【ERNIE】芝麻街跨界NLP，没有一个ERNIE是无辜的

深度学习自然语言处理 2020-04-27

以下文章来源于NewBeeNLP ，作者kaiyuan

NewBeeNLP

永远有料，永远有趣

之前发在知乎、AINLP以及CSDN上的预训练模型系列文章，最近打算整理到公号上。另外欢迎大家左下角**阅读原文**关注我的知乎专栏：【BERT巨人肩膀】

这篇文章会为大家介绍下同名的"ERNIE"小伙伴们，在预训练模型的飞速发展下，**芝麻街恐成最大赢家**🙈

> ERNIE: Enhanced Language Representation with Informative Entities（THU）
> [1]

本文的工作也是属于对BERT锦上添花，将知识图谱的一些结构化信息融入到BERT中，使其更好地对真实世界进行语义建模。也就是说，原始的bert模型只是机械化地去学习语言相关的“合理性”，而并学习不到语言之间的语义联系，打个比喻，就比如掉包xia只会掉包，而不懂每个包里面具体是什么含义。于是，作者们的工作就是如何将这些额外的知识告诉bert模型，而让它更好地适用于NLP任务。

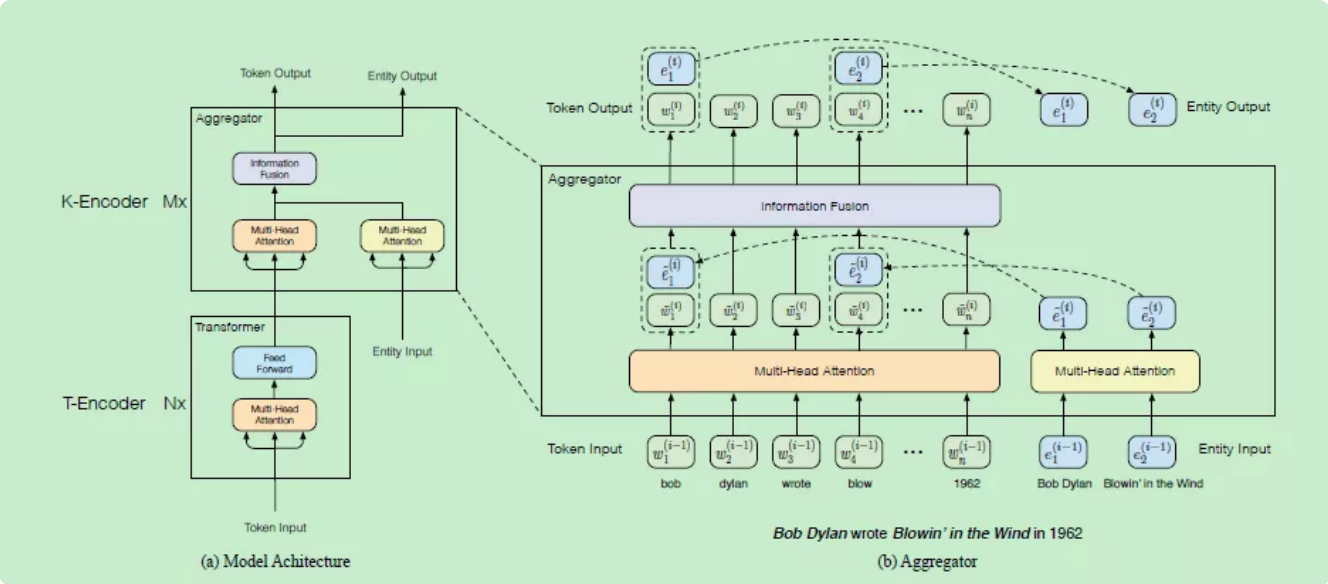
但是要将外部知识融入到模型中，又存在两个问题：

- 「**Structured Knowledge Encoding:**」 对于给定的文本，如何高效地抽取并编码对应的知识图谱事实；
- 「**Heterogeneous Information Fusion:**」 语言表征的预训练过程和知识表征过程有很大的不同，它们会产生两个独立的向量空间。因此，如何设计一个特殊的预训练目标，以融合词汇、句法和知识信息又是另外一个难题。

