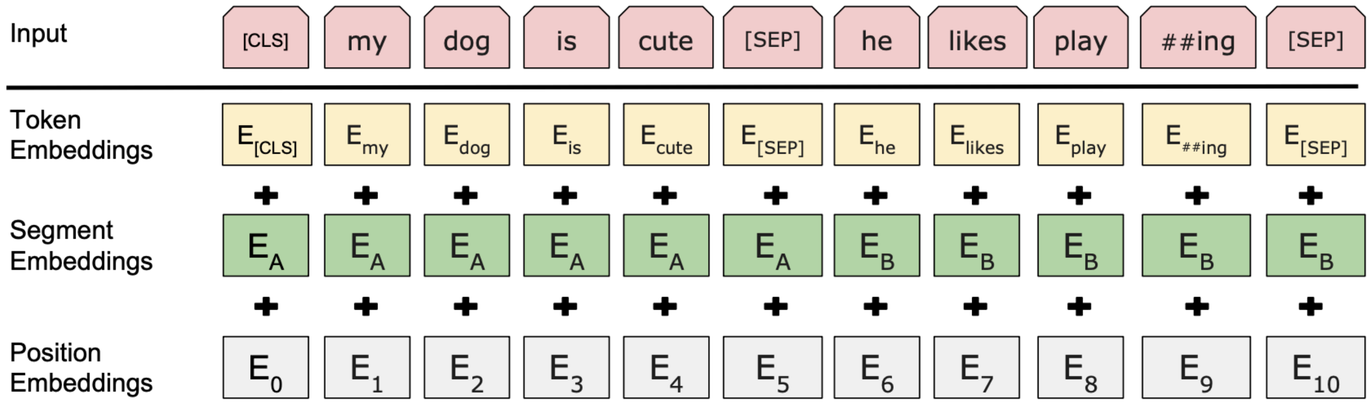


[Devlin et al., 2019] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL 2019.

BERT

- **模型结构**
  - 由深层Transformer模型构成，base版本（12层，110M）和large版本（24层，330M）
  - 使用了一个包含30,000个WordPiece的词表
  - 输入表示包含三部分：词向量、块向量、位置向量
    - 块向量（segment embedding）：表示当前token属于哪一个块（部分）
    - 位置向量（position embedding）：表示当前token的绝对位置

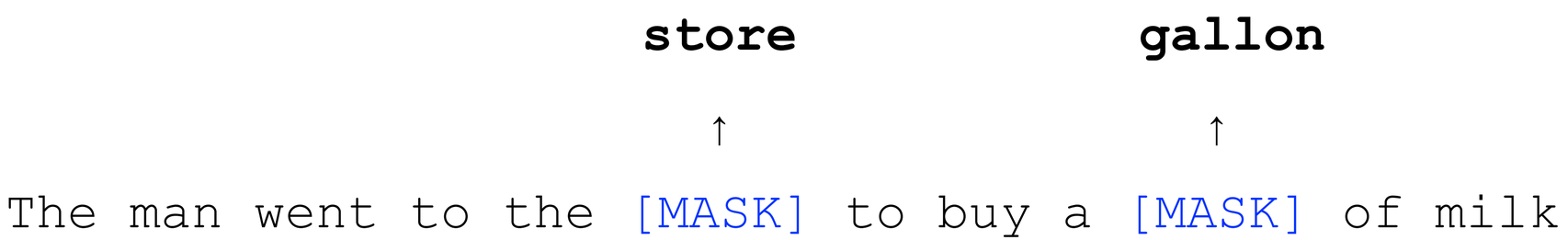




[Devlin et al., 2019] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL 2019.

BERT

- **预训练任务①：Masked Language Model (MLM)**
  - 将输入序列中的部分token进行掩码，并且要求模型将它们进行还原
  - 在BERT中，会将**15%**的输入文本进行mask操作
    - 如果掩码数量太少：模型很容易就能预测出原token是什么
    - 如果掩码数量太多：没有足够的上下文，模型的搜索空间会很大，进而很难进行预测



[Devlin et al., 2019] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL 2019.

BERT

- **预训练任务①：Masked Language Model (MLM)**
  - 问题：预训练阶段使用的表示掩码的[MASK]符号在下游任务中并不会出现
  - 解决方案：并不是将所有待掩码的token都替换成[MASK]

80%
replace w/ [MASK]

went to the **store** → went to the **[MASK]**

10%
replace w/ random word

went to the **store** → went to the **apple**

10%
keep original word

went to the **store** → went to the **store**

[Devlin et al., 2019] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL 2019.

BERT

- **预训练任务②：Next Sentence Prediction (NSP)**
  - 学习两段文本之间的语义关系，即上下文信息
  - 预测Sentence B是否是Sentence A的下一个句子

IsNextSentence

Sentence A: The man went to the store.

Sentence B: He bought a gallon of milk.

NotNextSentence

Sentence A: The man went to the store.

Sentence B: Penguins are flightless.

[Devlin et al., 2019] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL 2019.

BERT

- **实验结果**
  - 在GLUE和SQuAD等任务上相比GPT获得更优的实验结果

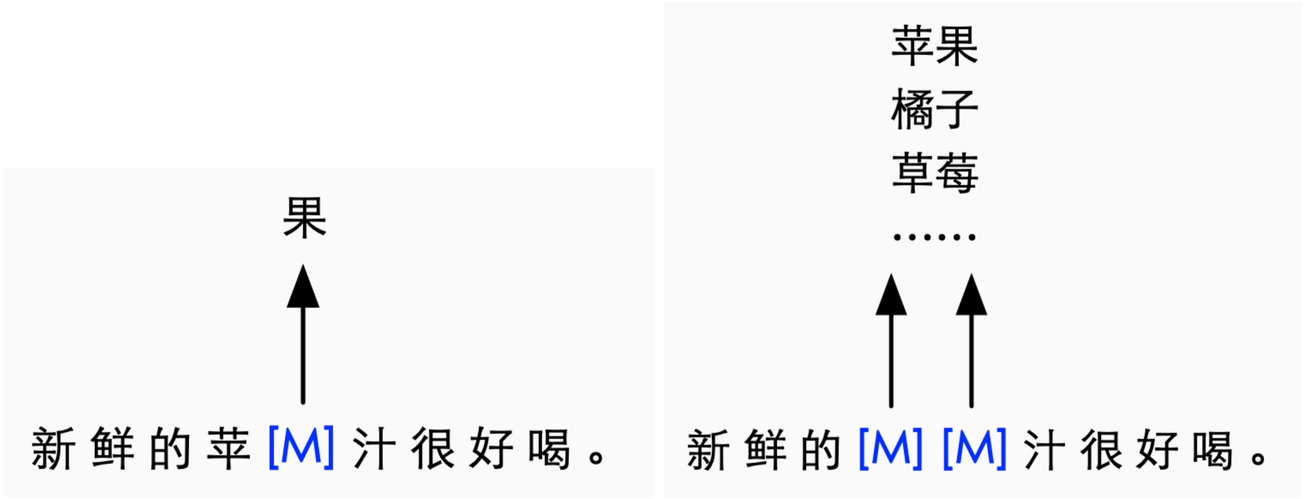
System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

System	Dev EM	Dev F1	Test EM	Test F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	86.6
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**

[Devlin et al., 2019] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL 2019.

中文BERT-wwm

- **基于全词掩码的中文BERT系列模型**
  - 将全词掩码技术（Whole Word Masking，wwm）应用在中文BERT的建模中
  - 不同掩码策略
    - 标准掩码语言模型（MLM）：将输入文本中的字看作独立分布，随机对字进行掩码
    - 全词掩码语言模型（WWM）：考虑了中文分词，将属于同一个词中的所有字进行掩码



(a) 以字为掩码单位　　(b) 以词为掩码单位

[Cui et al., 2020] Revisiting Pre-trained Models for Chinese Natural Language Processing. In Findings of EMNLP 2020.
[Cui et al., 2021] Pre-training with Whole Word Masking for Chinese BERT. In IEEE/ACM TASLP Vol.28.

中文BERT-wwm

- **关于Whole Word Masking的重要提示** 
  - “掩码”是一个广义概念，并不只是代表“替换为[MASK]符号”
  - “掩码”包含三种随机操作：替换为[MASK]、替换为随机词、保持原词

原句	BERT			model	uses	wordpiece		tokenization		.
分词	B	##ER	##T	model	uses	word	##piece	token	##ization	.

MLM	B	**[M]**	**[M]**	model	**uses!**	word	**[M]**	token	##ization	**[M]**
	B	##ER	**[M]**	**model!**	**good**	word	##piece	**[M]**	**[M]**	.
	[M]	##ER	##T	**[M]**	**[M]**	word	**hello**	**token!**	##ization	.

WWM	**[M]**	**[M]**	**[M]**	model	uses	word	##piece	**token!**	**apple**	.
	B!	hello	**[M]**	model	**[M]**	**[M]**	##piece	token	##ization	.
	B	##ER	##T	fruit	uses	word!	**[M]**	**[M]**	**[M]**	.

[M]表示掩码符号，!表示保留原词操作，假设掩码概率为50%

[Cui et al., 2020] Revisiting Pre-trained Models for Chinese Natural Language Processing. In Findings of EMNLP 2020.
[Cui et al., 2021] Pre-training with Whole Word Masking for Chinese BERT. In IEEE/ACM TASLP Vol.28.

中文BERT-wwm

- **实验结果**
  - 基于全词掩码的方法能够显著提升预训练模型性能

Model	SQuAD 1.1	MNLI
BERT-Large, Uncased (Original)	91.0 / 84.3	86.05
BERT-Large, Uncased (WWM)	92.8 / 86.7	87.07
BERT-Large, Cased (Original)	91.5 / 84.8	86.09
BERT-Large, Cased (WWM)	92.9 / 86.7	86.46

▲ 英文任务效果（from Google）

Model	CMRC 2018	XNLI
BERT-base	84.5 / 65.5	77.8
BERT-wwm (base)	85.6 / 66.3	79.0
BERT-wwm-ext (base)	85.7 / 67.1	79.4

▲ 中文任务效果

[Cui et al., 2020] Revisiting Pre-trained Models for Chinese Natural Language Processing. In Findings of EMNLP 2020.
[Cui et al., 2021] Pre-training with Whole Word Masking for Chinese BERT. In IEEE/ACM TASLP Vol.28.

N-gram Masking

- **N元文法掩码（N-gram masking）**
  - 对一个连续的N-gram单元进行掩码，进一步增加MLM任务的难度
  - 预测难度：N-gram Masking > Whole Word Masking > vanilla MLM

Original
原句

We went to the store to buy some fruits.

WWM
整词掩码

We went to the [M] to [M] some [M].

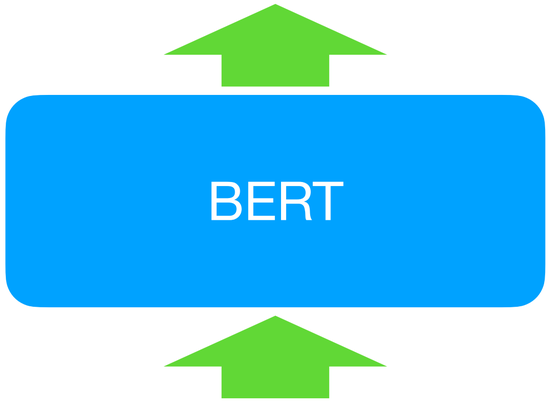
NM
N-gram掩码

We went to the store to [M] [M] [M].

[Cui et al., 2020] Revisiting Pre-trained Models for Chinese Natural Language Processing. In Findings of EMNLP 2020.
[Cui et al., 2021] Pre-training with Whole Word Masking for Chinese BERT. In IEEE/ACM TASLP Vol.28.

N-gram Masking

- **关于WWM、NM的重要提示** 💡
  - MLM、WWM、NM只影响预训练阶段的输入形式
  - 不论使用任意一种MLM、WWM、NM训练出的BERT，**下游精调时的输入均相同**

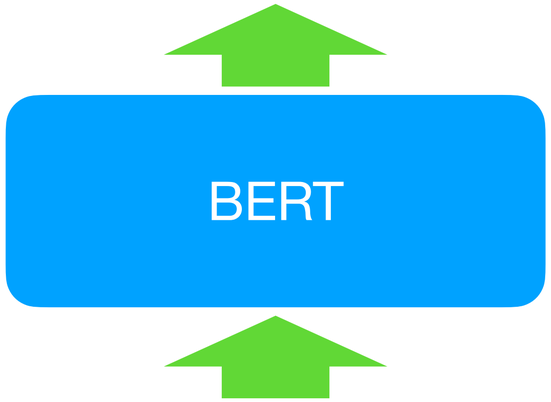


NM: 我 [M] [M] [M] 天 安 门

WWM: [M] 爱 [M] [M] 天 安 门

MLM: 我 [M] [M] 京 天 [M] 门

预训练阶段



✓ MLM/WWM/NM: 我 爱 北 京 天 安 门

✗ MLM/WWM/NM: 我 爱 北京 天安门

下游任务精调阶段

[Cui et al., 2020] Revisiting Pre-trained Models for Chinese Natural Language Processing. In Findings of EMNLP 2020.
[Cui et al., 2021] Pre-training with Whole Word Masking for Chinese BERT. In IEEE/ACM TASLP Vol.28.

MacBERT

- **基于纠错型掩码的预训练模型MacBERT**
  - 在可比条件下，评测了主流预训练模型在中文下的表现
  - 提出了一种新的预训练模型MacBERT，解决BERT中的“预训练-精调”不一致的问题
  - 基于相似词替换的预训练任务（**M**LM **a**s **c**orrection，Mac），类似“文本纠错”

用 语 言 模 型 **预 测** 下 一 个 词

BERT

- 80% of the time, replace with [M]
  - 用 语 言 模 型 **[M] [M]** 下 一 个 词
- 10% of the time, replace random word
  - 用 语 言 模 型 **预 见** 下 一 个 词
- 10% of the time, keep the same word
  - 用 语 言 模 型 **预 测** 下 一 个 词

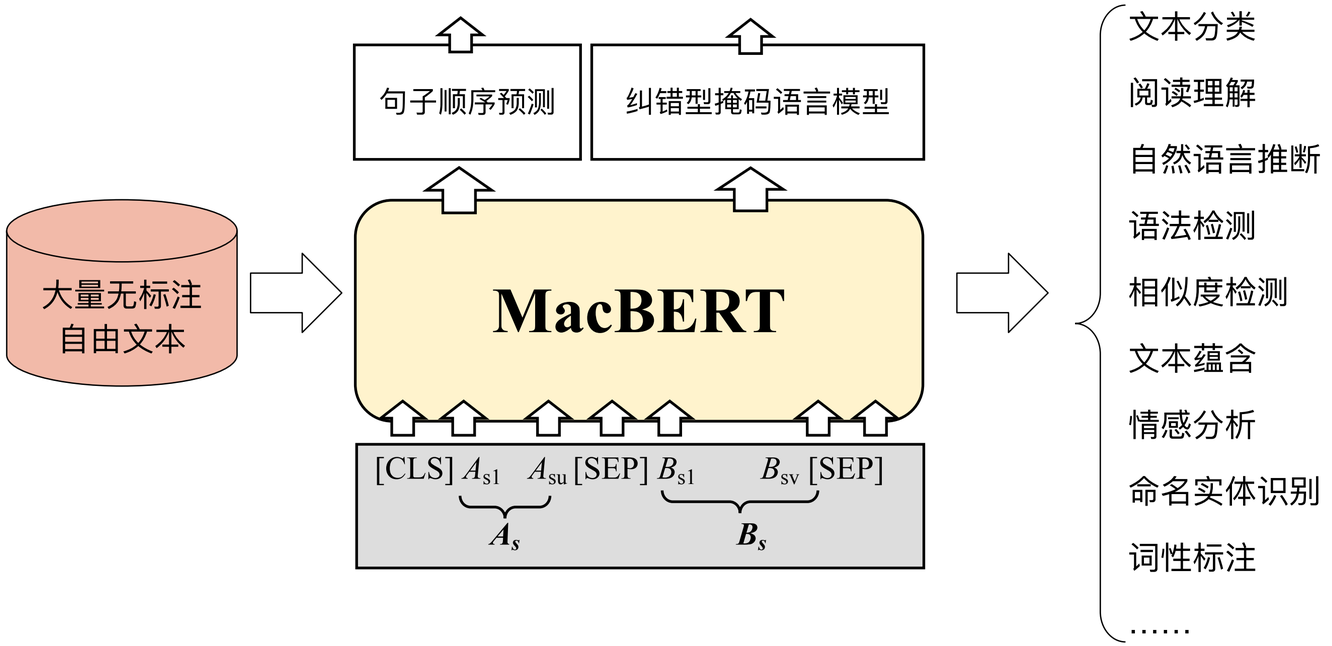
MacBERT

- 80% of the time, replace with synonym
  - 用 语 言 模 型 **预 见** 下 一 个 词
- 10% of the time, replace random word
  - 用 语 言 模 型 **好 是** 下 一 个 词
- 10% of the time, keep the same word
  - 用 语 言 模 型 **预 测** 下 一 个 词

[Cui et al., 2020] Revisiting Pre-trained Models for Chinese Natural Language Processing. In Findings of EMNLP 2020.
[Cui et al., 2021] Pre-training with Whole Word Masking for Chinese BERT. In IEEE/ACM TASLP Vol.28.

