分享主题：ERNIE 和 RoBERTa

ERNIE相关的论文

> **ERNIE: Enhanced Language Representation with Informative Entities** 清华大学&华为诺亚 ACL 2019

结合大规模语料库和**知识图谱**来增强预训练模型的语义表达能力，在原基础上增加**实体对齐**任务

> **ERNIE: Enhanced Representation through Knowledge Integration** 百度

引入了三种mask的方式，分别对Token，**Entity**，**Phrase**进行mask；还引入了对话语料，构建了一个**DLM**的任务。

> **ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding** 百度 AAAI 2020

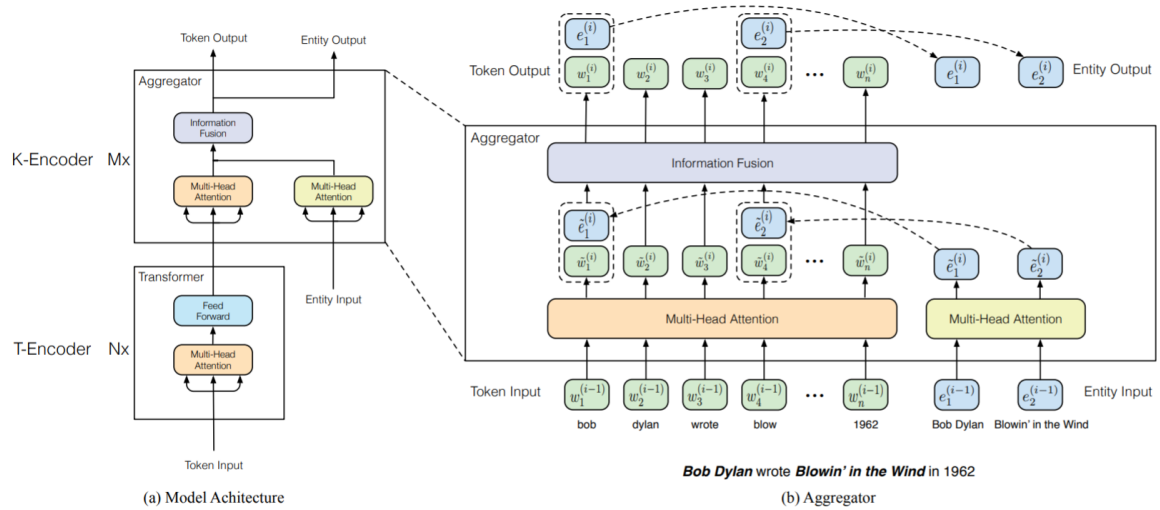
多任务连续训练和Task Embedding；先训练任务1，保存模型，然后加载保存模型，再同时训练任务1和任务2，依次类推，到最后同时训练7个任务。

ERNIE（清华）

主要创新点

> - 引入知识图谱
> - 引入实体对齐任务
>
> **抽取并编码的知识信息：**识别文本中的命名实体，然后将提到的实体与知识图谱中的实体进行匹配
>
> **目标：**随机 Mask 掉一些对齐了的输入文本的命名实体，并要求模型从知识图谱中选择合适的实体以完成对齐

模型架构



(a) Model Achitecture　　(b) Aggregator

T-Encoder:

- 底层文本编码器，获取输入标记中的基本词汇和语法信息
- 输入信息：token embedding 、 segment embedding 、 positional embedding

K-Encoder:

- 知识编码器，将知识信息集成到底层的文本信息中，就可以将token和实体的异构信息表示到一个统一的特征空间中
- 输入：lexical and syntactic features、 entity embedding
- 输出：token output 和 entity output（取top aggregator 的输出最为最终的输出）

Information Fusion：

- 具有对齐实体的token的融合计算：（将实体与其命名实体短语中的第一个token对齐）

$$
\begin{aligned}
\boldsymbol{h}_j &= \sigma(\tilde{\boldsymbol{W}}_t^{(i)}\tilde{\boldsymbol{w}}_j^{(i)} + \tilde{\boldsymbol{W}}_e^{(i)}\tilde{\boldsymbol{e}}_k^{(i)} + \tilde{\boldsymbol{b}}^{(i)}), \\
\boldsymbol{w}_j^{(i)} &= \sigma(\boldsymbol{W}_t^{(i)}\boldsymbol{h}_j + \boldsymbol{b}_t^{(i)}), \\
\boldsymbol{e}_k^{(i)} &= \sigma(\boldsymbol{W}_e^{(i)}\boldsymbol{h}_j + \boldsymbol{b}_e^{(i)}).
\end{aligned}
$$