为此，作者们提出了ERNIE模型，同时在大规模语料库和知识图谱上预训练语言模型：

1. **「抽取+编码知识信息：」** 识别文本中的实体，并将这些实体与知识图谱中已存在的实体进行实体对齐，具体做法是采用知识嵌入算法（如TransE），并将得到的entity embedding作为ERNIE模型的输入。基于文本和知识图谱的对齐，ERNIE 将知识模块的实体表征整合到语义模块的隐藏层中。
2. **「语言模型训练：」** 在训练语言模型时，除了采用bert的MLM和NSP，另外随机mask掉了一些实体并要求模型从知识图谱中找出正确的实体进行对齐（这一点跟baidu的entity-masking有点像）。

okay，接下来看看模型到底长啥样？



如上图，整个模型主要由两个子模块组成：

- 底层的**「textual encoder (T-Encoder)」**，用于提取输入的基础词法和句法信息，N个；
- 高层的**「knowledgeable encoder (K-Encoder)」**， 用于将外部的知识图谱的信息融入到模型中，M个。

knowledgeable encoder

这里T-encooder跟bert一样就不再赘述，主要是将文本输入的三个embedding加和后送入双向Transformer提取词法和句法信息：

$$\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\} = \mathrm{T} - \mathrm{Encoder}\left(\{w_1, \ldots, w_n\}\right)$$

K-encoder中的模型称为aggregator，输入分为两部分：

- 一部分是底层T-encoder的输出$\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$
- 一部分是利用TransE算法得到的文本中entity embedding，$\{e_1, \ldots, e_m\}$
- 注意以上为第一层aggregator的输入，后续第K层的输入为第K-1层aggregator的输出

接着利用multi-head self-attention对文本和实体分别处理：

$$\left\{\tilde{\boldsymbol{w}}_1^{(i)}, \ldots, \tilde{\boldsymbol{w}}_n^{(i)}\right\}=\mathrm{MH}-\mathrm{ATT}\left(\left\{\boldsymbol{w}_1^{(i-1)}, \ldots, \boldsymbol{w}_n^{(i-1)}\right\}\right)$$
$$\left\{\tilde{\boldsymbol{e}}_1^{(i)}, \ldots, \tilde{\boldsymbol{e}}_m^{(i)}\right\}=\mathrm{MH}-\mathrm{ATT}\left(\left\{\boldsymbol{e}_1^{(i-1)}, \ldots, \boldsymbol{e}_m^{(i-1)}\right\}\right)$$

然后就是将实体信息和文本信息进行融合，实体对齐函数为 $e_k=f(w_j)$:

- 对于有对应实体的输入：

$$\boldsymbol{h}_j=\sigma\left(\tilde{\boldsymbol{W}}_t^{(i)} \tilde{\boldsymbol{w}}_j^{(i)}+\tilde{\boldsymbol{W}}_e^{(i)} \tilde{\boldsymbol{e}}_k^{(i)}+\tilde{\boldsymbol{b}}^{(i)}\right)$$
$$\boldsymbol{w}_j^{(i)}=\sigma\left(\boldsymbol{W}_t^{(i)} \boldsymbol{h}_j+\boldsymbol{b}_t^{(i)}\right)$$
$$\boldsymbol{e}_k^{(i)}=\sigma\left(\boldsymbol{W}_e^{(i)} \boldsymbol{h}_j+\boldsymbol{b}_e^{(i)}\right)$$

- 对于没有对应实体的输入词：

$$\boldsymbol{h}_j=\sigma\left(\boldsymbol{W}_t^{(i)} \tilde{\boldsymbol{w}}_j^{(i)}+\tilde{\boldsymbol{b}}^{(i)}\right)$$
$$\boldsymbol{w}_j^{(i)}=\sigma\left(\boldsymbol{W}_t^{(i)} \boldsymbol{h}_j+\boldsymbol{b}_t^{(i)}\right)$$

上述过程就是一个aggregator的操作，整个K-encoder会叠加M个这样的block：

$$\left\{\boldsymbol{w}_1^{(i)}, \ldots, \boldsymbol{w}_n^{(i)}\right\},\left\{e_1^{(i)}, \ldots, \boldsymbol{e}_m^{(i)}\right\}=\text{ Aggregator }(\left\{\boldsymbol{w}_1^{(i-1)}, \ldots, \boldsymbol{w}_n^{(i-1)}\right\},\left\{e_1^{(i-1)}, \ldots\right.$$