MacBERT

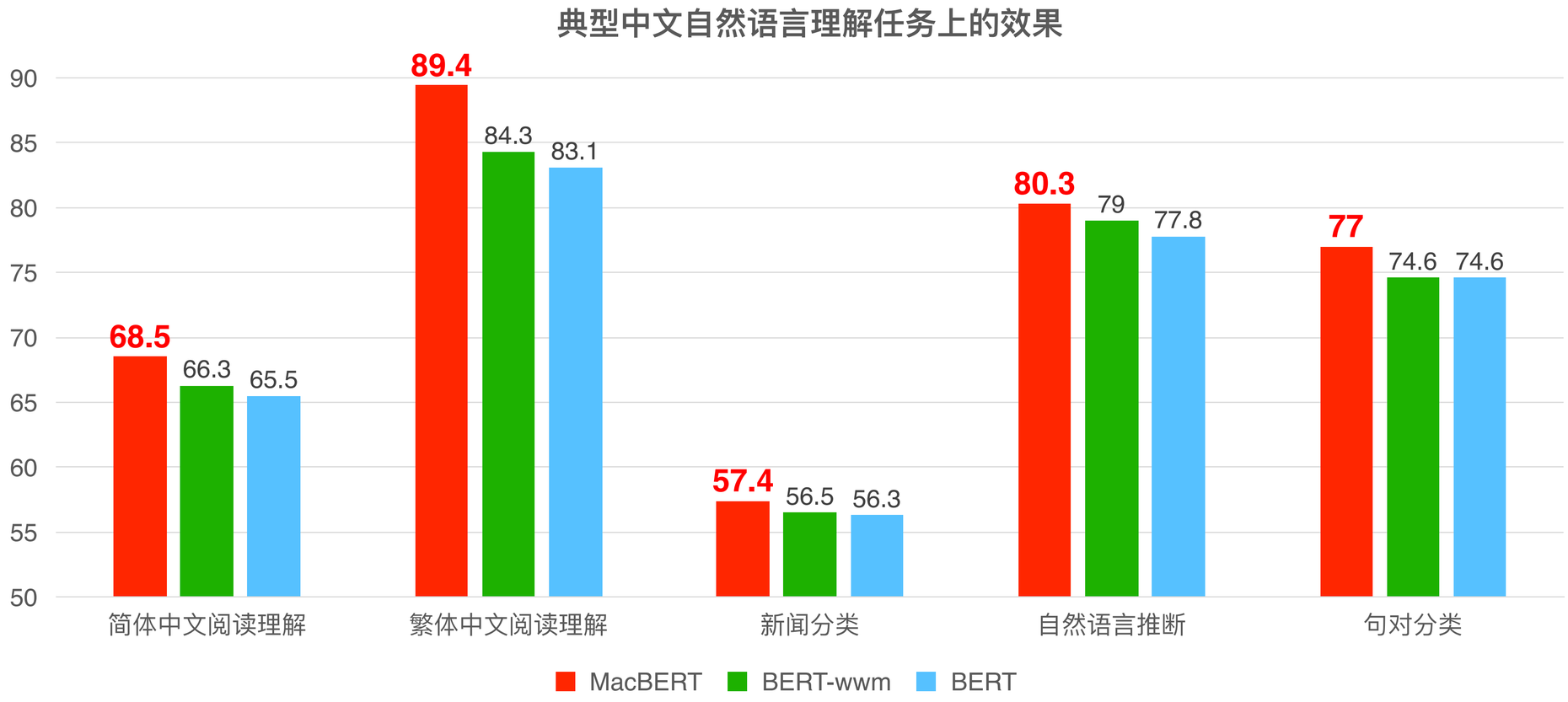
- **模型结构及训练任务**
  - **LM任务**：使用基于相似词替换的预训练任务进行训练
  - **SP任务**：将“下一个句子预测（NSP）”替换为ALBERT中的“句子顺序预测（SOP）”任务



[Cui et al., 2020] Revisiting Pre-trained Models for Chinese Natural Language Processing. In Findings of EMNLP 2020.
[Cui et al., 2021] Pre-training with Whole Word Masking for Chinese BERT. In IEEE/ACM TASLP Vol.28.

MacBERT

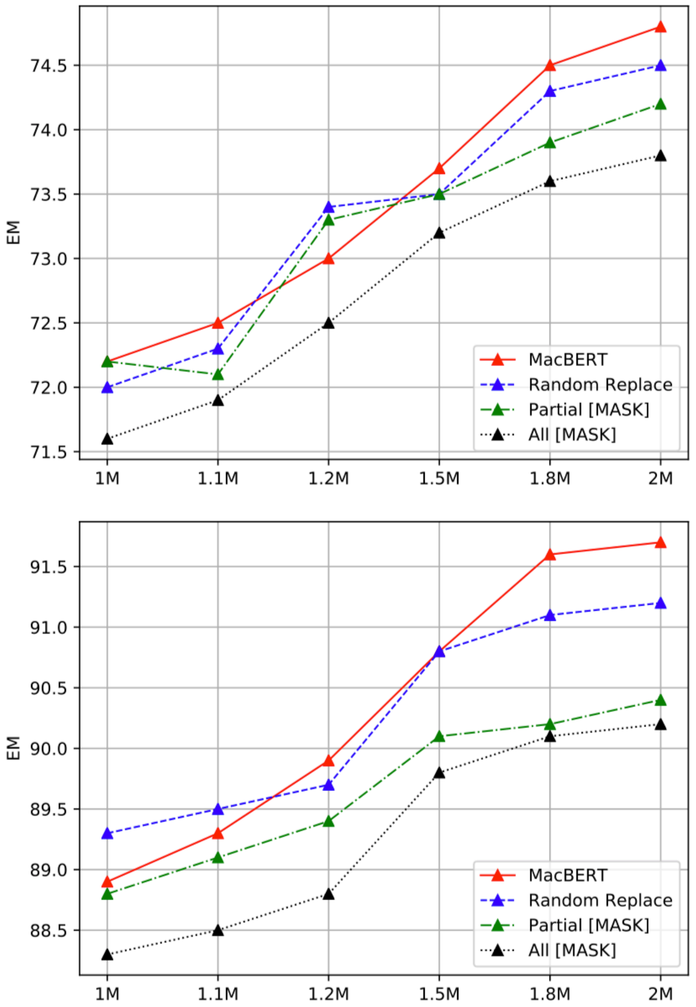
- **实验结果**
  - 在阅读理解、文本分类、自然语言推断等任务上获得显著性能提升



[Cui et al., 2020] Revisiting Pre-trained Models for Chinese Natural Language Processing. In Findings of EMNLP 2020.
[Cui et al., 2021] Pre-training with Whole Word Masking for Chinese BERT. In IEEE/ACM TASLP Vol.28.

MacBERT

- **实验分析：MLM任务**
  - 输入句子中15%的token被处理
    - MacBERT：80%的token被替换为相似词，10%被替换为随机词
    - Random Replace：90%的token被替换为随机词
    - Partial Mask：原始BERT实现，即80%的token被替换为`[MASK]`符号，10%被替换为随机词
    - All Mask：90%的token被替换为`[MASK]`
  - 剩余的10%不做任何替换（负样本）
  - 效果排序
    - MacBERT > random replace > partial mask > all mask



▲ 横轴：训练步数；纵轴：EM值

[Cui et al., 2020] Revisiting Pre-trained Models for Chinese Natural Language Processing. In Findings of EMNLP 2020.
[Cui et al., 2021] Pre-training with Whole Word Masking for Chinese BERT. In IEEE/ACM TASLP Vol.28.

PERT

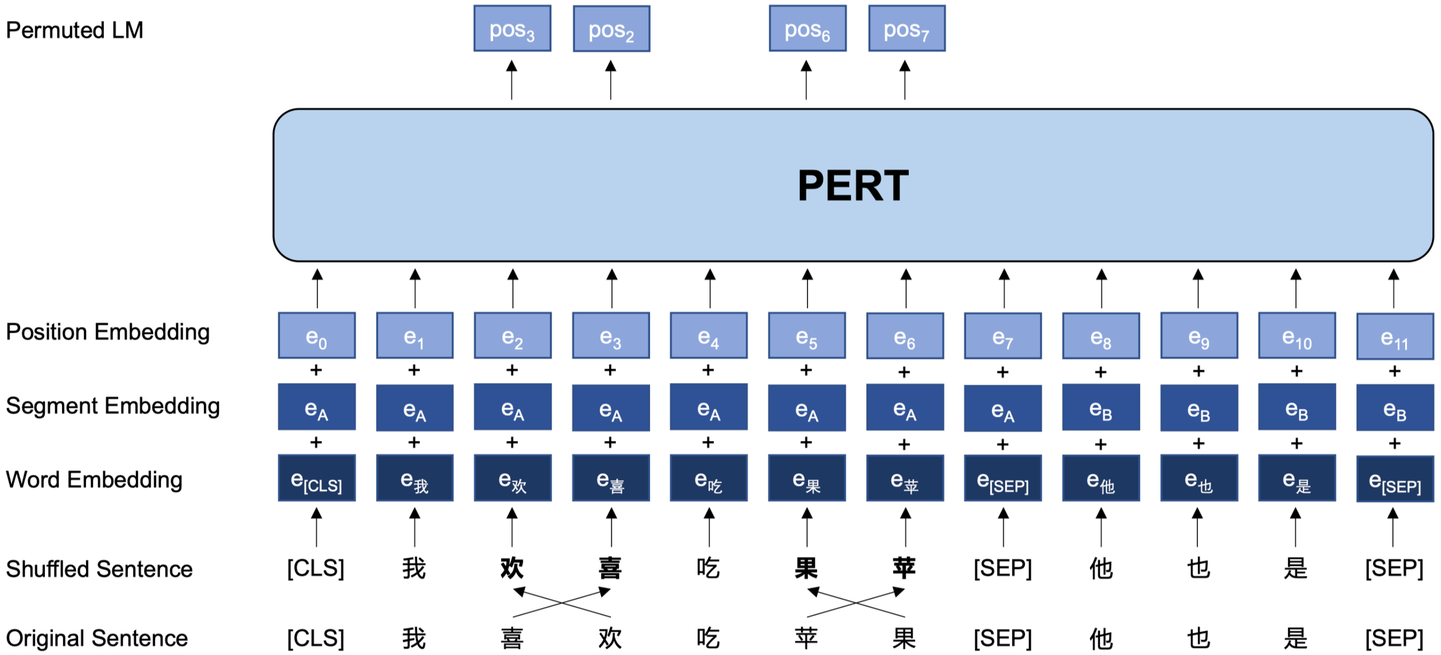
- **PERT: Pre-training BERT with Permuted Language Model**
  - 研究背景
    - 多数自编码预训练模型基于MLM及其变种进行训练
    - 基于MLM的训练方法依赖[MASK]标记，造成“预训练-精调”不一致的问题
  - 问题：有没有其他可以建模文本语义的方式?
  - 观察：发现乱序文本仍然包含与原文本相近的语义

研表究明汉字的序顺并不定一会影响读阅。

[Cui et al., 2022] PERT: Pre-training BERT with Permuted Language Model. arXiv pre-print.

PERT

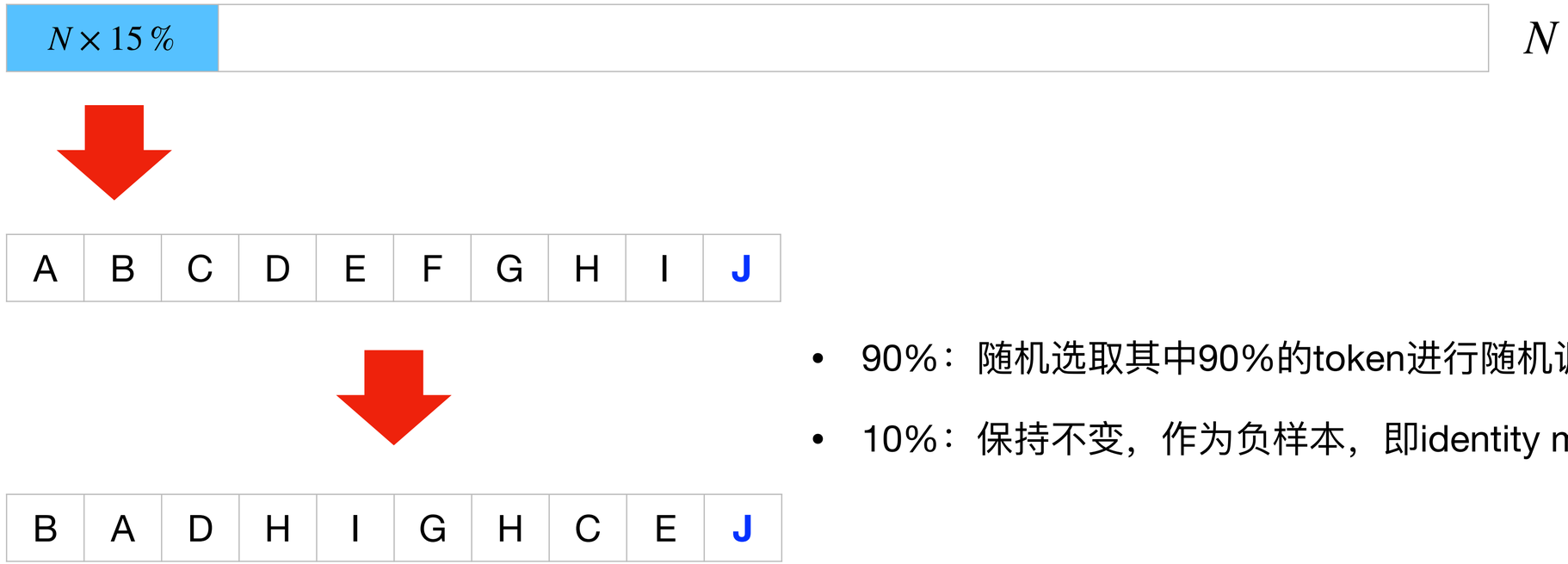
- **模型结构**
  - 提出利用乱序文本预测当前单词在原句的位置，自监督地学习文本语义信息
  - 该过程不会引入掩码标记[MASK]
  - 输出空间是输入序列长度N，而非词表大小V



[Cui et al., 2022] PERT: Pre-training BERT with Permuted Language Model. arXiv pre-print.

PERT

- **Permuted Language Model (PerLM)**
  - 随机选取输入序列长度15%的文本用于掩码
  - 同时使用全词掩码（wwm）和N-gram掩码技术（unigram 40% → 4-gram 10%）



- 90%：随机选取其中90%的token进行随机调序
- 10%：保持不变，作为负样本，即identity mapping

[Cui et al., 2022] PERT: Pre-training BERT with Permuted Language Model. arXiv pre-print.

PERT

- **实验效果：中文NLU任务**
  - 使用20G中文文本训练了PERT模型，其中包含维基百科、新闻、社区问答等语料
  - 实验结果：PERT在机器阅读理解、命名实体识别任务上能够获得显著性能提升

System	CMRC 2018						DRCD			
	D-EM	D-F1	T-EM	T-F1	C-EM	C-F1	D-EM	D-F1	T-EM	T-F1
$\text{BERT}_{\text{base}}$	67.1 (65.6)	85.7 (85.0)	71.4 (70.0)	87.7 (87.0)	24.0 (20.0)	47.3 (44.6)	85.0 (84.5)	91.2 (90.9)	83.6 (83.0)	90.4 (89.9)
$\text{RoBERTa}_{\text{base}}$	67.4 (66.5)	87.2 (86.5)	72.6 (71.4)	89.4 (88.8)	26.2 (24.6)	51.0 (49.1)	86.6 (85.9)	92.5 (92.2)	85.6 (85.2)	92.0 (91.7)
$\text{ELECTRA}_{\text{base}}$	68.4 (68.0)	84.8 (84.6)	73.1 **(72.7)**	87.1 (86.9)	22.6 (21.7)	45.0 (43.8)	87.5 (87.0)	92.5 (92.3)	86.9 (86.6)	91.8 (91.7)
$\text{MacBERT}_{\text{base}}$	**68.5** (67.3)	**87.9 (87.1)**	**73.2** (72.4)	**89.5 (89.2)**	**30.2** (26.4)	54.0 (52.2)	89.4 **(89.2)**	**94.3 (94.1)**	**89.5 (88.7)**	**93.8 (93.5)**
$\text{PERT}_{\text{base}}$	**68.5 (68.1)**	87.2 **(87.1)**	72.8 (72.5)	89.2 (89.0)	28.7 **(28.2)**	**55.4 (53.7)**	**89.5** (88.9)	93.9 (93.6)	89.0 (88.5)	93.5 (93.2)
$\text{RoBERTa}_{\text{large}}$	68.5 (67.6)	88.4 (87.9)	74.2 (72.4)	90.6 (90.0)	31.5 (30.1)	60.1 (57.5)	89.6 (89.1)	94.8 (94.4)	89.6 (88.9)	94.5 (94.1)
$\text{ELECTRA}_{\text{large}}$	69.1 (68.2)	85.2 (84.5)	73.9 (72.8)	87.1 (86.6)	23.0 (21.6)	44.2 (43.2)	88.8 (88.7)	93.3 (93.2)	88.8 (88.2)	93.6 (93.2)
$\text{MacBERT}_{\text{large}}$	70.7 (68.6)	88.9 (88.2)	74.8 (73.2)	**90.7** (90.1)	31.9 (29.6)	**60.2** (57.6)	**91.2 (90.8)**	**95.6 (95.3)**	**91.7 (90.9)**	**95.6 (95.3)**
$\text{PERT}_{\text{large}}$	**72.2 (71.0)**	**89.4 (88.8)**	**76.8 (75.5)**	**90.7 (90.4)**	**32.3 (30.9)**	59.2 **(58.1)**	90.9 **(90.8)**	95.5 (95.2)	91.1 (90.7)	95.2 (95.1)

▲ 阅读理解效果

System	MSRA-NER (Test)			People's Daily (Dev)		
	P	R	F	P	R	F
$\text{BERT}_{\text{base}}$	95.2 (94.8)	95.4 (95.1)	95.3 (94.9)	95.3 **(95.1)**	95.3 (95.1)	**95.3 (95.1)**
$\text{RoBERTa}_{\text{base}}$	95.3 (94.9)	95.6 (95.4)	95.5 (95.1)	94.9 (94.8)	95.3 (95.1)	95.1 (94.9)
$\text{ELECTRA}_{\text{base}}$	95.0 (94.5)	**95.9** (95.4)	95.4 (95.0)	94.8 (94.7)	95.3 **(95.2)**	95.1 (94.9)
$\text{MacBERT}_{\text{base}}$	95.2 (94.9)	95.4 (95.4)	95.3 (95.1)	94.9 (94.6)	95.6 (95.1)	95.2 (94.9)
$\text{PERT}_{\text{base}}$	**95.4 (95.2)**	95.5 **(95.5)**	**95.6 (95.3)**	**95.4 (95.1)**	95.2 (95.0)	**95.3 (95.1)**
$\text{RoBERTa}_{\text{large}}$	95.4 (95.3)	95.7 (95.7)	95.5 (95.5)	95.7 (95.4)	95.7 (95.4)	95.7 (95.4)
$\text{ELECTRA}_{\text{large}}$	94.9 (94.8)	95.5 (95.0)	95.0 (94.8)	94.8 (94.6)	95.3 (95.3)	94.9 (94.8)
$\text{MacBERT}_{\text{large}}$	96.3 (95.8)	96.3 (95.9)	**96.2** (95.9)	95.8 (95.6)	95.8 **(95.7)**	95.8 (95.7)
$\text{PERT}_{\text{large}}$	**96.4 (95.9)**	**96.4 (96.1)**	**96.2 (96.0)**	**96.3 (96.0)**	96.0 (95.7)	**96.1 (95.8)**

▲ 命名实体识别效果

[Cui et al., 2022] PERT: Pre-training BERT with Permuted Language Model. arXiv pre-print.