- 没有对齐实体的token计算：

$$
\begin{aligned}
\boldsymbol{h}_j &= \sigma(\tilde{\boldsymbol{W}}_t^{(i)}\tilde{\boldsymbol{w}}_j^{(i)} + \tilde{\boldsymbol{b}}^{(i)}), \\
\boldsymbol{w}_j^{(i)} &= \sigma(\boldsymbol{W}_t^{(i)}\boldsymbol{h}_j + \boldsymbol{b}_t^{(i)}).
\end{aligned}
$$

- **参数构成**：

  N=6，*M*=6，*Hw*=768，*He*=100，*Aw*=12，*Ae*=4，Total parameters = 114M

  注： *Hw*是token embeddings的隐藏层维度，*He*是entity embeddings的维度

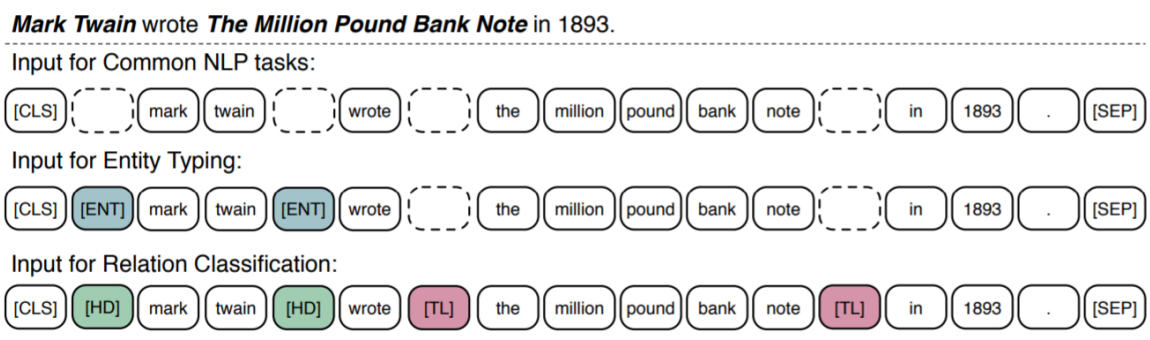
K-Encoder预训练任务：

- mask策略：对于给定的token-entity， 5% 的时间将原实体随机替换成其它实体，以便于训练模型时更正对齐错误；15% 时间 mask token-entity 对齐。基于对齐的token预测所有对应的实体。
- 给定token序列以及{w1, . . . , wn}以及对应的实体序列{e1, . . . , em}，对齐实体对于token wi 的概率分布计算：

$$
p(e_j|w_i) = \frac{\exp(\texttt{linear}(\boldsymbol{w}_i^o)\cdot\boldsymbol{e}_j)}{\sum_{k=1}^{m}\exp(\texttt{linear}(\boldsymbol{w}_i^o)\cdot\boldsymbol{e}_k)},
$$

- 考虑到实体数量过大对softmax的影响，只需要系统基于给定实体序列来预测实体，而不是KGs中的所有实体。
- 训练的loss：K-Encoder + MLM + NSP

Fine-tuning：



- **ENT**：entity mention，引导ERNIE更细致的组合上下文信息和entity mention的信息。
- **HD & TL**：在传统的关系分类模型中扮演着类似位置嵌入的角色，标识头实体和尾实体。

ERNIE 百度

主要创新点：

> - 增加基于Entity，和Phrase的mask 策略
> - 扩展语料库，新增对话语料并构建了DLM任务

多策略的mask方式

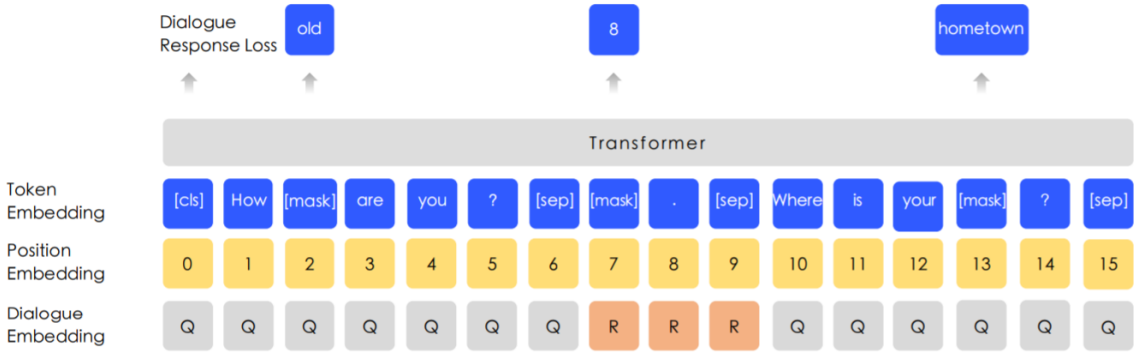
Sentence	Harry	Potter	is	a	series	of	fantasy	novels	written	by	British	author	J.	K.	Rowling
Basic-level Masking	[mask]	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	J.	[mask]	Rowling
Entity-level Masking	Harry	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	[mask]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