最终的输出为最顶层的Aggregator的token embedding和entity embedding。

改进的预训练

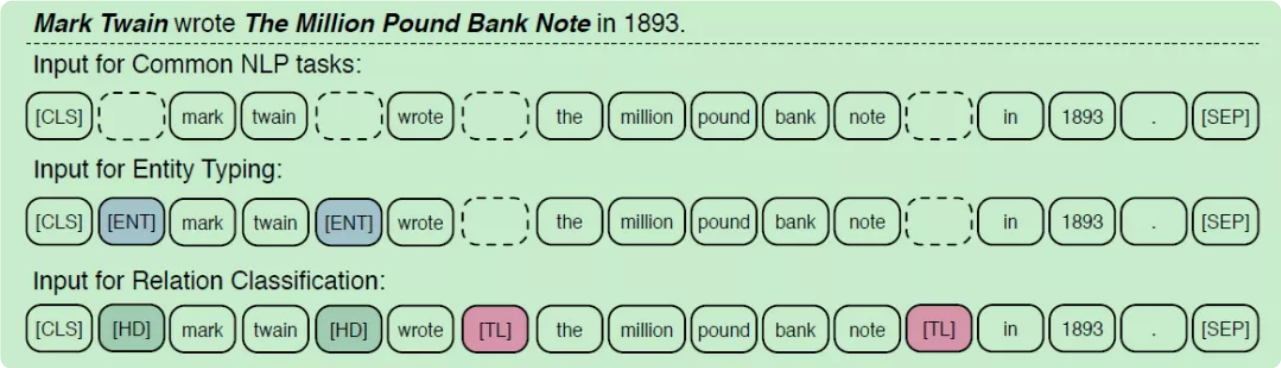
除了跟bert一样的MLM和NSP预训练任务，本文还提出了另外一种适用于信息融合的预训练方式，**「denoising entity auto-encoder (dEA).」** 跟下文baidu的还是有点不一样，这里是有对齐后的entity sequence输入的，而百度的是直接去学习entity embedding。dEA 的目的就是要求模型能够根据给定的实体序列和文本序列来预测对应的实体：

$$p(e_j|w_i)=\frac{\exp(1 \text{ inear }(\boldsymbol{w}_i^o) \cdot \boldsymbol{e}_j)}{\sum_{k=1}^m \exp(1 \text{ i near }(\boldsymbol{w}_i^o) \cdot \boldsymbol{e}_k)}$$

微调

为了使得模型可以更广泛地适用于不同的NLP任务，作者也学习BERT设计了不同的特殊的token：

- 【CLS】：该token含有句子信息的表示，可适用于一般任务
- 【HD】和【TL】：该token表示关系分类任务中的头实体和尾实体（类似于传统关系分类模型中的位置向量），然后使用【CLS】来做分类；
- 【ENT】：该token表示实体类型，用于entity typing等任务。



试验部分也略过了哈~感觉有些部分还不是很清晰，需要看看源码...

reference

- ACL 2019将会有哪些值得关注的论文？[2]
- ACL 2019 | 基于知识增强的语言表示模型，多项NLP任务表现超越BERT[3]
- ACL 2019 | 清华等提出ERNIE：知识图谱结合BERT才是「有文化」的语言模型[4]
- 官方源码[5]

> ERNIE: Enhanced Representation through Knowledge Integration（Baidu）[6]

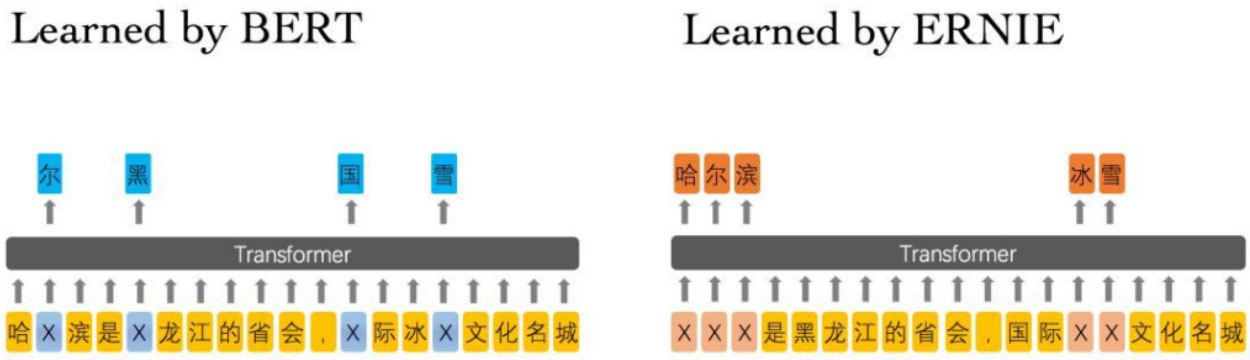
百度提出的ERNIE模型主要是针对BERT在中文NLP任务中表现不够好提出的改进。我们知道，对于中文，bert使用的基于字的处理，在mask时掩盖的也仅仅是一个单字，举个栗子：