PERT

- **实验效果：中文语法纠错任务**
  - 针对语法中的文本乱序问题，利用PERT搭建序列标注模型进行实验
  - 在四个领域（维基、公文、海关、政法）下，PERT均显著优于所有基线系统效果

我每天一个吃苹果 ⟶ 我每天吃一个苹果

System	Wikipedia			Formal Doc.			Customs			Legal		
	P	R	F	P	R	F	P	R	F	P	R	F
$\text{BERT}^{\text{Google}}_{\text{base}}$	83.6	76.3	79.8	92.1	87.1	89.6	85.7	85.1	85.4	94.3	89.8	92.0
$\text{RoBERTa}_{\text{base}}$	84.2	76.9	80.4	92.6	87.7	90.1	86.8	85.9	86.3	94.6	90.0	92.2
$\text{ELECTRA}_{\text{base}}$	69.9	57.8	63.6	88.1	81.6	84.7	69.6	71.0	70.3	91.7	85.4	88.4
$\text{MacBERT}_{\text{base}}$	84.3	77.1	80.5	92.7	87.8	90.2	86.4	86.5	86.4	94.6	90.1	92.3
$\text{PERT}_{\text{base}}$	**86.5**	**79.5**	**82.9**	**93.6**	**89.0**	**91.2**	**88.3**	**88.0**	**88.2**	**95.2**	**90.7**	**92.9**

▲ 文本纠错-乱序问题效果

[Cui et al., 2022] PERT: Pre-training BERT with Permuted Language Model. arXiv pre-print.

PERT

- **实验效果：英文NLU任务**
  - 使用了经典的Wikipedia+Books数据进行训练
  - 实验结果：部分任务上优于其他同等数据训练的预训练模型

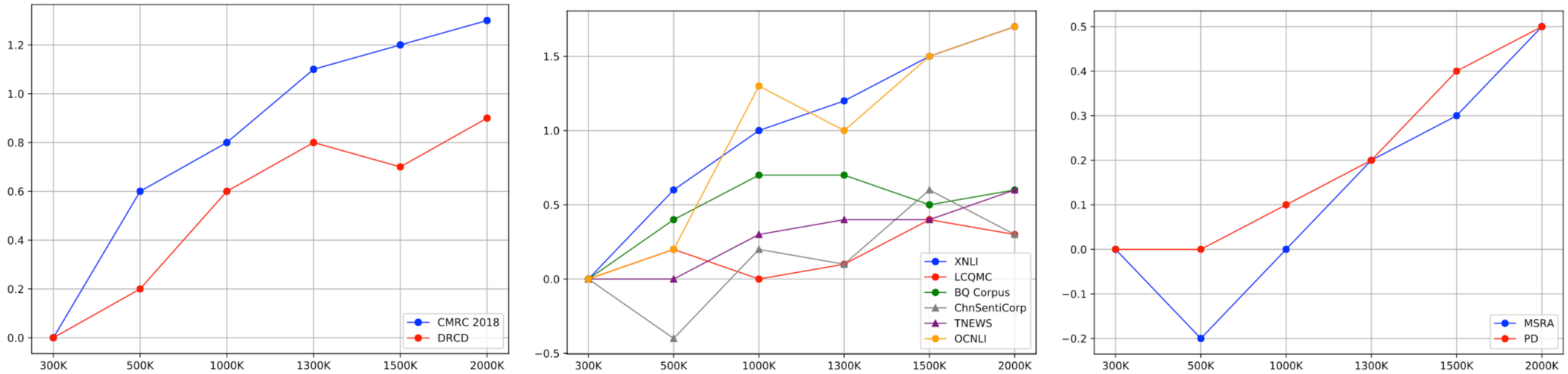
System	SQuAD EM	SQuAD F1	SQuAD 2.0 EM	SQuAD 2.0 F1	MNLI Acc	SST-2 Acc	CoLA M.C.	MRPC Acc
$\text{BERT}_{\text{base}}$	80.8	88.5	-	-	84.4	**92.7**	60.6	86.7
$\text{BERT}_{\text{base}}^{\dagger}$	81.2	88.5	72.4	75.4	84.4	92.6	59.3	86.0
$\text{RoBERTa}_{\text{base}}$	-	90.4	-	79.1	84.7	92.5	-	-
$\text{XLNet}_{\text{base}}$	-	-	**78.4**	**81.3**	**85.8**	92.6	-	-
$\text{ALBERT}_{\text{base}}$	82.1	89.3	76.1	79.1	81.9	89.4	-	-
$\text{PERT}_{\text{base}}$	**84.8**	**91.3**	78.3	81.0	84.5	92.0	**61.2**	**87.5**
$\text{BERT}_{\text{large}}$	84.1	90.9	78.7	81.9	86.6	93.2	60.6	88.0
$\text{BERT}_{\text{large-wwm}}^{\dagger}$	87.4	93.4	82.8	85.6	87.3	93.4	63.1	87.2
$\text{RoBERTa}_{\text{large}}$	-	93.6	-	87.3	**89.0**	**95.3**	-	-
$\text{XLNet}_{\text{large}}$	**88.2**	**94.0**	**85.1**	**87.8**	88.4	94.4	65.2	**90.0**
$\text{ALBERT}_{\text{large}}$	84.1	90.9	79.0	82.1	83.8	90.6	-	-
$\text{PERT}_{\text{large}}$	87.4	93.3	83.5	86.3	87.6	93.4	**65.7**	87.3

▲ 英文任务效果

[Cui et al., 2022] PERT: Pre-training BERT with Permuted Language Model. arXiv pre-print.

PERT

- **不同训练步数下的性能对比**
  - 阅读理解、命名实体识别任务性能趋势基本一致，随着训练步数增加，其性能也呈现增长趋势
  - 多数文本分类任务在训练中途达到局部最优效果
  - “预训练”也和“下游任务精调”一样，最终时刻并不一定是最好结果



(a) MRC (b) TC (c) NER

▲ **横轴**：训练步数；**纵轴**：以300K效果为基准计算性能差值

[Cui et al., 2022] PERT: Pre-training BERT with Permuted Language Model. arXiv pre-print.

PERT

- **分析：不同调序粒度**
  - 在PERT中，被选中的词可以和输入中的任何一个词进行调换，对文本自然度破坏较大
  - 分析不同调序粒度对性能的影响：word，N-gram，sentence
  - 实验结果表明
    - 调序粒度越小，虽然对文本的自然度破坏更小，却不利于文本的语义学习
    - 对阅读理解任务影响较大，对文本分类任务影响较小

System	CMRC 2018						XNLI		TNEWS	OCNLI	Average
	D-EM	D-F1	T-EM	T-F1	C-EM	C-F1	Dev	Test	Dev	Dev	
$\text{PERT}_{\text{base}}$ (no limit)	65.4	85.0	70.2	87.3	22.4	45.6	74.8	74.4	54.5	70.6	65.02
└Word	59.3	80.6	64.8	83.5	12.6	32.2	73.2	72.1	53.2	69.3	60.08
└N-gram	62.2	82.5	67.3	84.8	17.2	36.1	73.4	73.2	53.8	69.5	62.00
└Sentence	63.7	83.2	69.1	86.2	16.8	38.0	74.3	73.0	54.1	70.0	62.84

▲ 不同调换粒度下的实验结果

[Cui et al., 2022] PERT: Pre-training BERT with Permuted Language Model. arXiv pre-print.

PERT

- 分析：**Global v.s. Local Prediction**
  - **Global**：在整个词表上预测，即 $y \in \mathbb{R}^{|V|}$
  - **Local**：在输入句子中预测，即 $y \in \mathbb{R}^{N}$（其中N为输入长度）
  - 在PERT框架下，在局部空间预测效果相对较好，结合两者并未带来额外的性能提升

System	CMRC 2018						XNLI		TNEWS	OCNLI	Average
	D-EM	D-F1	T-EM	T-F1	C-EM	C-F1	Dev	Test	Dev	Dev	
PERT$_{\text{base}}$ (local)	64.1	84.0	69.1	86.5	21.0	43.3	74.1	74.4	54.5	70.6	64.16
└Global	61.1	81.4	65.8	84.3	15.8	36.2	73.6	74.0	55.4	69.4	61.70
└Local + Global	63.2	83.6	67.7	85.8	19.3	42.0	74.6	74.6	55.1	70.0	63.59

▲ Global v.s. Local Prediction实验效果对比

[Cui et al., 2022] PERT: Pre-training BERT with Permuted Language Model. arXiv pre-print.

PERT

- 分析：**Partial v.s. Full Prediction**
  - **Partial**：只对调换的token进行预测，即整个序列长度的15%
  - **Full**：对输入序列中的所有token进行预测
  - Full Prediction未能在PERT框架下带来更好的实验结果
  - 并非所有预训练任务都适合ELECTRA-style预测方式，即Full Prediction

System	CMRC 2018						XNLI		TNEWS	OCNLI	Average
	D-EM	D-F1	T-EM	T-F1	C-EM	C-F1	Dev	Test	Dev	Dev	
$\text{PERT}_{\text{base}}$ (partial)	64.1	84.0	69.1	86.5	21.0	43.3	74.1	74.4	54.5	70.6	64.16
└Full Prediction	63.7	84.1	68.0	86.1	18.7	40.9	74.3	73.9	54.2	71.0	63.49

▲ Partial v.s. Full Prediction实验效果对比

[Cui et al., 2022] PERT: Pre-training BERT with Permuted Language Model. arXiv pre-print.

CINO

- **少数民族语言预训练模型CINO**
  - 少数民族语言NLP资源较为稀缺，现有多语言预训练模型无法很好地处理少数民族语言文字
  - 在这项工作中，
    - 推出业界**首个少数民族语言预训练模型CINO**，支持8种语言或方言，填补了相关资源空缺
    - 构建两个面向少数民族语言的NLU评测**数据集WCM和CMNews**

少数民族语言预训练模型

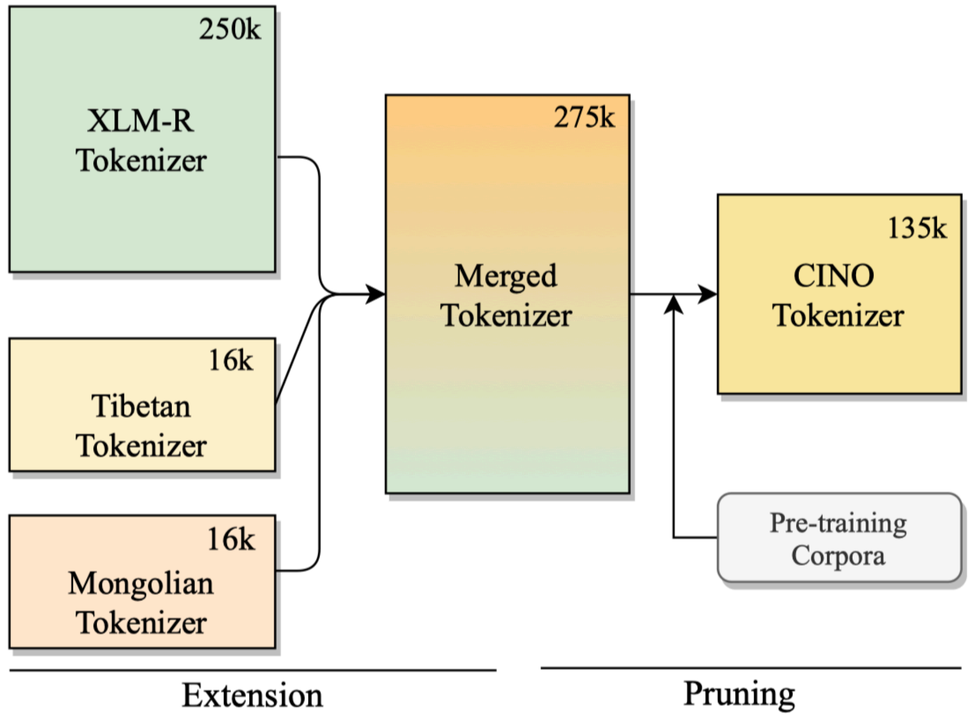
CINO: Chinese Minority PLM

蒙古语	藏语	维吾尔语	壮语	朝鲜语	哈萨克语	普通话	粤语
ᠲᠡᠷᠡᠭᠦᠨ ᠤ ᠳᠡᠭᠡᠭᠦᠷᠢ ᠳᠦᠷᠪᠡᠨ ᠵᠢᠯ ᠤᠨ ᠲᠡᠮᠳᠡᠭᠯᠡᠯ ᠲᠠᠢ ᠪᠤᠯᠤᠭ ᠠᠷᠪᠠᠨ ᠨᠢᠭᠡᠳᠦᠭᠡᠷ ᠰᠠᠷ ᠤᠨ ᠡᠬᠢᠨ ᠡᠷᠬᠢᠮ ᠵᠢᠨ ᠪᠠᠭᠲᠠᠯᠠᠭᠰᠠᠨ ᠪᠠᠶᠢᠨ ᠠ	འདི་ནི་དམངས་ཚན་རིག་ཟློས་གཞུང་ས་ཆེས་ཀྱི་ མི་དམངས་རིག་གནས་ཀྱི་ ས་ཆེས་པས་ དམངས་ཚན་རིག་ཟློས་ཀྱི་ ས་ཆེས་སུ་སྣོར་བར་འགྲོ་མ་བོད...	شىنجاڭ قۇمۇل شەھىرى باركۆل قازاق ئاپتونوم ناھىيىسى ئۆمىكىنىڭ ئارتىسلىرى كۆلياقا	Mbouj lumz cosim, cij ndaej miz goek miz byai. Gij cosim caeuq sawjmingh...	현재 고령수당제도는 이미 성급 차원에서 모두 실현되였다...	سول كۆنى 2021 – جىلغى دۇنيە جۇزلىك سيفرلى ساۋدا ءماجلىسى حۇبەيدىڭ ۋ	神舟十二号载人飞船刚刚平安返回，神舟十三号又进入了...	交警方面呢坚决唔允许白天施工，嘅啊政协委员就问喇...

[Yang et al., 2022] CINO: A Chinese Minority Pre-trained Language Model. In COLING 2022.

CINO

- **模型结构和训练①：词表对少数民族语言进行适配**
  - 在XLM-R的基础上，增加了对藏语和蒙古语的支持，分别增加了16K词表
  - 去除了语料中不会出现的多语种单词，以减小词向量矩阵大小，**从275K缩减至135K**



[Yang et al., 2022] CINO: A Chinese Minority Pre-trained Language Model. In COLING 2022.

