中国科学技术大学人工智能实践课程
技术报告

基于 BERT 和 RoBERTa 的中文文本二分类

朱云沁 PB20061372

中国科学技术大学

2024 年 1 月

目录

1 实验目标

训练任务为文本分类. 数据集是标签为“0”和“1”的两类文本. 在样本不足的情况下完成对 Transformer 模型的训练, 尽可能正确分类文本.

1.1 数据描述

训练集中含有 1599 条文本, 其中 799 条标签为“0”, 800 条标签为“1”. 测试集中含有 401 条文本, 其中 199 条标签为“0”, 202 条标签为“1”. 每条文本的长度不一, 最长的文本含有 26 个字符. 我们将测试集进一步划分为 5 折, 用于交叉验证. 各子集中标签数量如下表所示.

	Training Set					Test Set
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
# of 0	160	160	160	160	159	199
# of 1	160	160	160	160	160	202

表 1: 数据集划分

通过随机采样, 得到训练集中若干文本样本如下.

Label	Text
0	高考理科录取人数最多 20 个主流大众专业解析
0	中央民族大学 2009 年普通本科招生章程
0	研究生考试迟到 15 分钟不得入场
1	教育部就中小学教师队伍补充等有关工作答问
1	安理会半数成员反对巴勒斯坦加入联合国
1	美媒称国际刑事法庭决定逮捕苏丹领导人

表 2: 训练集中文本样本

初步推测, 数据集可能来源于新闻标题文本, 标签“0”可能与高等教育、考试等话题相关, 标签“1”可能与国内政治、国际关系等话题相关.

2 国内外研究现状

文本分类是自然语言处理领域中的经典问题. 由于非结构化的特性, 从文本中提取信息极具挑战性. 传统方法利用一组预定义的规则将文本分类到不同的类别

中, 并需要特定领域大量的专家知识, 目前已基本淘汰. 另一方面, 得益于神经网络强大的表达能力, 深度学习已经成为文本分类的主流方法.

2.1 词元级别的表示学习

用于文本分类的神经网络的核心组件是一个词嵌入模型, 将词元映射到低维连续特征向量. 这类模型通常采用自监督学习的方法, 在大规模文本语料上进行训练. 1989 年, Dumais 等人开发了潜在语义分析 (LSA) [1], 成为最早的词嵌入模型之一. 2001 年, Bengio 等人提出利用前馈神经网络学习语言的概率分布 [2]. 2013 年, Google 研究团队提出 Skip-Gram [3] 和 CBoW [4], 所得 word2vec 词嵌入在下游任务中得到广泛应用. 2014 年, Pennington 等人提出 GloVe [5], 通过统计全局词汇的共现信息来生成词向量. 2016 年, Bojanowski 等人提出 fastText [6], 在具有相似结构的词之间共享来自子词的参数.

近年来, 上下文敏感的词表示学习得到了广泛关注. 2017 年, Peters 等人开发了一个基于双向 LSTM 的词嵌入模型 ELMo [7], 由于捕捉了上下文信息, 相较 word2vec 效果显著提升. 与此同时, 随着 Transformer [8] 的提出, 词嵌入模型迎来了新一轮革命. 2018 年, OpenAI 开始使用 Transformer 构建语言模型, 提出的 GPT [9] 及后续版本已被广泛用于文本生成任务. 同年, Google 开发了基于双向 Transformer 编码器的 BERT [10]. BERT-large 包含 3 亿个参数, 训练数据包括 33 亿个单词, 是当前最先进的词嵌入模型之一.

目前, 基于 Transformer 的词嵌入模型仍在不断发展. 代表性工作有 RoBERTa [11], XLNet [12], ALBERT [13], ELECTRA [14] 等. 这些模型在预训练阶段采用了改进的训练目标或模型架构, 在训练速度、模型大小、计算量等方面进行了优化, 在下游任务上取得了更好的效果. 中文预训练词嵌入模型的代表性工作包括: (1) 清华大学的 ERNIE [15], 百度的 ERNIE 2.0 [16] & 3.0 [17], 核心思想在于通过知识增强预训练效果. (2) 哈工大讯飞联合实验室的 BERT-wwm & RoBERTa-wwm & MacBERT [18, 19], 核心思想在于掩蔽中文整词以及采用更先进的预训练任务.