> 我在上海交通大学玩泥巴-------> 我 在 上 【mask】 交 通 【mask】学 玩 【mask】 巴。

作者们认为通过这种方式学习到的模型能很简单地推测出字搭配，但是并不会学习到短语或者实体的语义信息， 比如上述中的【上海交通大学】。于是文章提出一种知识集成的BERT模型，别称ERNIE。ERNIE模型在BERT的基础上，加入了海量语料中的实体、短语等先验语义知识，建模真实世界的语义关系。

在具体模型的构建上，也是使用的Transformer作为特征抽取器。这里如果对于特征抽取不是很熟悉的同学，强烈推荐张俊林老师的"放弃幻想，全面拥抱Transformer：自然语言处理三大特征抽取器（CNN/RNN/TF）比较[7]"。

那么怎么样才能使得模型学习到文本中蕴含的潜在知识呢？不是直接将知识向量直接丢进模型，而是在训练时将短语、实体等先验知识进行mask，强迫模型对其进行建模，学习它们的语义表示。



哈尔滨是黑龙江的省会，国际冰雪文化名城

具体来说， ERNIE采用三种masking策略：

- 「**Basic-Level Masking：**」 跟bert一样对单字进行mask，很难学习到高层次的语义信息；
- 「**Phrase-Level Masking：**」 输入仍然是单字级别的，mask连续短语；
- 「**Entity-Level Masking：**」 首先进行实体识别，然后将识别出的实体进行mask。

经过上述mask训练后，短语信息就会融入到word embedding中了

Sentence	Harry	Potter	is	a	series	of	fantasy	novels	written	by	British	author	J.	K.	Rowling
Basic-level Masking	[mask]	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	J.	[mask]	Rowling
Entity-level Masking	Harry	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]
Phrase-level Masking	Harry	Potter	is	[mask]	[mask]	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]