CINO

- **模型结构和训练②：重采样机制优化**
  - 为了平衡不同语种的数据，XLM-R和InfoXLM中使用了基于多项式分布的采样策略
  - 采样时进一步考虑将序列长度分布纳入计算，采样粒度从sample → token

$$p_i = \frac{n_i^\alpha}{\sum_k^N n_k^\alpha},$$

XLM-R
(Conneau and Lample, 2019)

InfoXLM
(Chi et al., 2021)

$$p_i = \frac{n_i^\alpha / l_i^\beta}{\sum_i^N n_k^\alpha / l_k^\beta}$$

调节系数

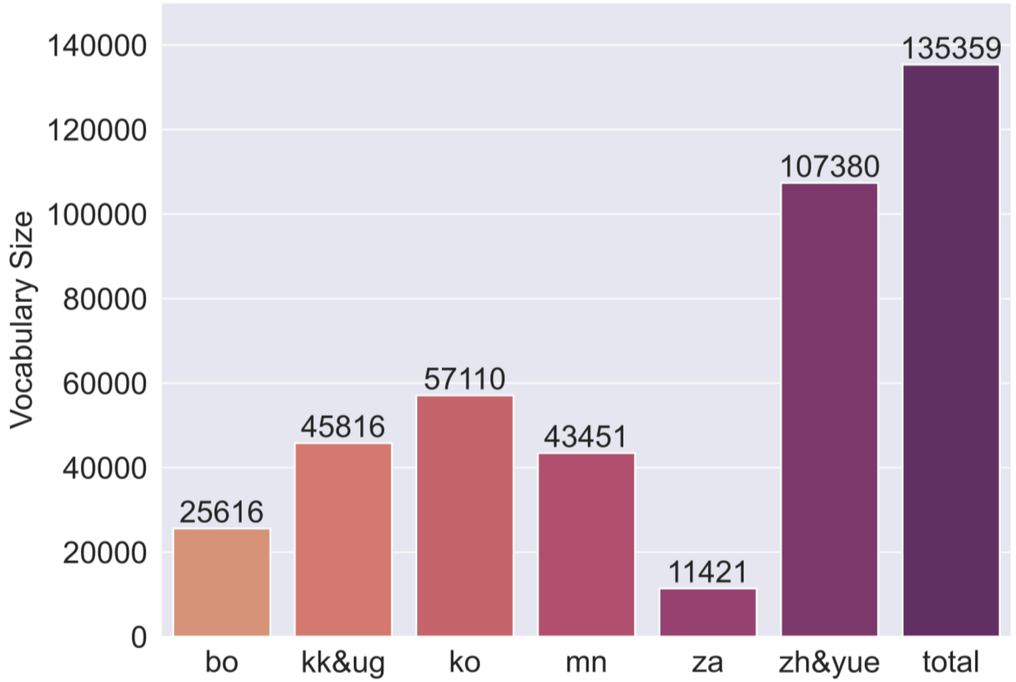
$$\tilde{c}_i \propto p_i l_i \propto n_i^\alpha l_i^{1-\beta} = (n_i l_i)^\alpha = c_i^\alpha$$

[Yang et al., 2022] CINO: A Chinese Minority Pre-trained Language Model. In COLING 2022.

CINO

- **模型结构和训练③：快速MLM**
  - 不同语种可能属于同一语系，其在整个词表中的占比可能较低
  - 因此在MLM预测时可将部分语种进行合并，并只在对应语种词表上进行预测
  - 应用快速MLM技术可**节省10%**训练时间；与词表裁剪共同使用可**节省40%**训练时间

ISO Code	Language Name	Language Family	Writing System
zh	Standard Chinese (Mandarin)	Sino-Tibetan	Chinese characters
yue	Yue Chinese (Cantonese)	Sino-Tibetan	Chinese characters
bo	Tibetan	Sino-Tibetan	Tibetan script
mn	Mongolian	Mongolic	Traditional Mongolian script
ug	Uyghur	Turkic	Uyghur Arabic alphabet
kk	Kazakh	Turkic	Kazakh Arabic alphabet
za	Zhuang	Kra-Dai	Latin alphabet
ko	Korean	Isolate	Hangul



[Yang et al., 2022] CINO: A Chinese Minority Pre-trained Language Model. In COLING 2022.

CINO

- **基于维基百科的分类数据集WCM**
  - 基于少数民族维基百科语料和维基分类标签体系构建的分类数据集
  - 10个标签类别（艺术、地理、历史、自然、自然科学、人物、技术、教育、经济、健康），覆盖7种语言

	训练集	开发集	测试集							
	中文	中文	蒙语	藏语	维吾尔语	哈萨克语	朝鲜语	粤语	中文	总计
数据量	32,000	3,995	2,973	1,110	300	6,258	6,558	5,943	4,000	**27,164**

- **少数民族新闻主题分类数据集CMNews**
  - 基于少数民族新闻网站数据构建，以新闻主题作为分类标签
  - 8个标签类别（教育、体育、健康、旅游、法律、经济、文化、社会），覆盖7种语言

	中文	蒙语	藏语	维吾尔语	哈萨克语	朝鲜语	粤语	总计
训练集	15,762	2,341	6,693	4,108	1,432	1,632	2,579	**35,547**
开发集	8,000	1,564	4,466	2,742	960	1,095	2,390	**21,217**

[Yang et al., 2022] CINO: A Chinese Minority Pre-trained Language Model. In COLING 2022.

CINO

- **预训练设置**
  - 数据：训练使用了14G中文语料和14G少数民族单语语料（in-house）
  - 训练：使用XLM-R模型进行初始化，并使用快速MLM任务进行预训练

	Language	# tokens	Mean sequence Length
藏语	bo	130M	13.4
哈萨克语	kk	238M	60.7
朝鲜语	ko	170M	20.0
蒙古语	mn	337M	25.7
维吾尔语	ug	1B	23.1
粤语	yue	276M	12.6
壮语	za	23M	58.1
中文	zh	1.2B	254

▲ 预训练语料统计

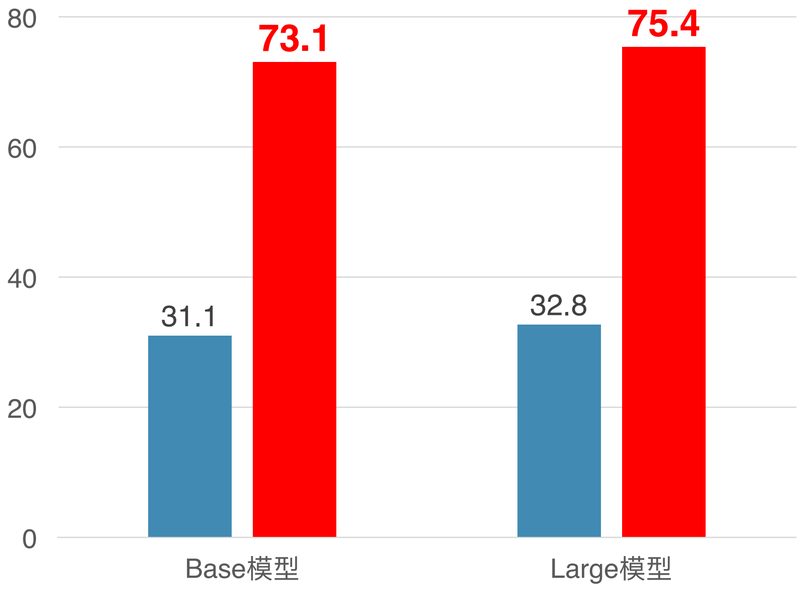
Hyperparamter	Base Model	Large Model
Batch Size	4,096	8,192
Warm-up Steps	10k	5k
Training Steps	150k	75k
Learning Rate	2e-4	1e-4
Max Length	256	256
MLM probability	0.2	0.2
Adam ϵ	1e-8	1e-8
Adam β_1	0.9	0.9
Adam β_2	0.999	0.999
Gradient Clipping	1.0	1.0
Weight Decay	0	0
Sampling α	0.7	0.7

▲ 预训练超参设置

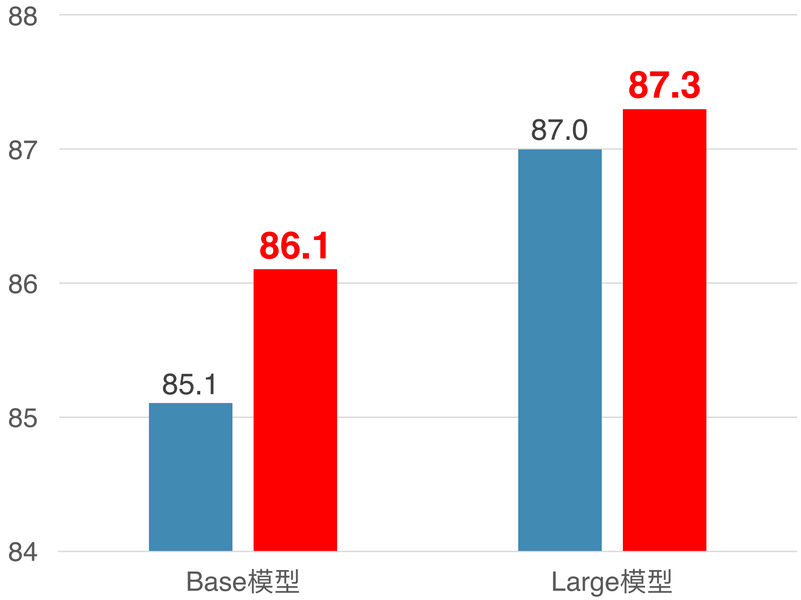
[Yang et al., 2022] CINO: A Chinese Minority Pre-trained Language Model. In COLING 2022.

CINO

- **实验结果：单语训练测试**
  - 在藏语文本分类数据集TNCC上，CINO大幅超越XLM-R效果
  - 在朝鲜语文本分类数据集YNAT上，相比XLM-R能够带来稳定性能提升



▲ 藏语文本分类TNCC (测试集结果，ACC指标)



▲ 朝鲜语文本分类YNAT（开发集结果，ACC指标）

[Yang et al., 2022] CINO: A Chinese Minority Pre-trained Language Model. In COLING 2022.

CINO

- **实验结果：跨语言性能**
  - CINO相比XLM-R模型在少数民族语言上具有更好的跨语言能力

训练：中文
测试：少数民族语言

	Model	bo	kk	ko	mn	ug	yue	zh	Avg (Minorities)	Avg (All)
WCM	*base models*									
zh → min.	XLM-R-base	19.0	16.7	43.2	15.2	23.3	58.3	78.1	29.3	36.2
	CINO-base	36.2	43.2	**44.9**	39.1	**33.4**	59.7	78.0	42.6	47.6
	large models									
	XLM-R-large	18.4	32.9	43.8	22.2	27.8	**60.0**	77.3	34.2	40.3
	CINO-large	**40.6**	**44.8**	**44.8**	**41.6**	28.8	59.8	**79.2**	**43.3**	**48.4**

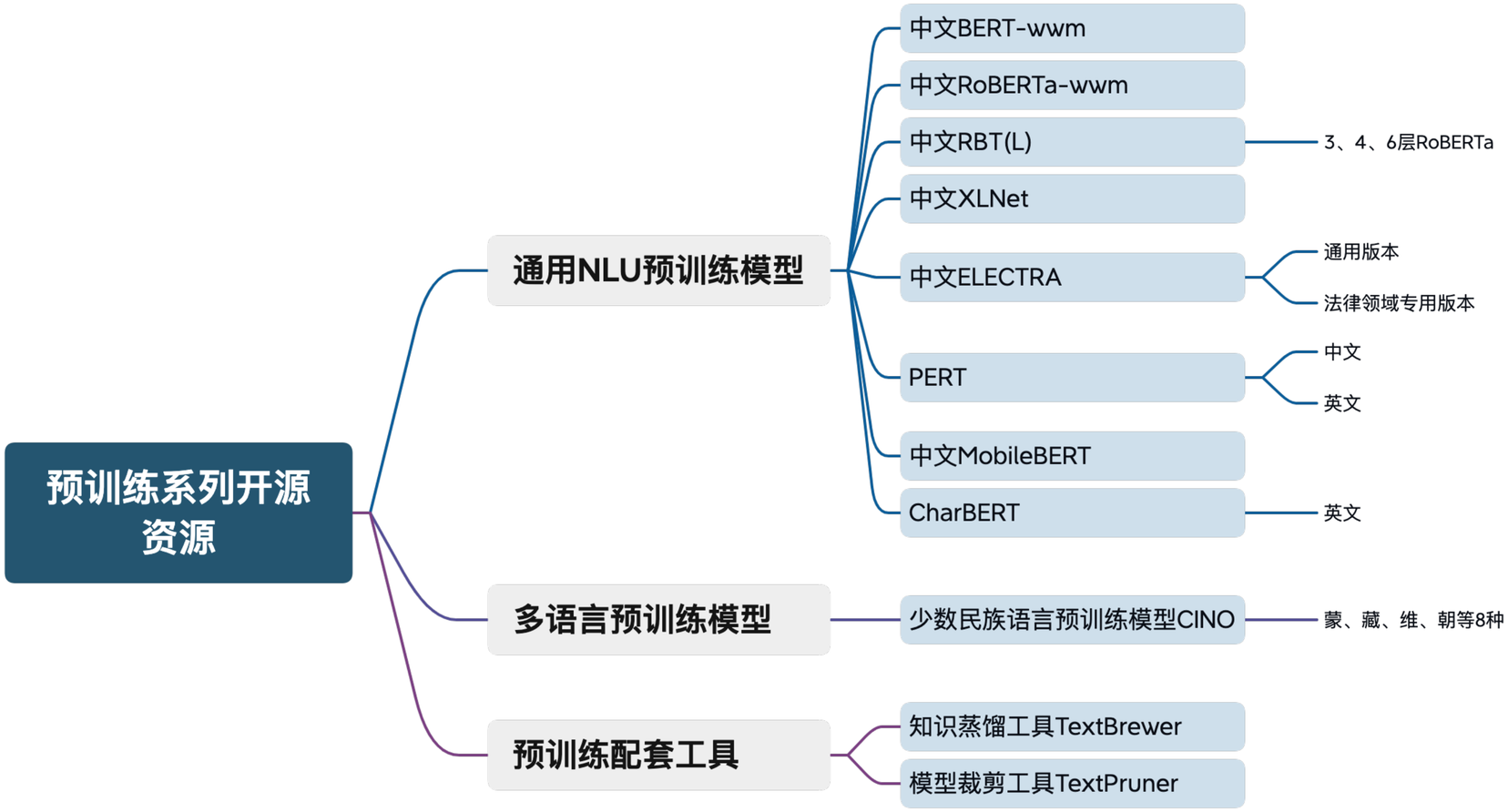
训练：少数民族语言
测试：中文

	Model	bo	kk	ko	mn	ug	yue	zh	Avg (Minorities)	Avg (All)
CMNews	*base models*									
min. → zh	XLM-R-base	38.1	69.6	88.3	35.1	77.5(67.7/88.6)	**87.8**	58.6	66.1	65.0
	CINO-base	85.5	79.2	89.0	77.3	77.4(77.0/78.0)	86.9	68.8	82.6	80.6
	large models									
	XLM-R-large	30.1	80.8	88.9	30.8	**85.1**(76.4/91.0)	87.5	63.6	67.2	66.7
	CINO-large	**86.8**	**83.0**	**90.3**	**79.4**	78.8(68.4/91.3)	**87.9**	**71.2**	**84.4**	**82.5**

[Yang et al., 2022] CINO: A Chinese Minority Pre-trained Language Model. In COLING 2022.