2.2 序列级别的表示学习

为了从词元的向量表示中提取对文本分类有用的信息, 通常的做法是设计专用的网络模块, 组合得到整个序列的向量表示. 2014 年, Le 和 Mikolov 提出 doc2vec [20], 将各个词向量与段落向量进行平均或拼接, 用于文本分类. 同年, Kim 在预训练 word2vec 的基础上设计卷积神经网络, 用于文本分类 [21]. 循环神经网络, 例如 LSTM, 也被广泛地应用于提取序列表示, 代表性工作包括 [22].

基于 Transformer 的预训练语言模型使得任务不可知的文本表示成为可能. 利用 BERT 等模型得到的上下文敏感的词向量, 通常只需简易的平均池化和全连接

层即可较好地提取序列信息, 迁移至文本分类等下游任务, 并取得最先进的性能 [10]. 针对特定任务设计卷积或循环神经网络不再必需. 因此, 预训练-微调已经成为当前文本分类的主流范式.

3 课题总体方案设计

根据调研结果, 我们选用 BERT 作为本实验的词嵌入模型, 因为其有着良好的生态、先进的性能和广泛的应用. 作为改进, 我们同样尝试 RoBERTa, 以期进一步提升模型性能. 本节将简要介绍基于 BERT 的模型架构, BERT 和 RoBERTa 的预训练过程, 以及文本分类器的设计.

3.1 Transformer 文本分类网络

基于 BERT 的文本分类模型架构如图 1 所示.