Figure 2: Different masking level of a sentence

此外，为了更好地建模真实世界的语义关系，ERNIE预训练的语料引入了多源数据知识，包括了中文维基百科，百度百科，百度新闻和百度贴吧（可用于对话训练）。

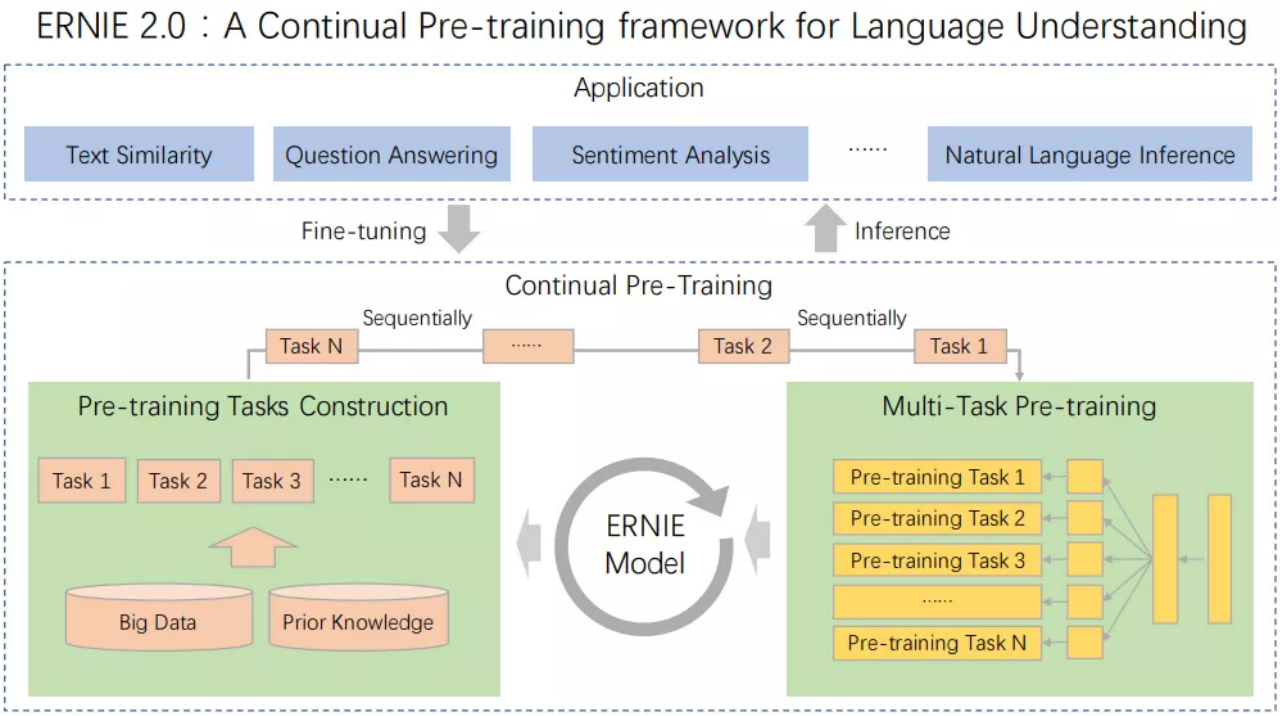
关于论文后面的试验就不再赘述。

reference：

- 如何评价百度新发布的NLP预训练模型ERNIE？[8]
- 中文任务全面超越 BERT：百度正式发布NLP预训练模型ERNIE[9]
- 官方源码[10]

ERNIE2.0: A Continual Pre-training Framework for Language Understanding [11]

Baidu团队之前发布的ERNIE1.0效果就不错，虽然基础框架沿袭BERT，但是训练语料以及mask策略的改进，使其在中文任务上表现更好。这刚过了几个月，又发布了增强版的ERNIE，最近NLP社区更新速度可见一斑。先前的模型比如ELMO、GPT、BERT、ERNIE1.0、XLNet等都是基于词和句子的共现关系来训练的，这导致模型不能够很好地建模词法、句法以及语义信息。为此，ERNIE2.0提出了**「通过不断增量预训练任务进行多任务学习」**来将词法句法以及语义信息融入到模型当中去。整体流程如下所示，首先利用简单的任务初始化模型，接着以串行的方式进行**「持续学习（Continual Learning）」**，对于每次新增的训练任务，模型可以利用之前已经训练过的任务信息去更好地学习新任务，这跟人类的学习方式是一样的。



模型框架

整体框架还是基本跟ERNIE1.0的一样，不过ERNIE2.0为了匹配多任务持续学习的理念，需要在输入的时候额外增加一部分「**Task Embedding**」，用来告诉模型这是在处理哪个任务。

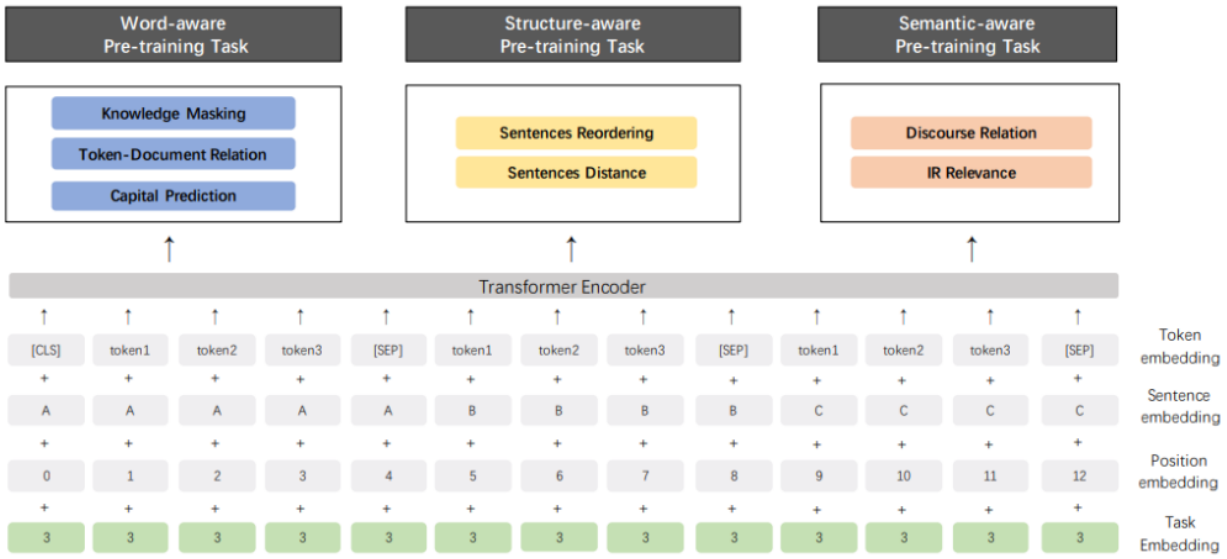


Figure 3: The structure of the ERNIE 2.0 model. The input embedding contains the token embedding, the sentence embedding, the position embedding and the task embedding. Seven pre-training tasks belonging to different kinds are constructed in the ERNIE 2.0 model.