模型结构及其参数设置

- 该模型结构与BERT一样是由多个Transformer 编码 层堆叠而成
- 选用了和 BERT-base相同的模型大小：12 encoder layers，768 hidden units和12 attention heads

DLM任务：

随机生成一些假的多轮QR对，然后让模型去预测当前的多轮对话是真实的还是假的。



ERNIE 2.0百度

主要创新点：

> 多任务（7个子任务）连续学习

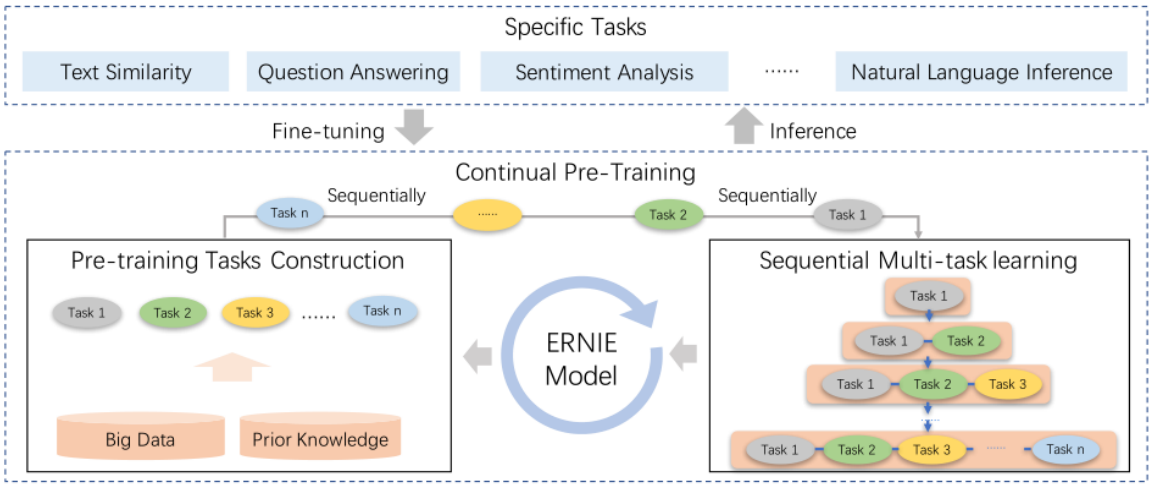
模型结构

- 利用大量的数据和先验知识构造无监督预训练任务
- 通过连续的多任务学习逐步更新ERNIE模型

多任务训练的有效性如何保证：

- 如何在不忘记以前所学知识的情况下连续不断地训练任务
- 如何以有效的方式预先训练这些任务

是先训练任务1，保存模型，然后加载刚保存的模型，再同时训练任务1和任务2，依次类推，到最后同时训练7个任务



RoBERTa：站在BERT的肩膀上

基于BERT的一种改进版本。

主要改进策略

> 用更大批次、更多的数据对模型进行更长时间的训练
>
> 去掉NSP 任务
>
> 在更长的序列上进行训练
>
> 动态改变应用于训练数据的mask模式

数据

- BOOKCORPUS 和英文维基百科：原始 BERT 的训练集，大小 16GB 。
- CC-NEWS：包含2016年9月到2019年2月爬取的6300万篇英文新闻，大小 76 GB（经过过滤之后）。
- OPENWEBTEXT：从 Reddit 上共享的 URL （至少3个点赞）中提取的网页内容，大小 38 GB 。
- STORIES：CommonCrawl 数据集的一个子集，包含 Winograd 模式的故事风格，大小 31GB。

文本编码

- 原始的BERT：character-level BPE vocabulary of size 30K
- RoBERTa： byte-level BPE vocabulary containing 50K subword unit
- 改进后的编码方式仍然可以编码任何的输入文本，不会引入“un-known” 标记

静态mask 与 动态mask

- 原始的BERT：只在数据预处理的时候随机选择15%的token按照80%、10%、10%的策略进行mask，没有与下游任务进行匹配。
- RoBERTa：将每一个样本复制10次，每一份在都在被输入模型时按照原始策略进行随机mask。相同mask的序列被训练N/10次。
- 这种mask策略对更多步骤或更大数据集进行预处理时至关重要。

Masking	SQuAD 2.0	MNLI-m	SST-2
reference	76.3	84.3	92.8
Our reimplementation:			
static	78.3	84.3	92.5
dynamic	78.7	84.0	92.9

Table 1: Comparison between static and dynamic masking for $\text{BERT}_{\text{BASE}}$. We report F1 for SQuAD and accuracy for MNLI-m and SST-2. Reported results are medians over 5 random initializations (seeds). Reference results are from Yang et al. (2019).