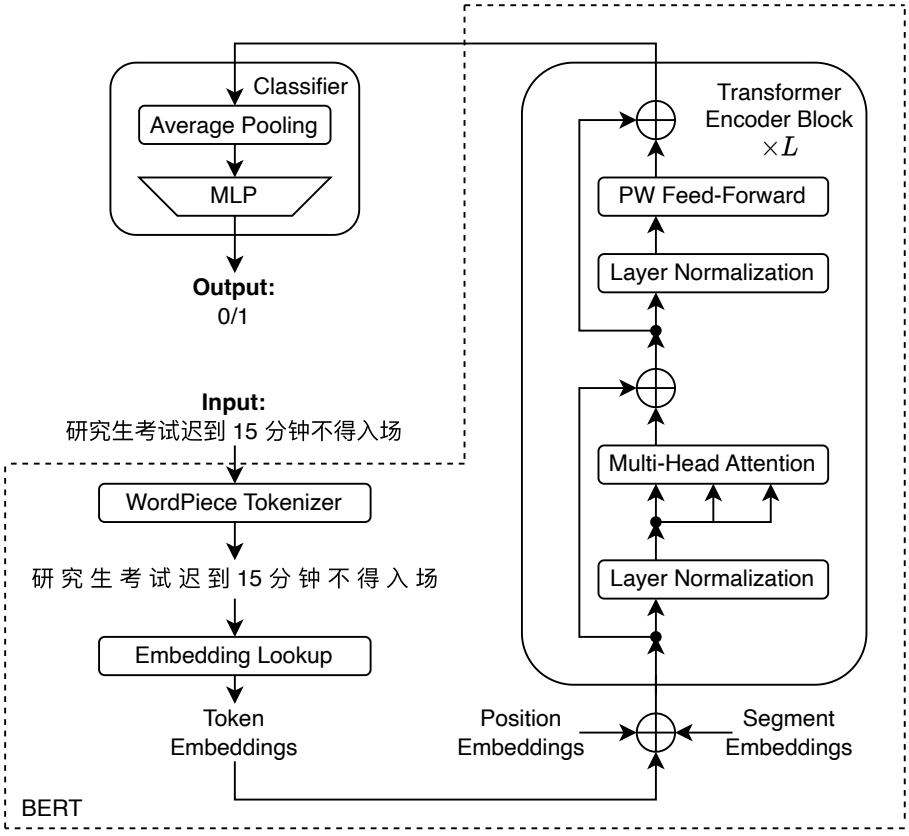


图 1: 基于 BERT 的文本分类网络

BERT 是一种基于 Transformer 的仅编码器的双向语言模型. 该模型首先将输入文本通过 WordPiece 分词器分割成词元序列, 并在首尾分别添加特殊的标记符号. 词元序列经过嵌入层, 得到每个词元的向量表示, 然后输入 Transformer 编码

器. 编码器模块由残差连接、归一化、多头自注意力、逐位置前馈网络等部分组成. 其中, 自注意力机制在序列建模中起到了至关重要的作用. 其核心公式为

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{1}$$

其中, Q, K, V 分别为查询、键、值的矩阵表示, d_k 为键的维度. 输入序列中第 i 个词元的向量表示 $\boldsymbol{x}_i$ 与三个权重矩阵依次相乘, 得到第 i 个查询 $\boldsymbol{q}_i = \boldsymbol{x}_i W_Q$, 键 $\boldsymbol{k}_i = \boldsymbol{x}_i W_K$, 值 $\boldsymbol{v}_i = \boldsymbol{x}_i W_V$. 通过归一化的点积计算注意力权重, 并与值矩阵相乘, 得到第 i 个词元的输出 $\boldsymbol{y}_i = \text{Attention}(\boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i)$. 一套权重矩阵 ($W_Q$, W_K, W_V) 称作一个注意力头, 多个注意力头并行计算, 并将结果拼接, 得到最终的输出 $\boldsymbol{y}_i = \text{Concat}(\boldsymbol{y}_i^1, \boldsymbol{y}_i^2, \ldots, \boldsymbol{y}_i^H) W_O$, 其中 H 为注意力头的数量, W_O 为输出矩阵. 通常, 设置 $d_k = d_v = d/H$, 其中 d 为词向量的特征维度.

经多头自注意力混合后的词向量将输入到逐位置的前馈网络, 捕捉同一向量内不同特征维度之间的依赖关系. 逐位置前馈网络由两层全连接层组成, 两层之间有 GELU 激活函数, 隐层神经元数量为特征维度的 4 倍. 为了训练的稳定性, 在每一层的输入和输出之间都有残差连接和归一化操作. 一个基础尺寸的 BERT 模型 (即 BERT-base) 通常设置 Transformer 编码器的模块数为 $L = 12$, 注意力头的数量为 $H = 12$, 词向量的特征维度为 $d = 768$, 模型参数总量约为 1.1 亿.

得益于先进的序列建模方式和充分的自监督预训练, BERT 模型最终输出的词向量具有上下文敏感、任务不可知的优点, 只需极小程度的扩展即可用于文本分类的下游任务. 具体而言, 我们首先将序列中的所有词向量进行平均池化, 得到整个序列的向量表示. 然后, 通过一个双层感知机将序列向量映射到二分类的概率分布.

3.2 预训练的改进: 从 BERT 到 RoBERTa

本实验中, 我们直接加载 Google 提供的 `bert-base-chinese`[1] 和哈工大讯飞联合实验室提供的 `hfl/chinese-roberta-wwm-ext`[2] 预训练 BERT 权重. 出于完整性考虑, 我们仍简要介绍 BERT 和 RoBERTa 的预训练过程.