预训练任务

前面说到要让模型获取词法、句法以及语义的信息，那么怎么设计合适的预训练任务就成了非常重要的一环。其实BERT本身也可以看做是多任务（MLM+NSP），然后对于扩展BERT至多任务，MTDNN也有过尝试，使用了GLUE相似的任务进行训练然后在GLUE上SOTA了。不过ERNIE2.0与MTDNN在任务设计上不同的是，在预训练阶段使用的任务基本都是无监督或者是弱监督的。要知道在NLP中有标注的数据不多，但是无标注的数据可以说是源源不断，如果能好好利用起来简直功德圆满。okay，下面我们来介绍一下具体的任务设计

Word-aware Pre-training Tasks

基于单词的预训练任务用于获取词法信息

- 「**Knowledge Masking Task：**」 就是ERNIE1.0使用的预训练任务，将实体与短语进行mask，具体可以上文
- 「**Capitalization Prediction Task：**」 预测单词是否大写。因为在语料中大写字词通常具有特殊含义
- 「**Token-Document Relation Prediction Task ：**」 预测某一个段落的token是否出现在同一篇文档的另外段落中。可以认为是对关键字进行建模

Structure-aware Pre-training Tasks

主要是用于建模句法信息

- 「Sentence Reordering Task：」 具体而言是把一段话拆分成多个segment，之后对其进行排列组合，让模型去预测正确的原始顺序。感觉有点像高中英语试卷大作文前面的那一题hhh...
- 「Sentence Distance Task：」 预测句子之间的距离，可以看做是三分类的任务，其中“0”表示两个句子是同一篇文档中相邻的，“1”表示两个句子在同一篇文档中但是不相邻，“2”表示两个句子不在同一个文档中。这个任务的话可以看做是BERT的NSP任务的扩展版

Semantic-aware Pre-training Tasks

主要用于建模语法信息

- 「Discourse Relation Task：」 预测两个句子之间的语义或修辞关系。
- 「IR Relevance Task：」 学习信息检索中短文本的相关性。百度作为搜索引擎的优势就是有大量的「query」和「answer」可以用于模型训练。这也是一个三分类的任务，输入为query+title，输出为标签，其中“0”表示这两个是强相关的（定义为用户点击的结果条目），“1”表示弱相关（定义为搜索返回结果中不被用户点击的条目），“2”表示不相关（定义为没有出现在返回结果里的条目）

模型效果

okay，介绍完模型，我们来看看效果怎么样~ERNIE2.0以及BERT在GLUE上的表现，可以看出基本在所有任务上ERNIE2.0的效果都超过了原始的BERT和XLNet。

Task(Metrics)	*BASE model*		*LARGE model*				
	Test		Dev			Test	
	BERT	ERNIE 2.0	BERT	XLNet	ERNIE 2.0	BERT	ERNIE 2.0
CoLA (Matthew Corr.)	52.1	**55.2**	60.6	63.6	**65.4**	60.5	**63.5**
SST-2 (Accuracy)	93.5	**95.0**	93.2	95.6	**96.0**	94.9	**95.6**
MRPC (Accurary/F1)	84.8/88.9	**86.1/89.9**	88.0/-	89.2/-	**89.7/-**	85.4/89.3	**87.4/90.2**
STS-B (Pearson Corr./Spearman Corr.)	87.1/85.8	**87.6/86.5**	90.0/-	91.8/-	**92.3/-**	87.6/86.5	**91.2/90.6**
QQP (Accuracy/F1)	89.2/71.2	**89.8/73.2**	91.3/-	91.8/-	**92.5/-**	89.3/72.1	**90.1/73.8**
MNLI-m/mm (Accuracy)	84.6/83.4	**86.1/85.5**	86.6/-	**89.8/-**	89.1/-	86.7/85.9	**88.7/88.8**
QNLI (Accuracy)	90.5	**92.9**	92.3	93.9	**94.3**	92.7	**94.6**
RTE (Accuracy)	66.4	**74.8**	70.4	83.8	**85.2**	70.1	**80.2**
WNLI (Accuracy)	**65.1**	**65.1**	-	-	-	65.1	**67.8**
AX(Matthew Corr.)	34.2	**37.4**	-	-	-	39.6	**48.0**
Score	78.3	**80.6**	-	-	-	80.5	**83.6**

Table 6: The results on GLUE benchmark, where the results on dev set are the median of five experimental results and the results on test set are scored by the GLUE evaluation server (https://gluebenchmark.com/leaderboard). The state-of-the-art results are in bold. All of the fine-tuned models of AX is trained by the data of MNLI.