中文预训练开源资源

- **构建预训练系列开源资源**
  - 已开源多种类型的预训练相关资源，在GitHub开源社区star数量突破**10,000+**

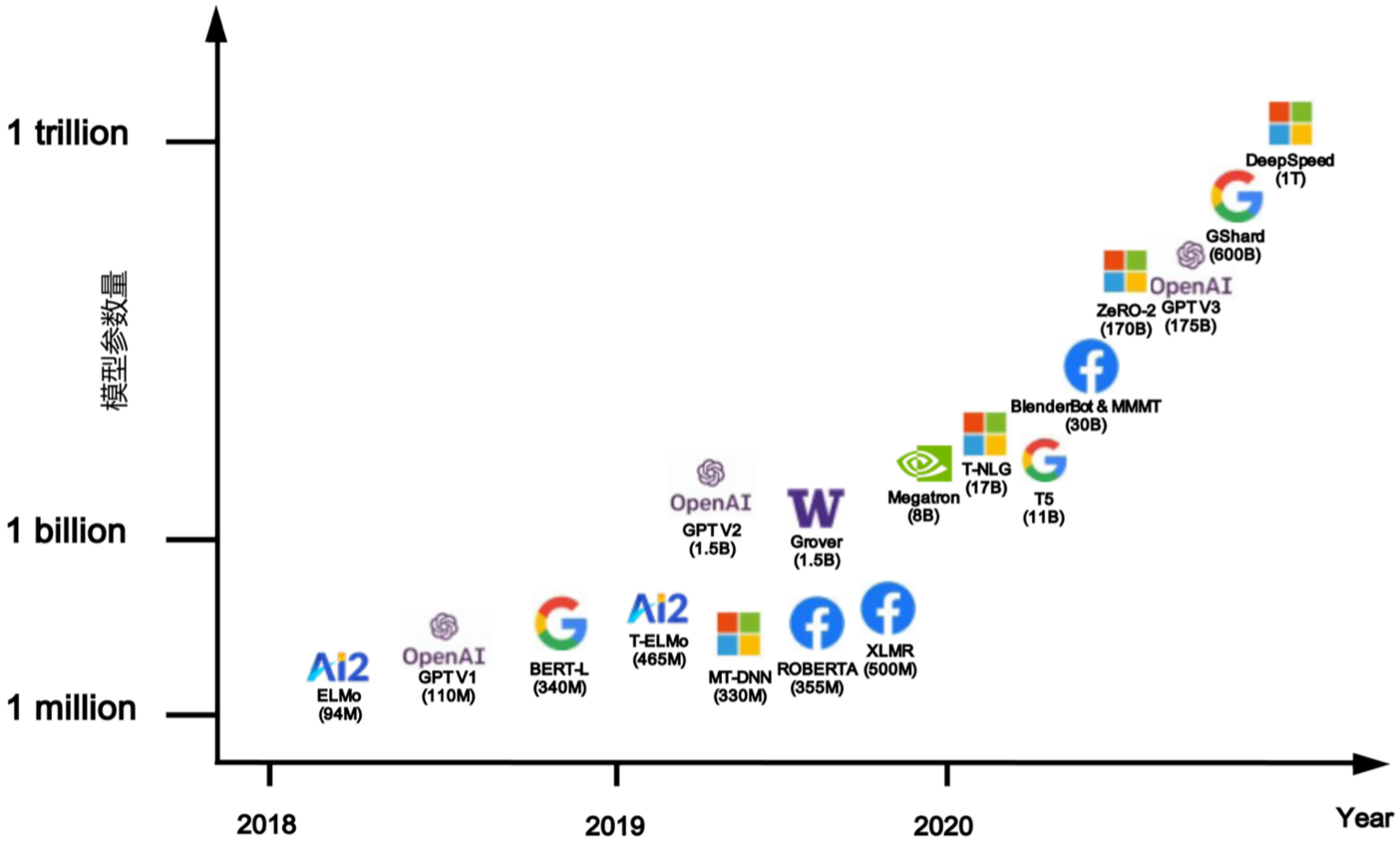


预训练模型的蒸馏与裁剪

Distillation and Pruning for Pre-trained Language Models

预训练模型的蒸馏与裁剪

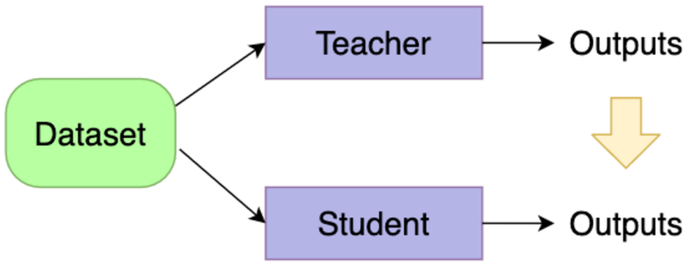
- **研究背景**
  - 预训练模型通常需要占用很大的空间，并且训练和推断时间也很慢，难以满足实际应用需求



预训练模型的蒸馏与裁剪

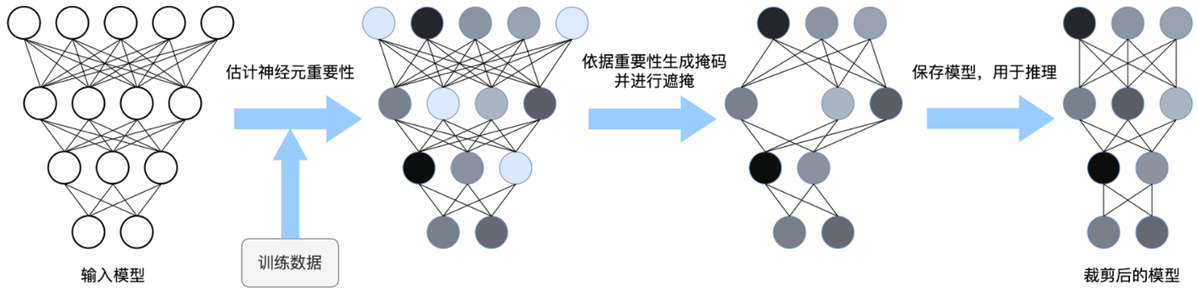
- **高效模型训练与推理的主要途径**
  - **知识蒸馏**：在少量性能损失的情况下，将大模型知识迁移到小模型，提升模型效果和推断速度
  - **模型裁剪**：对模型中的部分“不重要”或“冗余”的部分进行剪枝，从而缩小预训练模型体积

知识蒸馏



TextBrewer

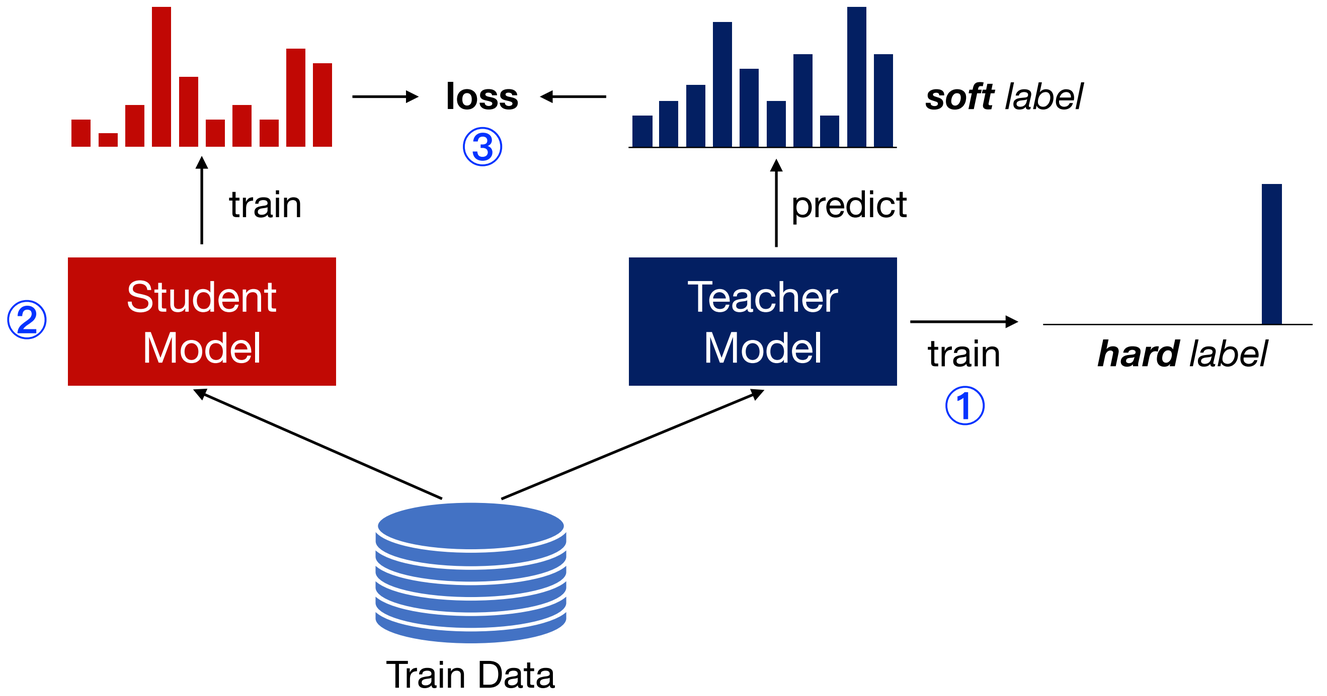
模型裁剪



TextPruner

知识蒸馏

- **知识蒸馏是什么?**
  - 在少量性能损失的情况下，将大模型知识迁移到小模型，可以看作**“老师”教“学生”的过程**
  - 知识蒸馏技术经常被用于减小模型体积、提升模型效果和推断速度



① 用数据训练一个教师模型

② 定义并初始化一个学生模型

③ 利用合适的训练损失，让学生模型学习教师模型

TextBrewer

- **TextBrewer: An Open-Source Knowledge Distillation Toolkit for NLP**
  - 推出了首个面向自然语言处理领域的基于PyTorch的知识蒸馏工具包
  - 提供了方便、快捷、易用的知识蒸馏框架，少量性能损失换取大幅速度提升
  - 访问 http://textbrewer.hfl-rc.com/ 或通过 `pip install textbrewer`安装

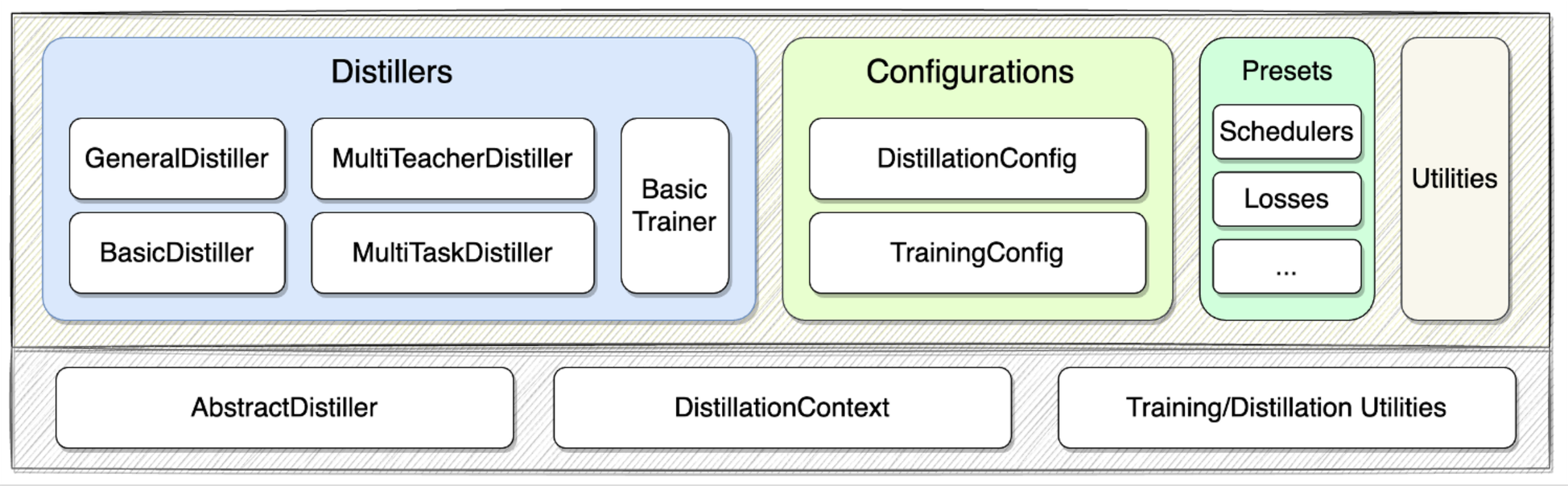
TextBrewer 1200+ ⭐

- **模型无关**：适用于多种模型结构（主要面向Transfomer结构）
- **方便灵活**：可自由组合多种蒸馏方法，支持增加自定义损失等模块
- **非侵入式**：无需对教师与学生模型本身结构进行修改
- **适用面广**：支持典型NLP任务，如文本分类、阅读理解、序列标注等

[Yang et al., 2020] TextBrewer: An Open-Source Knowledge Distillation Toolkit for Nature Language Processing. In ACL 2020.

TextBrewer

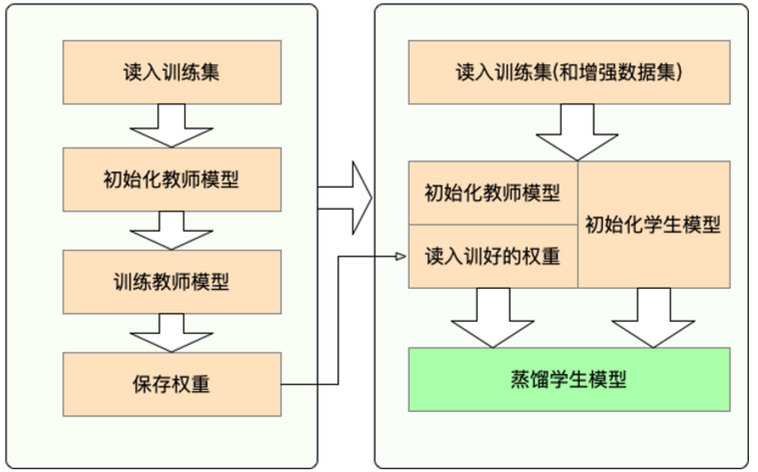
- **工具包设计架构**
  - Distillers：用于执行实际的知识蒸馏工作，定义了常规蒸馏策略
  - Configurations：为Distillers提供必要的配置信息
  - Utilities：包含一些辅助的功能，如模型参数统计等



[Yang et al., 2020] TextBrewer: An Open-Source Knowledge Distillation Toolkit for Nature Language Processing. In ACL 2020.

TextBrewer

- 工作流程
  - **第一步：蒸馏之前的准备工作**
    - 训练教师模型，定义并初始化学生模型
    - 构造蒸馏用数据集的DataLoader
  - **第二步：知识蒸馏**
    - 初始化Distiller，构造训练配置和蒸馏配置
    - 定义adaptors和callback，分别用于适配模型输入输出和训练过程中的回调
    - 调用Distiller的train方法开始蒸馏
  - 蒸馏已经预设的相关任务，只需要20行左右的代码！




```python
from textbrewer import GeneralDistiller
from textbrewer import TrainingConfig, DistillationConfig

# We omit the initialization of models, optimizer, and dataloader.
teacher_model : torch.nn.Module = ...
student_model : torch.nn.Module = ...
dataloader : torch.utils.data.DataLoader = ...
optimizer : torch.optim.Optimizer = ...
scheduler : torch.optim.lr_scheduler = ...

def simple_adaptor(batch, model_outputs):
    # We assume that the first element of model_outputs
    # is the logits before softmax
    return {'logits': model_outputs[0]}

train_config = TrainingConfig()
distill_config = DistillationConfig()
distiller = GeneralDistiller(
    train_config=train_config, distill_config = distill_config,
    model_T = teacher_model, model_S = student_model,
    adaptor_T = simple_adaptor, adaptor_S = simple_adaptor)

distiller.train(optimizer, scheduler,
    dataloader, num_epochs, callback=None)
```


[Yang et al., 2020] TextBrewer: An Open-Source Knowledge Distillation Toolkit for Nature Language Processing. In ACL 2020.

TextBrewer

- **知识蒸馏效果**
  - 教师模型：BERT-base (110M)
  - 学生模型
    - T6 (60%), T3 (41%), T3-small (16%), T4-tiny (same as TinyBERT, 13%)
  - 单教师知识蒸馏
    - T6结构可以达到教师模型效果的99%，模型体积缩小至60%
    - T4-tiny知识蒸馏结果优于TinyBERT
  - 多教师知识蒸馏
    - 蒸馏后的学生模型获得最优效果，且超过简单的模型融合方法（ensemble）

Model	MNLI m	MNLI mm	SQuAD EM	SQuAD F1	CoNLL-2003 F1
$\text{BERT}_{\text{BASE}}$	83.7	84.0	81.5	88.6	91.1
Public					
DistilBERT	81.6	81.1	79.1	86.9	-
TinyBERT	80.5	81.0	-	-	-
+DA	82.8	82.9	72.7	82.1	-
TextBrewer					
BiGRU	-	-	-	-	85.3
T6	83.6	84.0	80.8	88.1	90.7
T3	81.6	82.5	76.3	84.8	87.5
T3-small	81.3	81.7	72.3	81.4	78.6
T4-tiny	82.0	82.6	73.7	82.5	77.5

▲ 单教师蒸馏效果

Model	MNLI m	MNLI mm	SQuAD EM	SQuAD F1	CoNLL-2003 F1
Teacher 1	83.6	84.0	81.1	88.6	91.2
Teacher 2	83.6	84.2	81.2	88.5	90.8
Teacher 3	83.7	83.8	81.2	88.7	91.3
Ensemble	84.3	84.7	82.3	89.4	91.5
Student	**84.8**	**85.3**	**83.5**	**90.0**	**91.6**

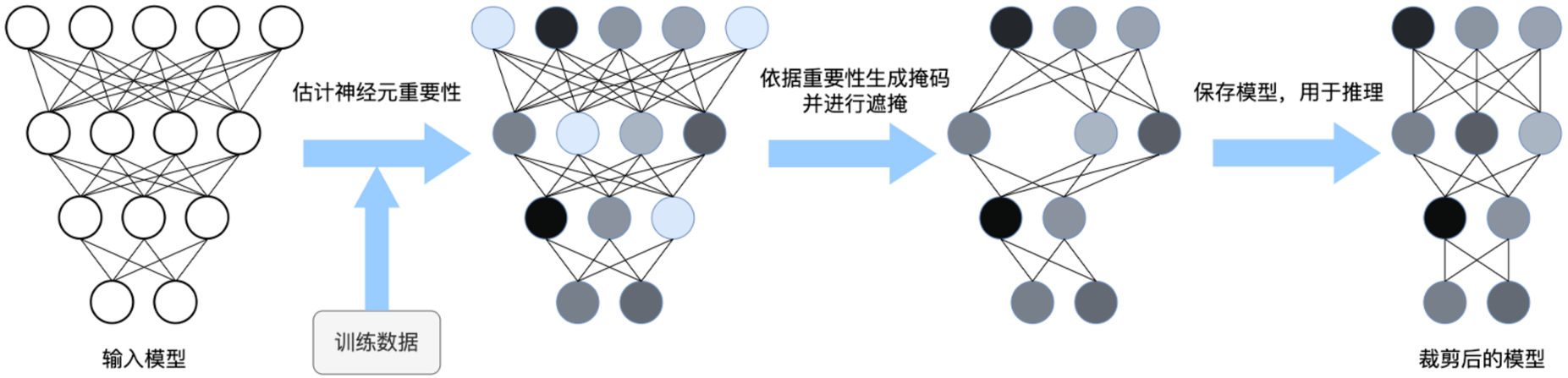
▲ 多教师蒸馏效果

[Yang et al., 2020] TextBrewer: An Open-Source Knowledge Distillation Toolkit for Nature Language Processing. In ACL 2020.