BERT 的预训练过程包含两个任务: (1) Masked Language Model (MLM), (2) Next Sentence Prediction (NSP). MLM 任务的目标是预测序列中被掩蔽的词元, 以此鼓励模型学习上下文信息. 例如, 对于输入句子“中科大真是牛校出牛子”, 我们随机选择一个单词, 将其替换为特殊的 [MASK] 词元, 得到“中科 [MASK] 真是牛校出牛子”, 并让模型预测被掩蔽的词元; NSP 任务的目标是预测两个句子是否连续, 以此鼓励模型学习句子之间的关系. 例如, 模型输入为“[CLS] 中科大真是牛校出牛子 [SEP] 我也想去中科大 [SEP]”, 则模型应当判断两个句子连续. 其中,

[1] https://huggingface.co/bert-base-chinese

[2] https://huggingface.co/hfl/chinese-roberta-wwm-ext

[CLS] 和 [SEP] 为特殊的词元, 前者用于 NSP 分类任务, 后者用于分割句子. 通常, 以 50% 的概率随机选择两个连续句子或任意句子作为输入对. 本实验中使用的 BERT-base 特指在中文维基百科上预训练得到的模型.

RoBERTa 同样采用 BERT 架构, 但对预训练过程进行了改进, 主要包括: (1) 采用动态掩码而非在预处理阶段进行的静态掩码. (2) 放弃了 NSP 训练任务. (3) 采用更大的 Batch Size 和词典大小. 本实验中使用的 RoBERTa-wwm-ext 则是针对中文数据的改进, 主要包括: (1) 采用中文整词掩码而非单个汉字. 例如, “牛子”作为一个整体被掩蔽, 得到“中科大真是牛校出 [MASK][MASK]”. (2) 采用来自百科、新闻、问答网站的扩展数据集, 相较于中文维基百科, 语料规模增加了 10 倍以上. 我们期望 RoBERTa-wwm-ext 在中文文本分类任务上取得更好的效果.

3.3 两种微调方式: 冻结与不冻结

视觉基础模型的微调过程通常只对预测头进行参数更新, 而预训练权重一律冻结 (例如, 第一次作业). 然而, 基于 Transformer 的预训练语言模型的微调往往对两者同时更新. 我们对两种情况均感兴趣, 因此将 BERT-base 和 RoBERTa-wwm-ext 分别冻结和不冻结, 计划进行对比.

在我们的模型架构中, BERT 模型由预训练权重初始化, MLP 分类器则随机初始化, 按照同样的训练策略进行更新. 与预训练过程不同, 微调属于有监督学习, 旨在最小化分类器输出 Logits 与标签之间的交叉熵损失. 我们将在下一节给出与训练相关的超参数.

4 模型测试实验设计

4.1 对比实验设计

为了展示不同预训练权重和微调方式对模型性能的影响, 我们设计了四组实验, 对应如下四个模型:

1. **BERT-base (frozen)**: 加载 `bert-base-chinese` 权重, 词嵌入冻结, 仅分类器更新.
2. **RoBERTa-wwm-ext (frozen)**: 加载 `hfl/chinese-roberta-wwm-ext` 权重, 词嵌入冻结, 仅分类器更新.
3. **BERT-base**: 加载 `bert-base-chinese` 权重, 词嵌入和分类器同时更新.
4. **RoBERTa-wwm-ext**: 加载 `hfl/chinese-roberta-wwm-ext` 权重, 词嵌入和分类器同时更新.

对于每一个模型, 我们使用 Fold 2~5 作为训练集, Fold 1 作为验证集. 将准确率和 F_1 分数作为评价指标. 在训练过程中记录验证集上的损失函数及准确率变化.

最终, 我们将在测试集上比较文本二分类的准确率及 F_1 分数.

4.2 超参数设置

预训练 BERT 模型的超参数与原论文保持一致, Transformer 编码器的模块数为 $L = 12$, 注意力头的数量为 $H = 12$, 词向量的特征维度为 $d = 768$, 使用 GELU 激活函数, 模型参数总量约为 1.1 亿. 此外, 我们自行设计 MLP 分类器, 隐层大小为 $2d = 1536$, 输出层大小为 2, 采用 SiLU 激活函数, 交叉熵损失函数, 不使用标签平滑. 为了防止过拟合, 我们在 MLP 分类器中采用了 0.1 的 Dropout 概率. 优化器采用 Adam, 学习率为 2×10^{-5}, 批大小为 32, 训练 3 个 Epoch. 以上超参数设置对所有四个模型适用. 模型架构的更多细节参见实验代码.