这是中文数据集上的模型比对效果，目前中文版的模型好像还没有发布出来

Task	Metrics	BERT_{BASE}		ERNIE 1.0_{BASE}		ERNIE 2.0_{BASE}		ERNIE 2.0_{LARGE}	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test
CMRC 2018	EM/F1	66.3/85.9	-	65.1/85.1	-	69.1/88.6	-	**71.5/89.9**	-
DRCD	EM/F1	85.7/91.6	84.9/90.9	84.6/90.9	84.0/90.5	88.5/93.8	88.0/93.4	**89.7/94.7**	**89.0/94.2**
DuReader	EM/F1	59.5/73.1	-	57.9/72.1	-	61.3/74.9	-	**64.2/77.3**	-
MSRA-NER	F1	94.0	92.6	95.0	93.8	95.2	93.8	**96.3**	**95.0**
XNLI	Accuracy	78.1	77.2	79.9	78.4	81.2	79.7	**82.6**	**81.0**
ChnSentiCorp	Accuracy	94.6	94.3	95.2	95.4	95.7	95.5	**96.1**	**95.8**
LCQMC	Accuracy	88.8	87.0	89.7	87.4	**90.9**	**87.9**	**90.9**	**87.9**
BQ Corpus	Accuracy	85.9	84.8	86.1	84.8	86.4	85.0	**86.5**	**85.2**
NLPCC-DBQA	MRR/F1	94.7/80.7	94.6/80.8	95.0/82.3	95.1/82.7	95.7/84.7	95.7/85.3	**95.9/85.3**	**95.8/85.8**

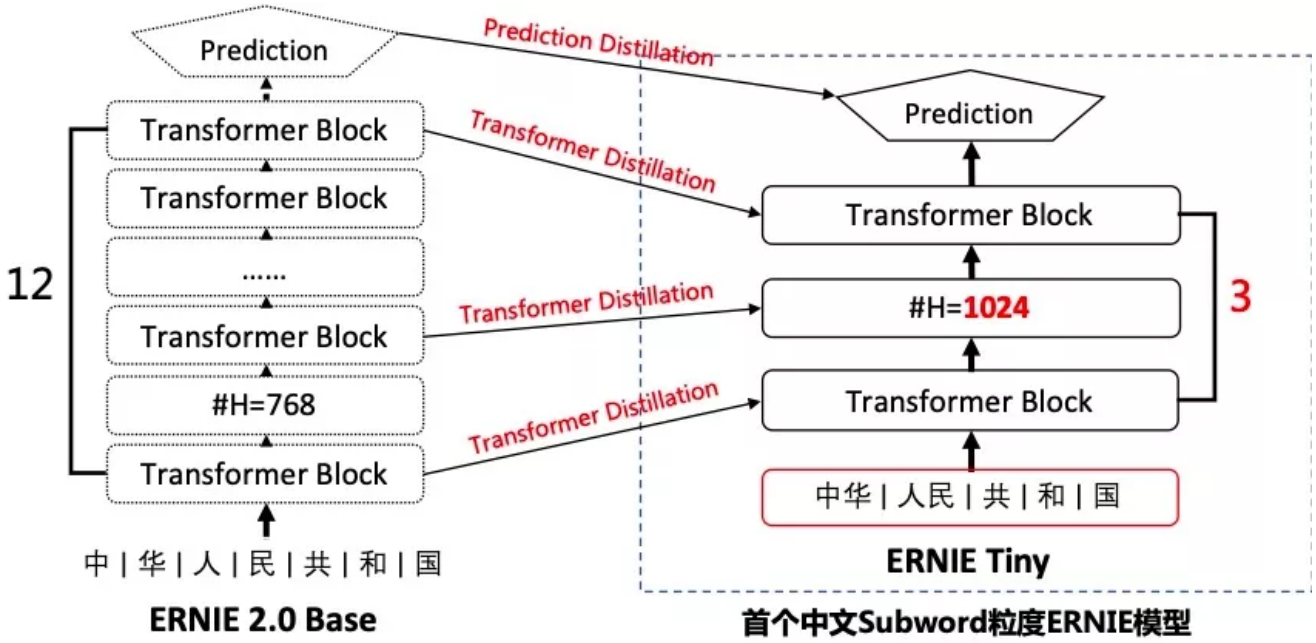
Table 7: The results of 9 common Chinese NLP tasks. ERNIE 1.0 indicates our previous model ERNIE[4]. The reported results are the average of five experimental results, and the state-of-the-art results are in bold.