模型输入与NSP任务的讨论

bert模型原始的任务：

预测两个concatenated document segments是否来自同一个document

多种输入结构：

- SEGMENT-PAIR+NSP：与原始的bert模型一致。最长512个token
- SENTENCE-PAIR+NSP：包含一对自然句子，要么从一个文档的连续部分采样，要么从单独的文档采样。
- FULL-SENTENCES：从一个或者多个document中连续采样的完整句子，直到最大长度512。如果采样到达document末尾，从下一个文档中采样，并添加分隔符：去掉NSP loss
- DOC-SENTENCES：与FULL-SENTENCES类似，但是不跨document，遇到document末尾就结束采样，输入长度可能 < 512

Model	SQuAD 1.1/2.0	MNLI-m	SST-2	RACE
Our reimplementation (with NSP loss):				
SEGMENT-PAIR	90.4/78.7	84.0	92.9	64.2
SENTENCE-PAIR	88.7/76.2	82.9	92.1	63.0
Our reimplementation (without NSP loss):				
FULL-SENTENCES	90.4/79.1	84.7	92.5	64.8
DOC-SENTENCES	90.6/79.7	84.7	92.7	65.6
$\text{BERT}_{\text{BASE}}$	88.5/76.3	84.3	92.8	64.3
$\text{XLNet}_{\text{BASE}}$ (K = 7)	–/81.3	85.8	92.7	66.1
$\text{XLNet}_{\text{BASE}}$ (K = 6)	–/81.0	85.6	93.4	66.7

更大批次训练

- 过去在神经机器翻译方面的工作表明，当学习速率适当增加时，用非常大的mini-batches训练可以提高优化速度和最终任务性能。
- 原始的BERT：11 M steps，256 batch size

bsz	steps	lr	ppl	MNLI-m	SST-2
256	1M	1e-4	3.99	84.7	92.7
2K	125K	7e-4	**3.68**	**85.2**	**92.9**
8K	31K	1e-3	3.77	84.6	92.8

- 大批量训练改善了掩蔽语言建模目标的困惑，也提高了最终任务的准确性。通过分布式数据并行训练，大批量也更容易并行化，论文中采用的batch size大小为 8K 。

实验结果

GLUE SQuAD RACE

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
$\text{BERT}_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
$\text{XLNet}_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**	**91.3**	-
Ensembles on test (from leaderboard as of July 25, 2019)										
ALICE	88.2/87.9	95.7	**90.7**	83.5	95.2	92.6	**68.6**	91.1	80.8	86.3
MT-DNN	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2/89.8	98.6	90.3	86.3	**96.8**	**93.0**	67.8	91.6	**90.4**	88.4
RoBERTa	**90.8/90.2**	**98.9**	90.2	**88.2**	96.7	92.3	67.8	**92.2**	89.0	**88.5**

Table 5: Results on GLUE. All results are based on a 24-layer architecture. $\text{BERT}_{\text{LARGE}}$ and $\text{XLNet}_{\text{LARGE}}$ results are from Devlin et al. (2019) and Yang et al. (2019), respectively. RoBERTa results on the development set are a median over five runs. RoBERTa results on the test set are ensembles of *single-task* models. For RTE, STS and MRPC we finetune starting from the MNLI model instead of the baseline pretrained model. Averages are obtained from the GLUE leaderboard.

Model	SQuAD 1.1 EM	SQuAD 1.1 F1	SQuAD 2.0 EM	SQuAD 2.0 F1
Single models on dev, w/o data augmentation				
$\text{BERT}_{\text{LARGE}}$	84.1	90.9	79.0	81.8
$\text{XLNet}_{\text{LARGE}}$	**89.0**	94.5	86.1	88.8
RoBERTa	88.9	**94.6**	**86.5**	**89.4**
Single models on test (as of July 25, 2019)				
$\text{XLNet}_{\text{LARGE}}$			86.3†	89.1†
RoBERTa			86.8	89.8
XLNet + SG-Net Verifier			**87.0**†	**89.9**†

Table 6: Results on SQuAD. † indicates results that depend on additional external training data. RoBERTa uses only the provided SQuAD data in both dev and test settings. $\text{BERT}_{\text{LARGE}}$ and $\text{XLNet}_{\text{LARGE}}$ results are from Devlin et al. (2019) and Yang et al. (2019), respectively.

Model	Accuracy	Middle	High
Single models on test (as of July 25, 2019)			
$\text{BERT}_{\text{LARGE}}$	72.0	76.6	70.1
$\text{XLNet}_{\text{LARGE}}$	81.7	85.4	80.2
RoBERTa	**83.2**	**86.5**	**81.3**

Table 7: Results on the RACE test set. $\text{BERT}_{\text{LARGE}}$ and $\text{XLNet}_{\text{LARGE}}$ results are from Yang et al. (2019).