本实验所有模型采用 Tensorflow 和 Keras 实现, 在 1 块 NVIDIA GeForce GTX 1080 Ti GPU 上完成微调.

5 模型测试结果及分析

实验主要结果如表 3, 5 和图 2 所示.

5.1 混淆矩阵

		Predicted	
		0	1
Actual	0	187	12
	1	6	196

(a) BERT-base (frozen)

		Predicted	
		0	1
Actual	0	186	13
	1	8	194

(b) RoBERTa-wwm-ext (frozen)

		Predicted	
		0	1
Actual	0	196	3
	1	4	198

(c) BERT-base

		Predicted	
		0	1
Actual	0	198	1
	1	4	198

(d) RoBERTa-wwm-ext

表 3: 不同版本模型的混淆矩阵

表 3 给出了四个版本模型在测试集上的混淆矩阵. 可见, 四个模型均能准确分类大部分中文文本样本. 其中 RoBERTa-wwm-ext 的结果最优, 仅有 1 个标签为“0”的样本被误分类为“1”, 4 个标签为“1”的样本被误分类为“0”.

Actual	Predicted	Text
0	1	美媒：“海归派”在中国势力逐渐壮大
1	0	武汉高校公寓楼突发大火百余人获疏散
1	0	我国 10 个专业学位硕士将增招应届毕业生
1	0	教育部：加大对民族教育支持力度
1	0	高校毕业生将可获至少 10 天创业培训

表 4: RoBERTa-wwm-ext 错误分类的样本

我们列举出一些错误分类的样本, 并尝试分析其原因. 观察发现, 被错误分类的“0”与高等教育话题相关, 可能是由于涉及“美媒”“中国”等字眼, 因此模型错误预测为“1”. 被错误分类的“1”同时与教育和国家政策相关, 标签本身存在一定的噪声或混淆, 因此导致模型错误分类. 总体而言, 微调得到的模型能够合理地区分两类文本.

5.2 性能比较

Dataset	Model	Accuracy	F_1 Score
Train.	BERT-base (frozen)	0.9492	0.9492
	RoBERTa-wwm-ext (frozen)	0.9515	0.9515
	BERT-base	0.9906	0.9906
	RoBERTa-wwm-ext	0.9898	0.9898
Valid.	BERT-base (frozen)	0.9563	0.9562
	RoBERTa-wwm-ext (frozen)	0.9625	0.9625
	BERT-base	0.9781	0.9781
	RoBERTa-wwm-ext	0.9781	0.9781
Test	BERT-base (frozen)	0.9551	0.9551
	RoBERTa-wwm-ext (frozen)	0.9476	0.9476
	BERT-base	0.9825	0.9825
	RoBERTa-wwm-ext	0.9875	0.9875

表 5: 不同版本模型的分类准确率及 F_1 分数

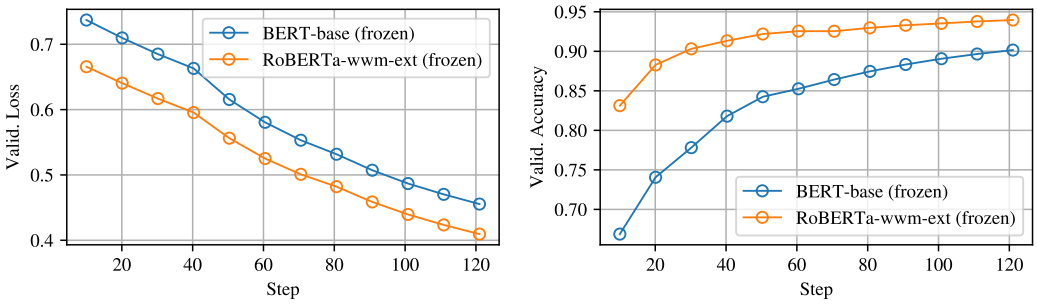
表 5 给出了四个版本模型在训练集、验证集和测试集上的分类准确率及 F_1 分

数. 我们可以看到, 四种模型均能较好地区分两类文本, 准确率普遍达到 95% 乃至更高, 体现出基于 BERT 的预训练-微调范式的优越性.