reference

- 官方开源代码[12]
- 如何评价百度最新发布的ERNIE2.0？[13]
- ERNIE 2.0：芝麻街 2.0？[14]

ERNIE-Tiny

ERNIE-Tiny也是baidu的工作，在一波模型轻量化的风潮之下（更小的模型！迈向更快更环保的NLP），好多预训练模型都出现了XXX-Tiny的延伸，旨在提升预训练模型在实际工程应用中的落地能力。相较于base模型，ERNIE-Tiny采用了以下4点技术，保证了在实际真实数据中将近4.3倍的预测提速。



更浅的模型

最直观的想法就是直接截取base模型的前几层进行下游任务的finetune，但是这样会造成pretrain-finetune discrepancy。因此需要重新训练一个浅层的模型，将12层的ERNIE Base模型直接压缩为3层，线性提速4倍，但效果也会有较大幅度的下降；

更大的hidden_size

为了弥补模型变浅带来的效果下降，这里将原始的hidden_size由768提升至1024，**你看这模型它又矮又胖。**

subword词粒度

transformer-based模型预测时间性能与输入长度线性相关，通过subword粒度替换字（char）粒度，能够明显地缩短输入文本的长度。统计表明，在XNLI dev集上采用subword字典切分出来的序列长度比字表平均缩短40%。

知识蒸馏

为了进一步提升模型的效果，ERNIE Tiny扮演学生角色，利用模型蒸馏的方式在Transformer层和Prediction层去学习教师模型ERNIE模型对应层的分布或输出，这种方式能够缩近ERNIE Tiny和ERNIE的效果差异。

reference

- ERNIE-tiny GITHUB[15]

ERNIE-GEN

ERNIE-GEN也是baidu的ERNIE套餐之一，看名字就知道是把ERNIE用在生成任务上的，论文在今年一月就放出来了，说的也都是SOTA，但是好像一直也没见宣传和讨论。这里篇幅原因先不展开了，感兴趣可以去了解下。

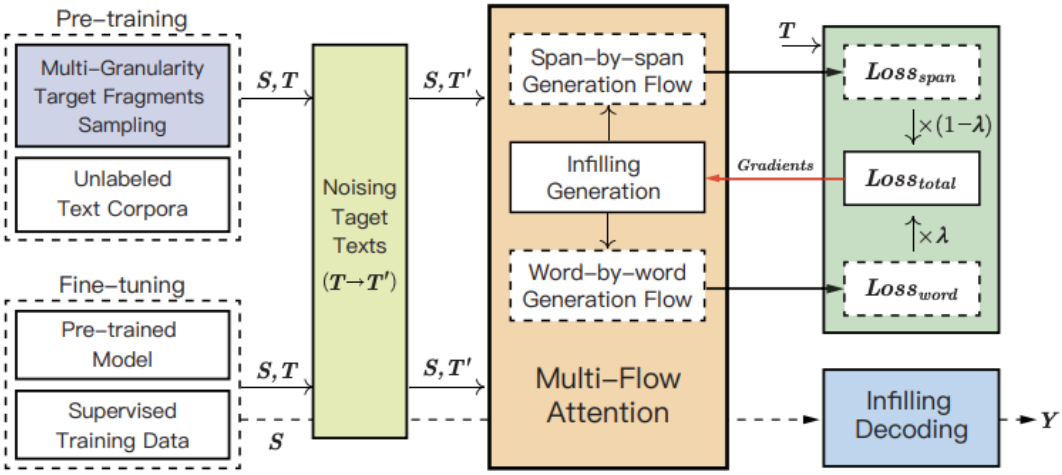


Figure 2: Overview of ERNIE-GEN. $\boldsymbol{S}, \boldsymbol{T}$ and $\boldsymbol{Y}$ donate source, target, and generated texts, $\boldsymbol{T}'$ is the noised version of $\boldsymbol{T}$.

ERNIE-Classification

这是一个github项目，基于Keras/TensorFlow 2以及HuggingFace's Transformers，集成多个bert-based模型用于句子分类，有需要的可以试试效果。

本文参考资料

[1]ERNIE: Enhanced Language Representation with Informative Entities (THU/ACL2019)：https://arxiv.org/pdf/1905.07129.pdf

[2]ACL 2019将会有哪些值得关注的论文？：https://www.zhihu.com/question/324223170/answer/686289852

[3]ACL 2019 | 基于知识增强的语言表示模型，多项NLP任务表现超越BERT：http://mr