模型裁剪

- **模型裁剪是什么?**
  - 对模型中的部分“不重要”或“冗余”的权重或网络结构进行剪枝，从而缩小预训练模型体积、提升推理速度
    - 与知识蒸馏等技术正交，可相互叠加使用
    - （与知识蒸馏相比）无需设计学生模型，小模型的结构由算法得到

	非结构化裁剪	结构化裁剪
裁剪粒度	单个神经元	整个矩阵，或其整行整列
加速设备	需要特殊硬件	大多数计算硬件
压缩率	较高	中等
权重稀疏性	稀疏矩阵	稠密矩阵



TextPruner

- **TextPruner: A Model Pruning Toolkit for Pre-trained Language Model**
  - 推出首个面向自然语言处理领域的基于PyTorch的模型裁剪工具包
  - 通过轻量、快速的裁剪方法对模型进行结构化剪枝，从而实现压缩模型体积、提升模型速度
  - 访问 http://textpruner.hfl-rc.com/ 或通过 `pip install textpruner`安装

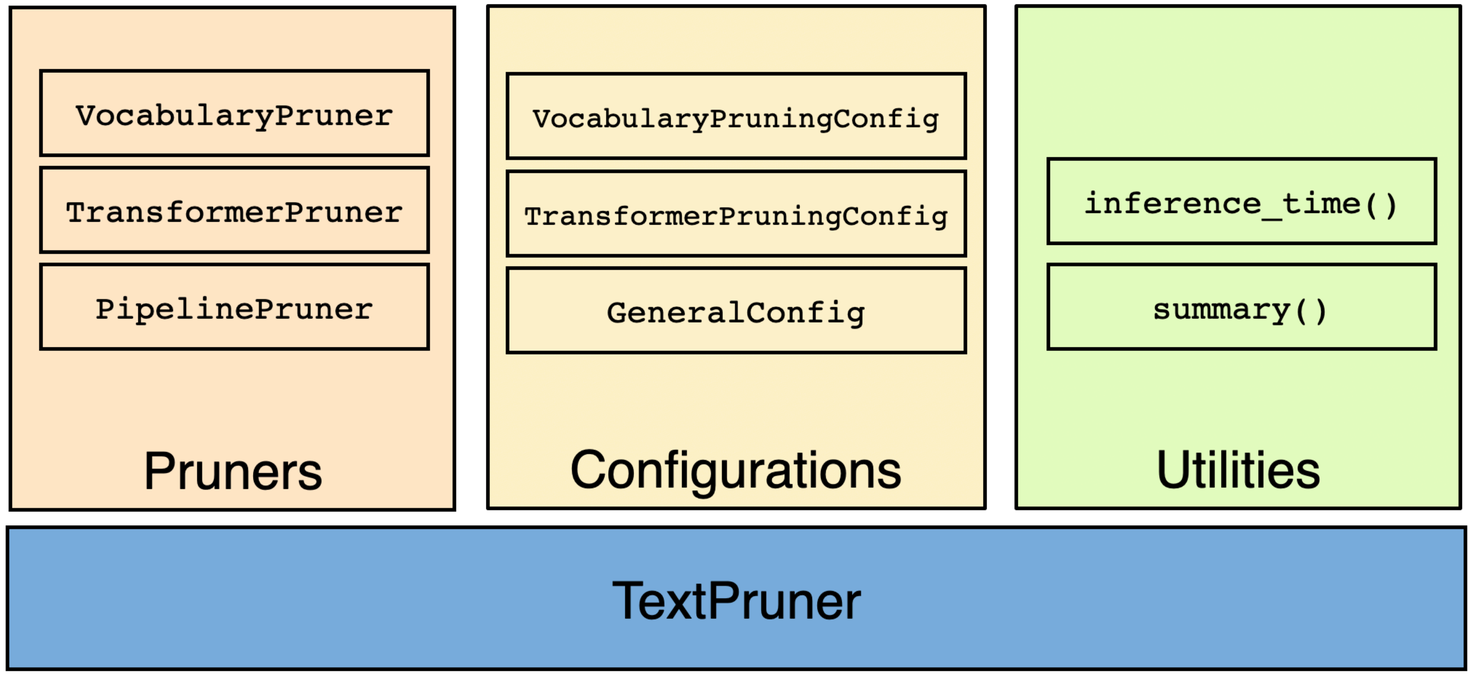
TextPruner 190+ ⭐

- **功能通用：**适配多种预训练模型，以及多种NLU任务
- **可定制化：**除了标准PLM外，也可使用TextPruner裁剪基于标准PLM开发的自定义模型
- **灵活便捷：**可作为Python包在Python脚本中使用，也提供了独立的命令行工具
- **运行高效：**使用无训练的结构化裁剪方法，运行迅速，快于知识蒸馏等基于训练的方法

[Yang et al., 2022] TextPruner: A Model Pruning Toolkit for Pre-trained Language Models. In ACL 2022.

TextPruner

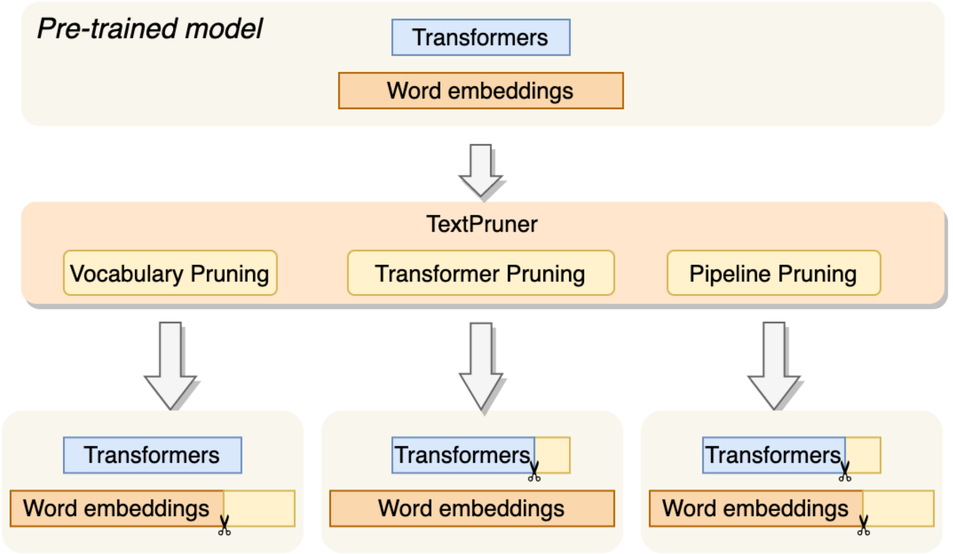
- **工具包设计架构**
  - Pruners：用于执行实际的模型裁剪，包含词表裁剪、Transformer裁剪、流水线裁剪等
  - Configurations：为Distillers提供必要的配置信息
  - Utilities：包含一些辅助的功能，如模型参数统计、计算推断时间等



[Yang et al., 2022] TextPruner: A Model Pruning Toolkit for Pre-trained Language Models. In ACL 2022.

TextPruner

- **裁剪模式**
  - **词表裁剪**：移除词表中未被使用的单词，实现减小模型体积，提升MLM任务速度的效果
  - **Transformer裁剪**：裁剪“不重要”的注意力头和全连接层神经元，在减小模型体积的同时把对模型性能的影响尽可能降到最低
  - **流水线裁剪**：依次分别进行Transformer裁剪和词表裁剪，对模型体积做全面的压缩



☞ 计算“重要性” $\mathrm{IS}(\mathbf{\Theta}) = \mathbb{E}_{x\sim X}\left|\frac{\partial \mathcal{L}(x)}{\partial \mathbf{\Theta}}\mathbf{\Theta}\right|$

☞ **有**监督裁剪 $\mathscr{L}_{CE}(x) = -\sum_{x} y \log q(x)$

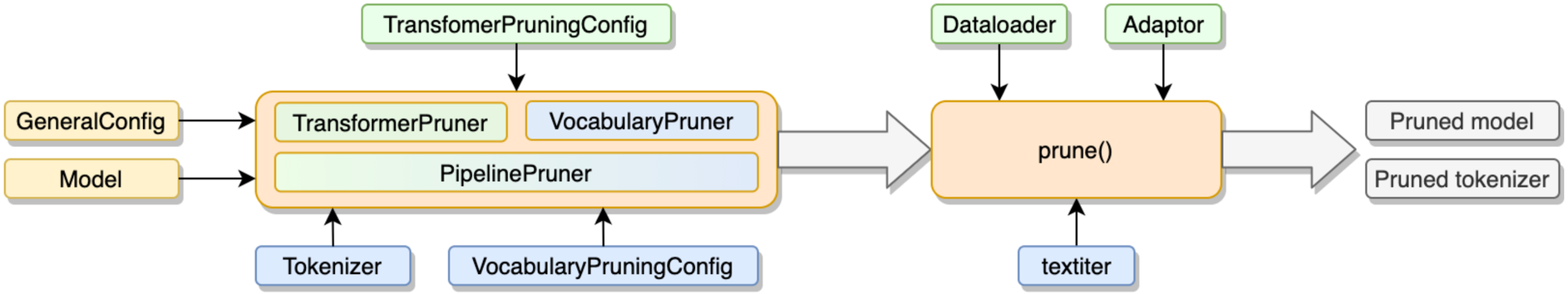
☞ **无**监督裁剪 $\mathscr{L}_{KL}(x) = \mathbf{KL}(\mathbf{stopgrad}(q(x))||p(x))$

原模型 待裁剪模型

[Yang et al., 2022] TextPruner: A Model Pruning Toolkit for Pre-trained Language Models. In ACL 2022.

TextPruner

- 裁剪方法



Pruning with Python API

```
from textpruner import PipelinePruner
from textpruner import TransformerPruningConfig

config = TransformerPruningConfig(
    pruning_method='iterative', n_iters=4,
    target_ffn_size=2048,target_num_of_heads=8)
pruner = PipelinePruner(model, tokenizer,config)
pruner.prune(dataloader=dataloader, dataiter=texts)
```

▲ 通过Python API裁剪

Pruning with CLI

```
textpruner-cli \
 --pruning_mode pipeline \
 --configurations vc.json trm.json \
 --model_class BertForClassification \
 --tokenizer_class BertTokenizer \
 --model_path models \
 --vocabulary texts.txt \
 --dataloader_and_adaptor dl.py
```

▲ 通过命令行裁剪

[Yang et al., 2022] TextPruner: A Model Pruning Toolkit for Pre-trained Language Models. In ACL 2022.

TextPruner

- **模型裁剪效果：词表裁剪**
  - 实验设置
    - 实验以多语言预训练模型XLM-R在自然语言推断任务XNLI为基准进行测试
    - 使用英文数据训练，使用中文数据测试 (zero-shot)
    - 词表裁剪后，预训练模型只保留对应语种的单词
  - 实验结果表明，缩减词表后模型体积减小60%左右，基本保持了与基线系统持平的效果
  - 在指定语种的情况下，词表裁剪特别适合应用在多语言预训练模型

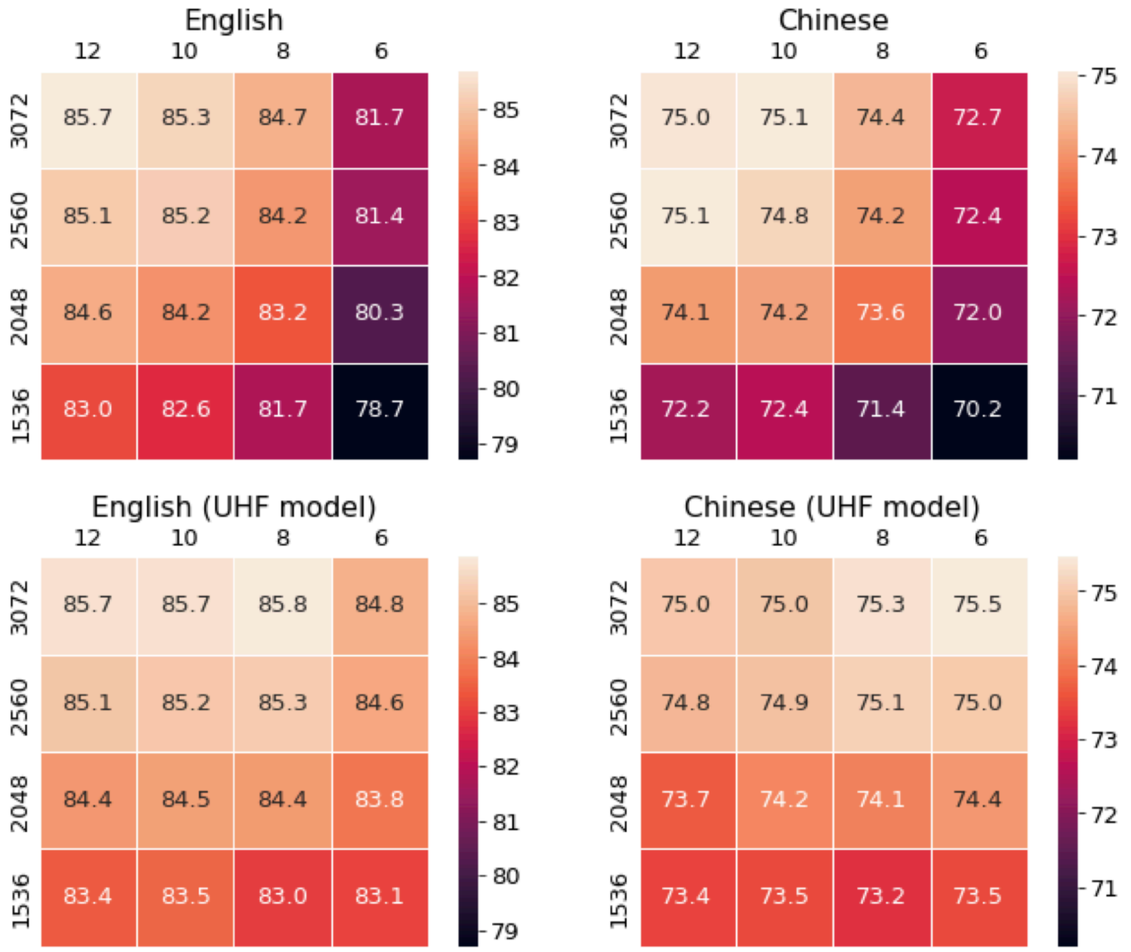
	Model	Vocabulary size	Model size	Dev (en)	Dev (zh)	Test (en)	Test (zh)
基模型	XLM-R	250002	1060 MB (100%)	84.8	75.1	85.7	75.0
裁剪模型	+ Vocabulary Pruning on en	26653	406 MB (38.3%)	84.6	-	85.9	-
	+ Vocabulary Pruning on zh	23553	397 MB (37.5%)	-	74.7	-	74.5
	+ Vocabulary Pruning on en and zh	37503	438 MB (41.3%)	84.8	74.3	85.8	74.5

▲ 词表裁剪效果

[Yang et al., 2022] TextPruner: A Model Pruning Toolkit for Pre-trained Language Models. In ACL 2022.

TextPruner

- **模型裁剪效果：Transformer裁剪**
  - 使用XNLI英文开发集计算Importance Scores
  - 两种裁剪方法
    - UHF (**U**neven attention **H**eads and **F**FN sizes)
    - HP (**H**omogenous **P**runing)
  - 观察结论
    - UHF相比HP具有更大的灵活度，能够获得更好的裁剪效果
    - 中文测试集（zero-shot）上的性能损失与英文相近
      - 对中文任务重要的神经元仍然被保留
      - 提供通用跨语言理解能力的神经元被保留

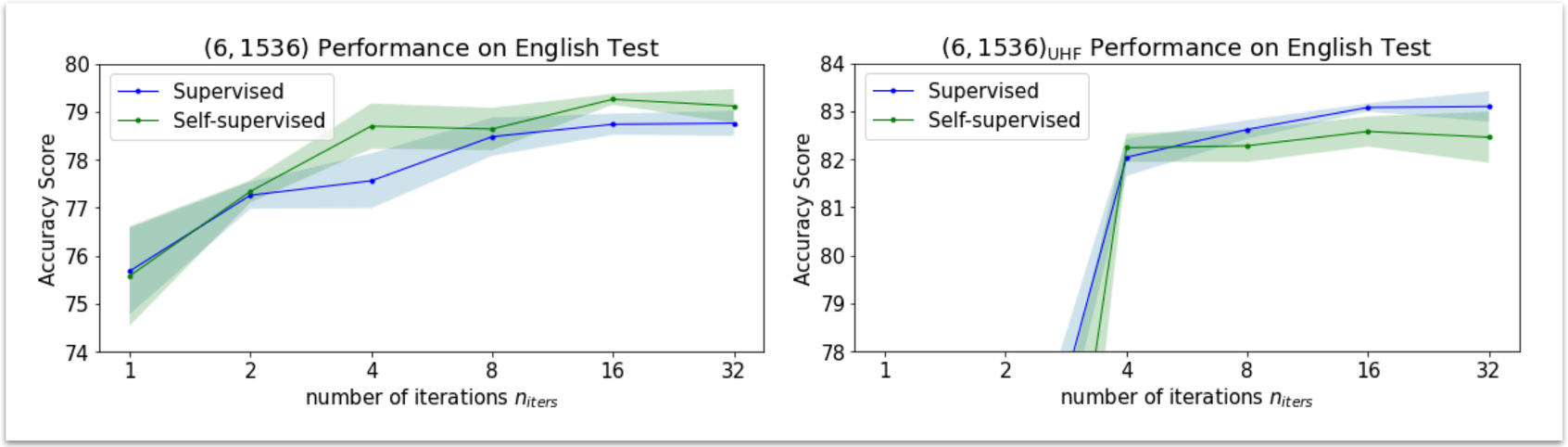


▲ XNLI测试集结果，横轴：平均注意力头数，
纵轴：FFN大小

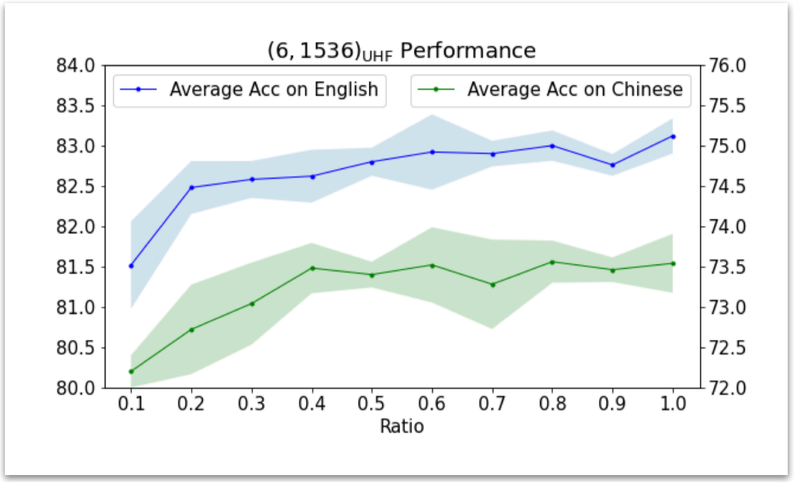
[Yang et al., 2022] TextPruner: A Model Pruning Toolkit for Pre-trained Language Models. In ACL 2022.

TextPruner

- **分析①：有监督 v.s. 无监督**
  - 自监督方法能够在不利用标签的情况下达到与有监督方法可比甚至更优的实验结果
  - 为了保证IS计算准确，建议迭代轮数不低于8轮
- **分析②：需要多少数据来计算IS?**
  - 当使用70%的开发集用于计算IS，其效果与使用全量开发集达到可比状态



▲ 有监督v.s.无监督



▲ 用于计算IS的开发集数量

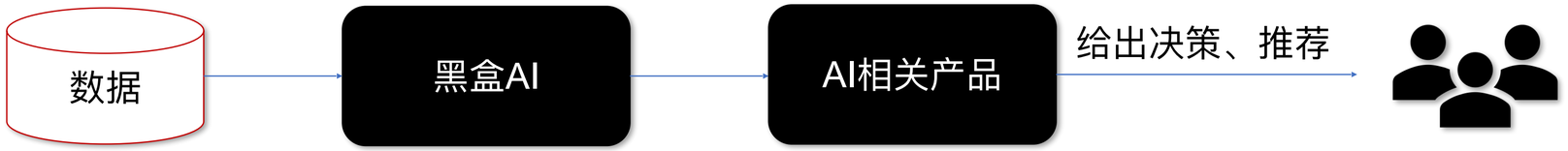
[Yang et al., 2022] TextPruner: A Model Pruning Toolkit for Pre-trained Language Models. In ACL 2022.

预训练语言模型的可解释性

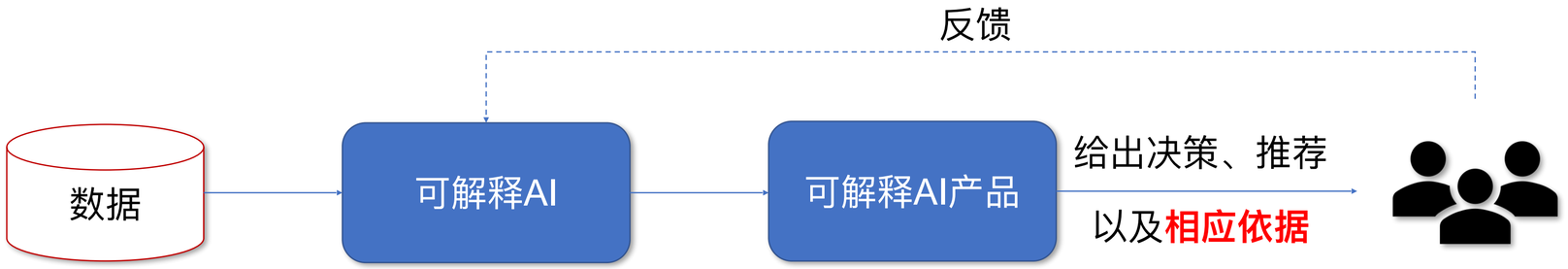
Explainability of Pre-trained Language Model

预训练语言模型的可解释性

- **黑盒AI和可解释AI**
  - **黑盒AI**：为何作出这种决策，依据是？什么时候会导致方法失效？如何调整和修改决策过程？

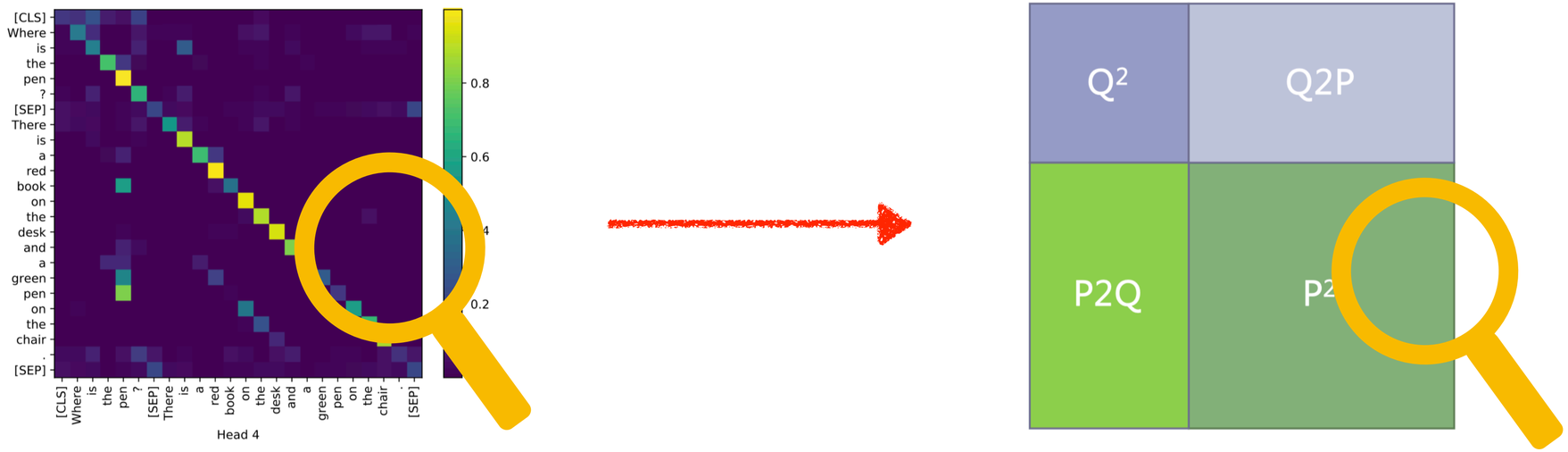


  - **可解释AI**：同时给出决策和相应依据；可以知道AI方法在什么情况下有效；决策结果可控可反馈；用户信赖度更高



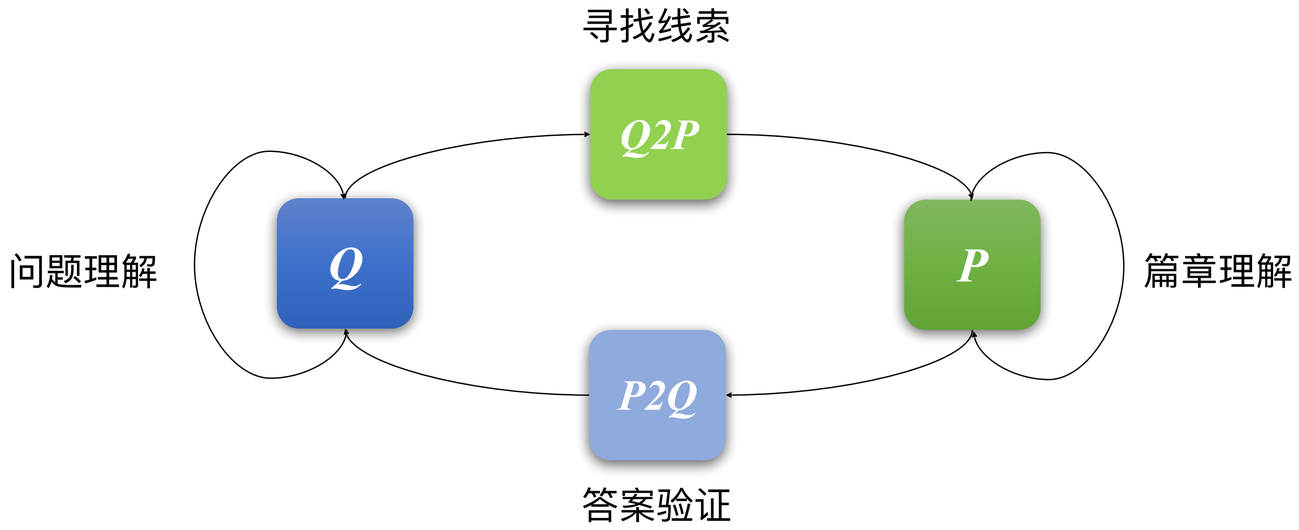
预训练语言模型的可解释性

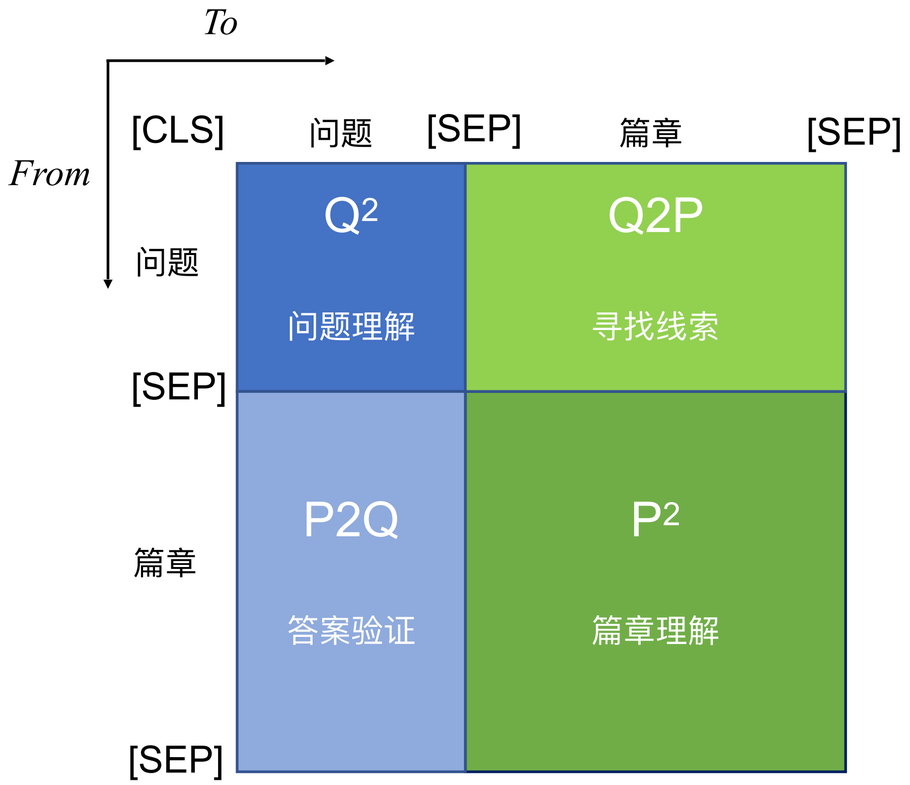
- **进一步理解注意力机制**
  - 注意力机制成为了NLP模型中不可或缺的组成部分，常被认为是可解释性的来源
  - 那么应该如何看待可解释性和注意力机制的关系？现有对注意力矩阵的分析存在什么问题?
  - 以机器阅读理解任务为例，对注意力机制进行深入剖析



预训练语言模型的可解释性

- **机器阅读理解中的注意力机制**
  - 机器阅读理解任务的常规输入形式： [CLS] Question [SEP] Passage [SEP]
  - 提出将注意力矩阵按篇章P和问题Q进行划分，形成4个注意力区域：Q^2, Q2P, P2Q, P^2

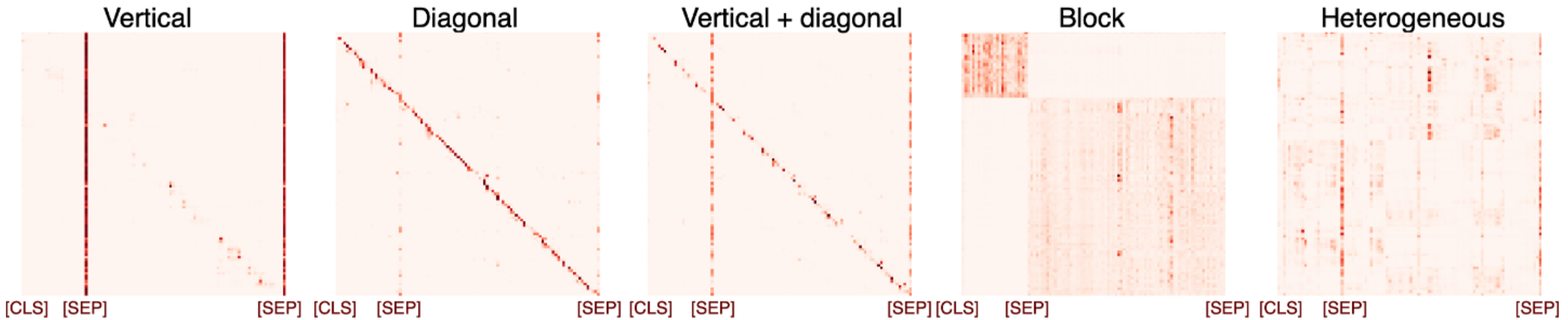




[Cui et al., 2022] Multilingual Multi-aspect Explainability Analyses on Machine Reading Comprehension. In iScience Vol.25.

预训练语言模型的可解释性

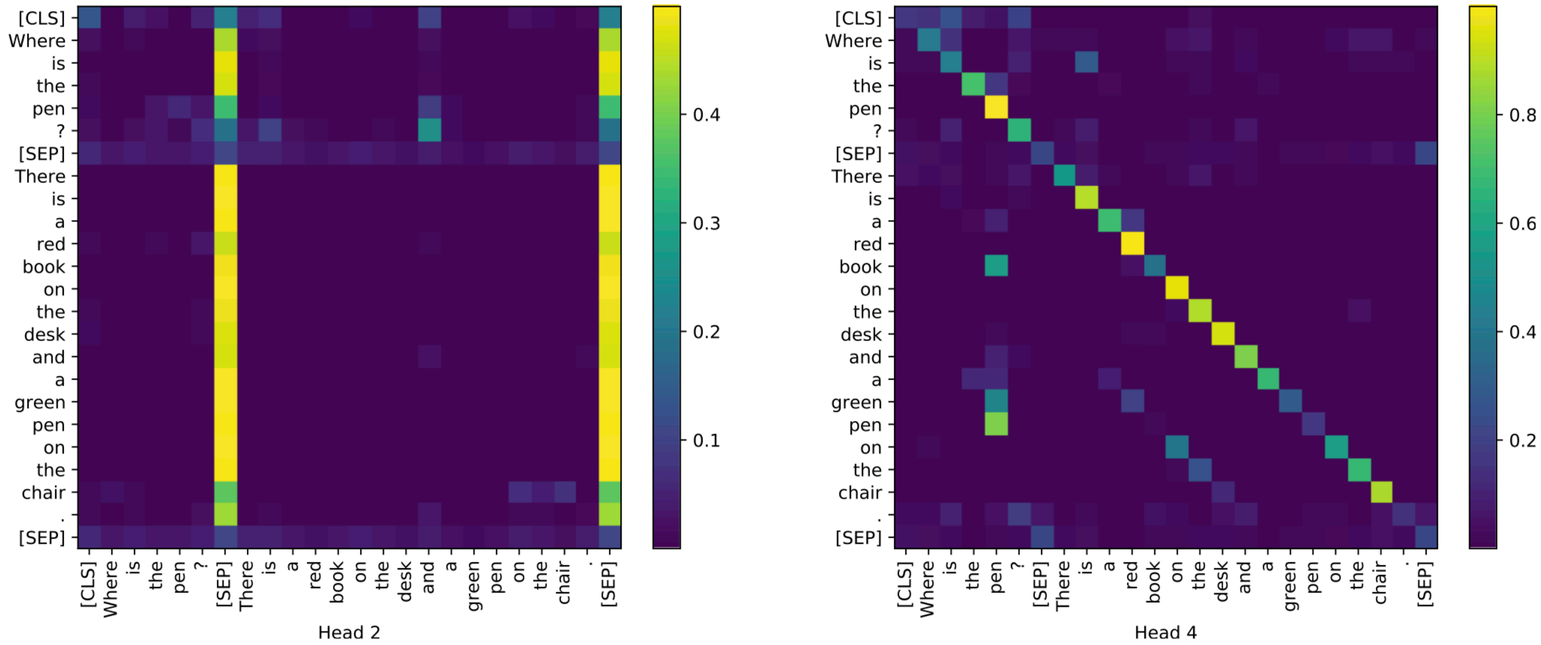
- **研究背景**
  - 早期研究（Kovaleva et al., 2019）表明，预训练模型中主要有5种自注意力模式
  - 可视化结果表明
    - 在任何模式下，特殊符号[CLS]和[SEP]对应的注意力值较高
    - 在部分模式下，对角线元素的注意力值较高



[Kovaleva et al., 2019] Revealing dark secrets of BERT. In EMNLP 2019.

预训练语言模型的可解释性

- **注意力值越高对性能影响更大?**
  - 在机器阅读理解任务中，注意力矩阵也有类似的现象
  - 直觉认为“注意力高”的元素可能对模型输出影响更大，但事实是否如此?



▲ 可视化经过SQuAD训练的模型

[Cui et al., 2022] Multilingual Multi-aspect Explainability Analyses on Machine Reading Comprehension. In iScience Vol.25.
[Cui et al., 2022] Visualizing Attention Zones on Machine Reading Comprehension Models. In STAR Protocols Vol.3.

预训练语言模型的可解释性

- **屏蔽不同注意力区域后的实验结果①**
  - 单独去掉特殊符号[CLS], [SEP]对模型效果的影响并不大，甚至能够提升模型效果
  - 同时去掉所有特殊符号时，对模型效果影响较大

Table 1. Results on masking part of attention map

	SQuAD		CMRC 2018	
	EM	F1	EM	F1
Baseline	80.687	88.129	63.796	84.789
No [CLS]	80.802	88.276	64.119	84.858
No Mid [SEP]	80.689	88.082	63.896	84.626
No End [SEP]	80.522	87.959	64.299	84.866
No All	78.956	86.414	63.659	83.945
No Diagonal	80.645	88.241	64.548	84.908
No Q^2	76.395	84.195	60.100	80.625
No Q2P	79.941	87.352	64.517	84.592
No P2Q	12.763	16.355	15.070	18.466
No P^2	34.441	51.792	16.278	42.906

[Cui et al., 2022] Multilingual Multi-aspect Explainability Analyses on Machine Reading Comprehension. In iScience Vol.25.

预训练语言模型的可解释性

- **屏蔽不同注意力区域后的实验结果②**
  - 去掉对角线元素时，模型效果反而会得到进一步提升
  - 不同注意力区域对模型效果的影响程度存在较大差异，因此对于注意力的分析应进一步细化

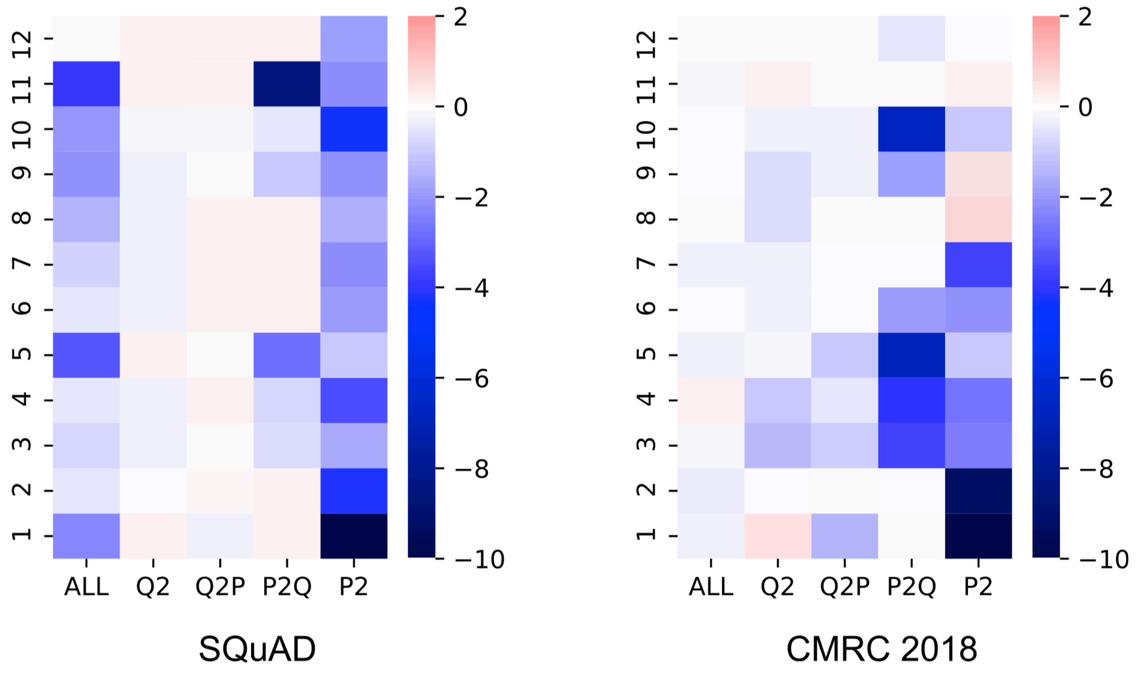
Table 1. Results on masking part of attention map

	SQuAD		CMRC 2018	
	EM	F1	EM	F1
Baseline	80.687	88.129	63.796	84.789
No [CLS]	80.802	88.276	64.119	84.858
No Mid [SEP]	80.689	88.082	63.896	84.626
No End [SEP]	80.522	87.959	64.299	84.866
No All	78.956	86.414	63.659	83.945
No Diagonal	80.645	88.241	64.548	84.908
No Q^2	76.395	84.195	60.100	80.625
No Q2P	79.941	87.352	64.517	84.592
No P2Q	12.763	16.355	15.070	18.466
No P^2	34.441	51.792	16.278	42.906

[Cui et al., 2022] Multilingual Multi-aspect Explainability Analyses on Machine Reading Comprehension. In iScience Vol.25.

预训练语言模型的可解释性

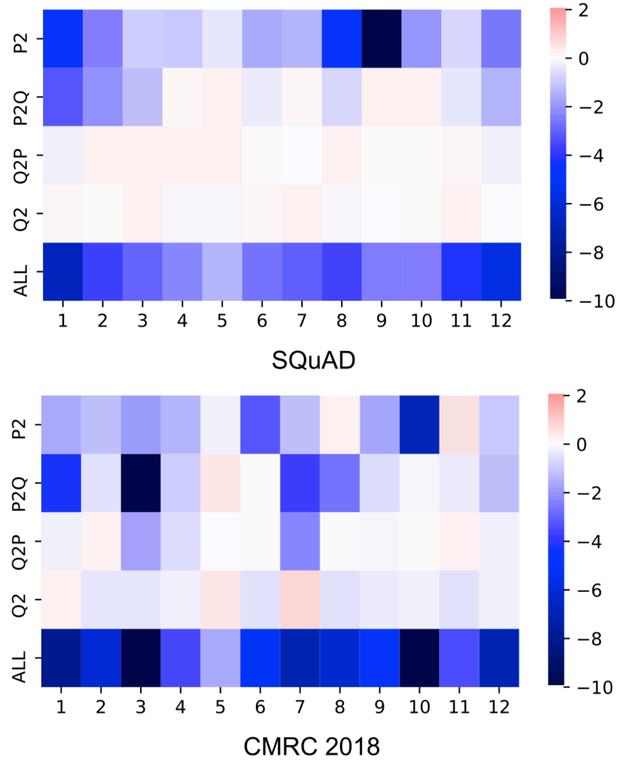
- **屏蔽各层中的特定注意力区域**
  - 将每层特定注意力区域进行屏蔽，在开发集上解码得出EM指标，将其与基线差值进行可视化
  - 不论在英文还是中文任务上，P2Q和P2区域对模型性能影响最大

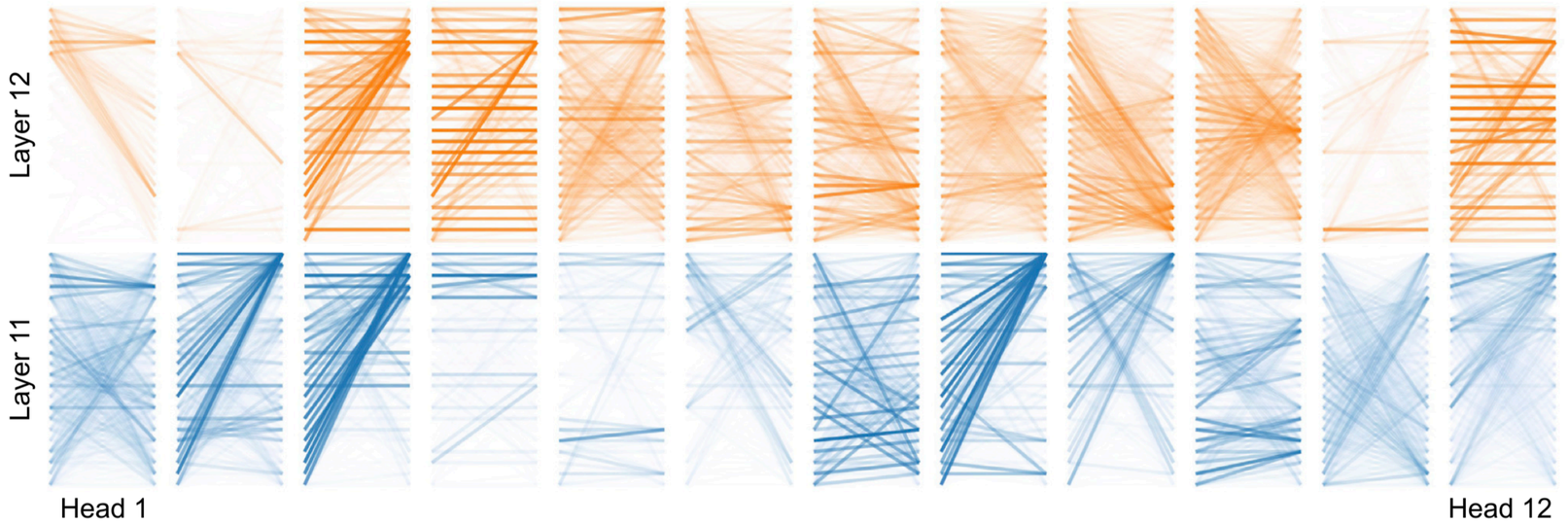


[Cui et al., 2022] Multilingual Multi-aspect Explainability Analyses on Machine Reading Comprehension. In iScience Vol.25.

预训练语言模型的可解释性

- **屏蔽各注意力头中的特定注意力区域**
  - 屏蔽所有层中某个特定注意力头，观察其效果变化
  - 整体上看，还是P2Q和P^2区域对模型性能的影响最大
  - 不同的注意力头并没有一致的变化趋势，因此不应观察某个特定注意力头在所有层中的表现

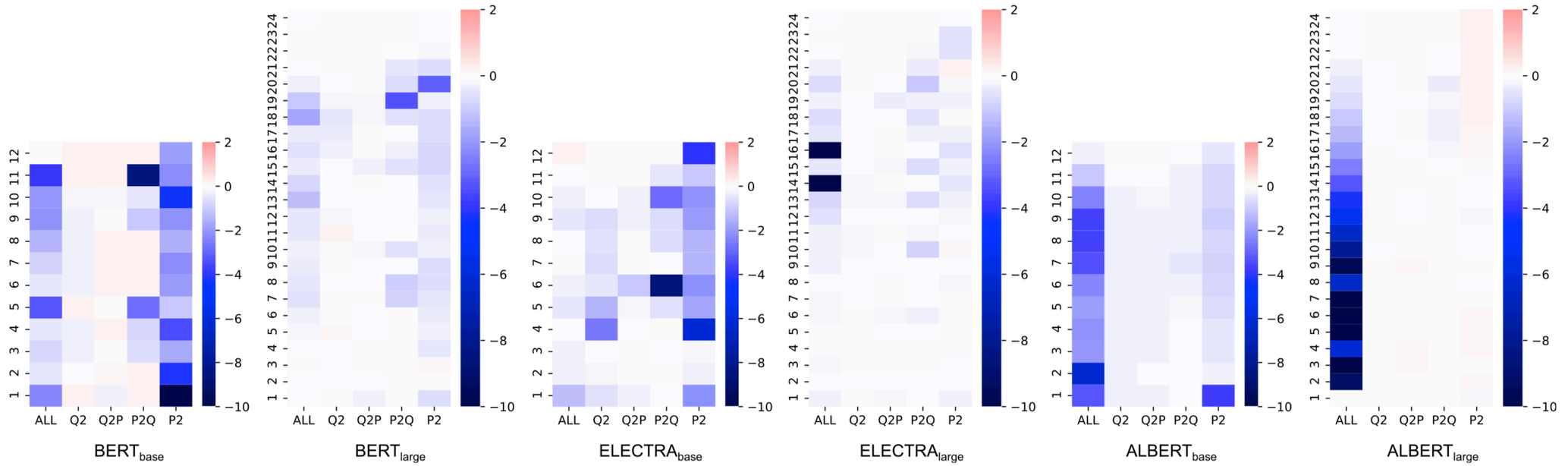




[Cui et al., 2022] Multilingual Multi-aspect Explainability Analyses on Machine Reading Comprehension. In iScience Vol.25.

预训练语言模型的可解释性

- 分析不同预训练模型的注意力行为模式
  - P2Q和P²仍然是对结果影响最重要的注意力区域
  - large模型相比base模型具有较高的容错性，且相关知识学习的分布相对均匀
  - ALBERT的跨层参数共享机制在注意力分布中得以体现（从某种程度上解释了这种现象）



[Cui et al., 2022] Multilingual Multi-aspect Explainability Analyses on Machine Reading Comprehension. In iScience Vol.25.

总结

SUMMARY

总结

- **通用预训练语言模型**
  - 经典预训练模型BERT以及MLM变种技术，常读常新
  - 解决“预训练-精调”不一致问题：基于纠错型掩码的预训练模型MacBERT
  - 非MLM预训练任务尝试：基于乱序语言模型训练的PERT
  - 预训练模型在稀缺资源语种上的尝试：面向少数民族语言的预训练模型CINO
- **面向预训练语言模型的知识蒸馏与压缩**
  - 知识蒸馏工具TextBrewer、模型裁剪工具TextPruner，助力模型的高效训练与推理
- **预训练模型的可解释性**
  - Attention机制具备一定的可解释性，但应与具体任务、语种、预训练模型相结合进行分析

相关资源

GitHub

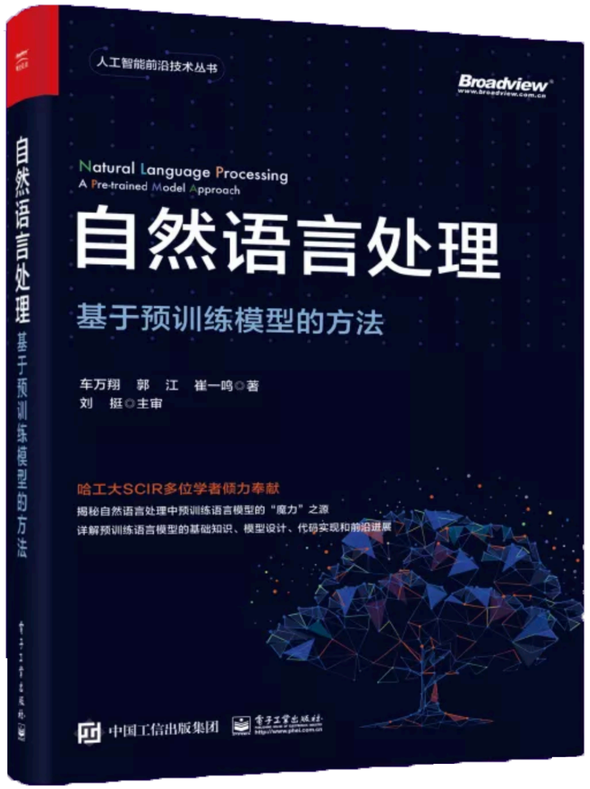
- 预训练模型：BERT、RoBERTa、RBT(L)、XLNet、ELECTRA、MacBERT、PERT、少数民族语言模型CINO
- 开源工具包：TextBrewer、TextPruner
- 中文数据集：阅读理解、文本分类、文本纠错等
- 访问HFL典藏集了解更多：http://anthology.hfl-rc.com

Model Hub

- 提供了9大类，共计44个优质预训练模型
- 搭配🤗transformers库，快速加载各类预训练模型
- 模型库地址：http://huggingface.co/hfl

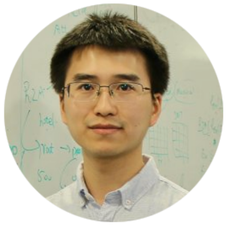
哈工大讯飞联合实验室
微信公众号

预训练书



车万翔

哈尔滨工业大学教授



郭江

麻省理工学院博士后

崔一鸣

科大讯飞资深科学家

适合哪些读者阅读?

- 高等院校本科生及研究生
- 研究机构的研究者
- 希望深入研究NLP算法的计算机工程师

配套资源齐备

- 实践案例源代码
- 课后习题答案
- 配套PPT

THANK YOU!

https://github.com/ymcui

https://ymcui.com

me@ymcui.com

讲义下载回复：
smp2022

哈工大讯飞联合实验室

科大讯飞 iFLYTEK