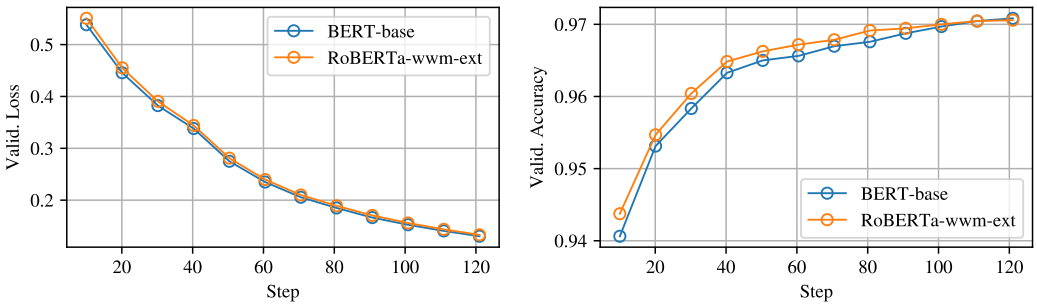
此外, RoBERTa-wwm-ext 的性能略优于 BERT-base, 说明 RoBERTa-wwm-ext 在中文文本分类任务上的确取得了更好的效果, 测试集准确率高达 98.75%, 体现出整词掩蔽、扩展数据集等预训练改进的有效性.

不同的微调方式也会对模型性能产生较大的影响. 在 BERT-base 上, 词嵌入与分类器共同更新使得准确率相较冻结的情况提高了 2.74 个百分点, 而在 RoBERTa-wwm-ext 上提高了 3.99 个百分点. 这说明词嵌入与下游任务的相关性较强, 适合与分类器共同更新, 以达到域适应的效果.

5.3 训练曲线分析



(a) BERT 冻结的情况下



(b) BERT 不冻结的情况下

图 2: 训练过程中的损失函数及准确率变化

图 2 给出了四个版本模型在训练过程中的损失函数及准确率变化. 冻结词嵌入进行微调具有更小的计算量和更快的训练速度, 但是在相同 Epoch 的情况下收敛水平远不及同时更新词嵌入和分类器的模型. 这进一步证明了微调方式对模型性能的重要性.

此外, 针对 RoBERTa-wwm-ext 在两种情况下均取得比 BERT-base 更好的收敛水平, 与我们的预期相符, 体现了先进的预训练策略的有效性.

6 实验总结

本实验中, 我们基于 BERT 和 RoBERTa, 采用预训练-微调范式, 设计了中文文本二分类模型, 在中文文本分类任务上取得了较好的效果. 通过对比实验, 我们发现: (1) 相较 BERT-base, 针对中文改进了预训练策略的 RoBERTa-wwm-ext 在中文文本分类任务上取得了更好的效果; (2) 相较单独更新分类器, 词嵌入与分类器共同更新的方式能够进一步提升模型性能, 适合于 BERT 等预训练模型.

7 参考文献

[1] Susan T. Dumais. Latent semantic analysis. 38(1):188–230. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aris.1440380105.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model.

[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. version: 3.

[5] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

[6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information.

[7] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[9] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach.

[12] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized autoregressive pretraining for language understanding.

[13] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations.

[14] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators.

[15] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451. Association for Computational Linguistics.

[16] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE 2.0: A continual pre-training framework for language understanding.

[17] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation.

[18] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. Revisiting pre-trained models for chinese natural language processing. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668. Association for Computational Linguistics.

[19] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. Pre-training with whole word masking for chinese BERT. 29:3504–3514. Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[20] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents.

[21] Yoon Kim. Convolutional neural networks for sentence classification.

[22] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading.