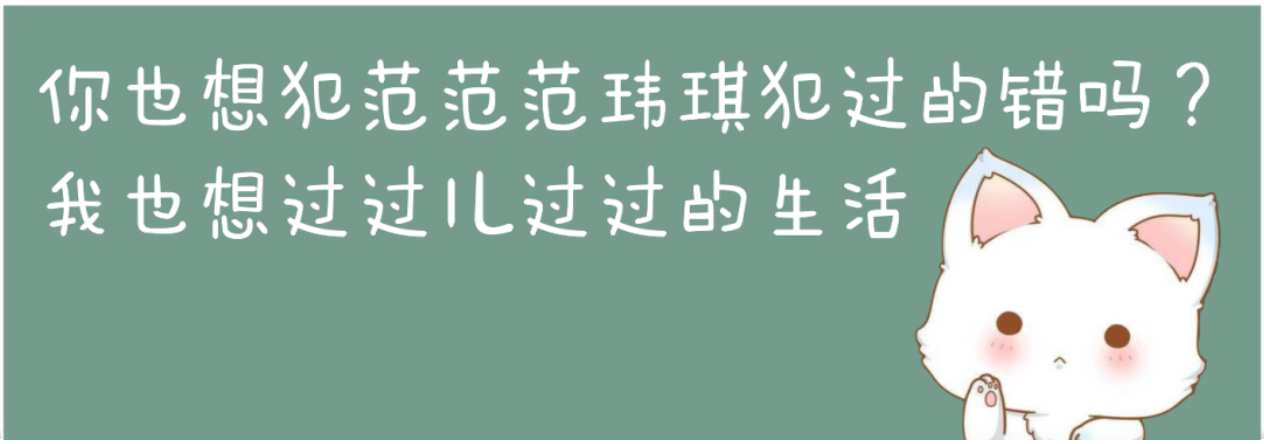
中文BERT上分新技巧，多粒度信息来帮忙

原创 ivem 夕小瑶的卖萌屋 2021-06-01 22:20



文 | ivem

自然语言处理实在是太难啦！中文尤其难！

相比于英文，**中文是以词作为语义的基本单位的**，因此传统的中文 NLP 都需要先进行分词。分词这步就劝退了很多人，比如“研究生活很充实”，怎么让模型分出“研究|生活”，而不是“研究生”呢？

随着预训练模型的到来，中文模型通常直接用字作为输入。甚至 19 年的一篇 ACL[1] 给出结论：基于“字”的模型要好于基于“词”的模型。但是，中文是以词作为语义的基本单位的呀，忽略这种粗粒度的信息，真的合理吗？

今天这篇发表在 NAACL 2021 的文章就让 BERT 在预训练中**学到了字和词的信息**，在自然语言理解的多个任务上，相对字级别的模型取得了性能提升，轻松摘得 SOTA。以后做中文任务想要刷分，可以直接拿来换掉自己的 BERT😺。

这篇文章为了让 BERT 学到字和词的信息，解决了三个问题：

1. 怎么将字和词的信息融合，送入 BERT？
2. 字和词有重叠，位置编码怎么设计？
3. 在 MLM 任务上，怎么才能同时将字和词的信息都 mask 掉？

下面就来看看这篇文章的解决办法吧~

> **论文题目:**
> **Lattice-BERT: Leveraging Multi-Granularity Representations in Chinese Pre-trained Language Models**
> **论文链接:**
> *http://arxiv-download.xixiaoyao.cn/pdf/2104.07204v1.pdf*

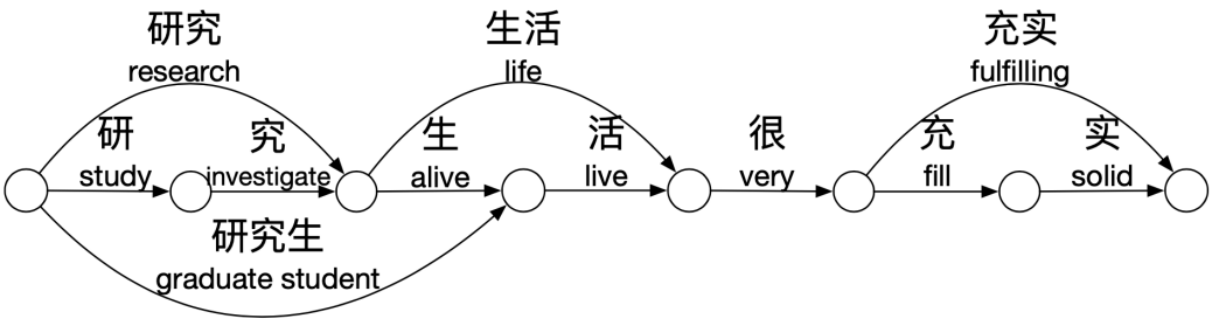
> 代码地址：
> *https://github.com/alibaba/AliceMind/tree/main/LatticeBERT*

方法

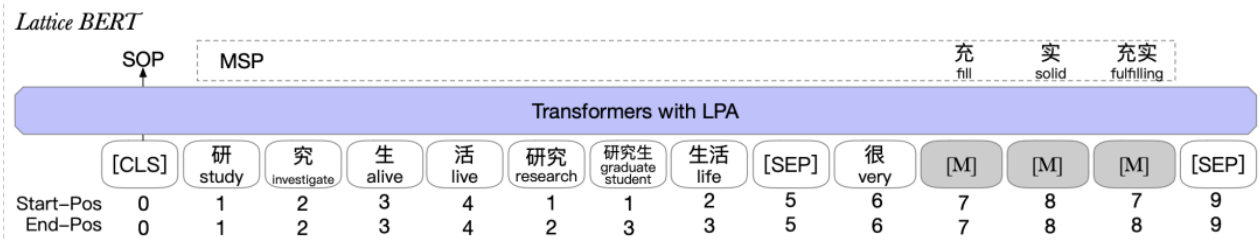
词格输入

为了解决问题 1，本文是将词格（word lattice）输入 BERT。

中文的词格图（lattice graph）是一个有向无环图，包含了句子里字和词的所有信息。以“研究生活很充实”这句话为例，词格如下图所示：



读到这里可能会有人疑惑了：BERT 只能处理序列呀？这样的有向无环图该怎么被 BERT 处理呢？简单！这篇文章直接将词格图中各粒度的信息“拍平”，得到一个线性序列，作为 BERT 的输入。其中的每一项无论是字还是词，我们都称为 token：



词格注意力机制

“拍平”词格的输入，就会造成不可避免的重复和冗余，那么对于位置编码，该怎么适应呢？另外，在“拍平”之后，原先二维的复杂图结构信息就会有所损失，怎样避免图结构的损失呢？为了解决问题 2，这篇文章又设计了新的词格注意力机制。

对于字级别的 BERT，计算 attention map 可以表达为两个字向量的内积：

$$\alpha_{ij} = \frac{1}{\sqrt{2d_k}} \left(h_i^{in,l} W^{q,l} \right) \left(h_j^{in,l} W^{k,l} \right)^T$$

其中 $h_i^{in,l}, h_j^{in,l}$ 分别是第 i 和 j 个字在第 l 层的表示。字级别 BERT 中，位置编码是在输入时，直接加到字的表示中的：

$$\tilde{h}_i^{in,0} = h_i^{in,0} + P_i$$

然而，很多工作 [2] 表明，这种在输入中混合位置编码的方式比较粗糙。在计算 attention map 时，将位置编码与字的表示解耦，专门设计一个位置编码的函数，会是一个更好的选择：

$$\tilde{\alpha}_{ij}^l = \alpha_{ij} + f(i,j)$$

这里 $f(i,j)$ 就是关于 i,j 两个字的位置编码的函数。本文也采取了这一类方法。具体地，attention map 可以通过四项相加的方式得到：

$$\tilde{\alpha}_{ij} = \alpha_{ij} + \text{att}_{ij} + b_{ij} + r_{ij}$$

第一项是字的表示得到的 attention score，后面三项都是与位置编码相关的，下面我就来一一介绍~

1. 绝对位置编码

$$\text{att}_{ij} = \frac{1}{\sqrt{2d_k}}\left(\left[P_{s_i}^S; P_{e_i}^E\right]W^q\right)\left(\left[P_{s_j}^S; P_{e_j}^E\right]W^k\right)^T$$

绝对位置编码表示了 token 在句子中的位置。式子里的 P^S 表示当前输入 token 的开始位置，P^E 表示结束的位置。这个式子就表示将 token 的起始位置的绝对位置编码拼接，进行 attention 操作。

这一项可以说是对原始 BERT 中的位置编码的复刻，并适应了词格的输入。因为词格输入的每一项长度是不固定的，引入头尾位置也是自然的想法。

然而，绝对位置编码是有缺陷的：在理论上，我们对绝对位置编码的限制只有一点，即不同位置的编码不同。但这样就忽略了很多信息，比如，位置 1 和 2 的距离与位置 5 和 6 的距离应该一样，位置 1 和 3 的距离比位置 4 和 10 的距离要小，等等。在绝对位置编码的设计里，我们只能让 BERT 隐式地“学习”。

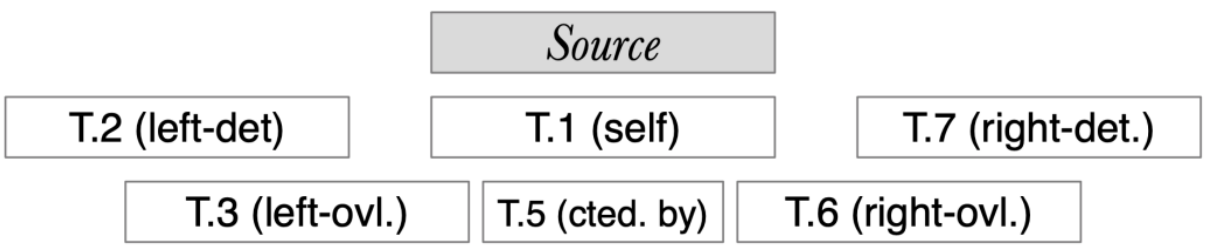
2. 相对位置编码

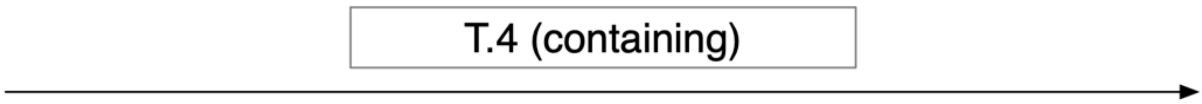
$$b_{ij} = b_{s_j - s_i}^{ss} + b_{s_j - e_i}^{se} + b_{e_j - s_i}^{es} + b_{e_j - e_i}^{ee}$$

因此，这篇文章也引入了相对位置编码，来表示 token 之间的相对距离。式子右边每一项都代表两个 token i,j 的起始位置之间的相对距离，例如，$b_{s_j - s_i}^{ss}$ 表示两个 token 的起始位置之间的相对距离 $s_j - s_i$ 的表示。引入了相对位置编码，模型就可以建模更长的文本。

3. 层叠关系编码

r_{ij} 表示两个 token 之间的层叠关系。根据这两个 token 起始相对位置的不同，两个 token 可以分成下列七种关系：





具体来说，这七种关系为：

1. 自身
2. 在左边，且无重叠
3. 在左边，且有重叠
4. 包含关系
5. 被包含关系
6. 在右边，且有重叠
7. 在右边，且无重叠

将 token 之间的关系分成以上七种，就可以显式地表示词格图中的复杂的二维关系。之前“拍平”词格图时削弱的信息，在这里又找回来了。

预训练任务：整段预测

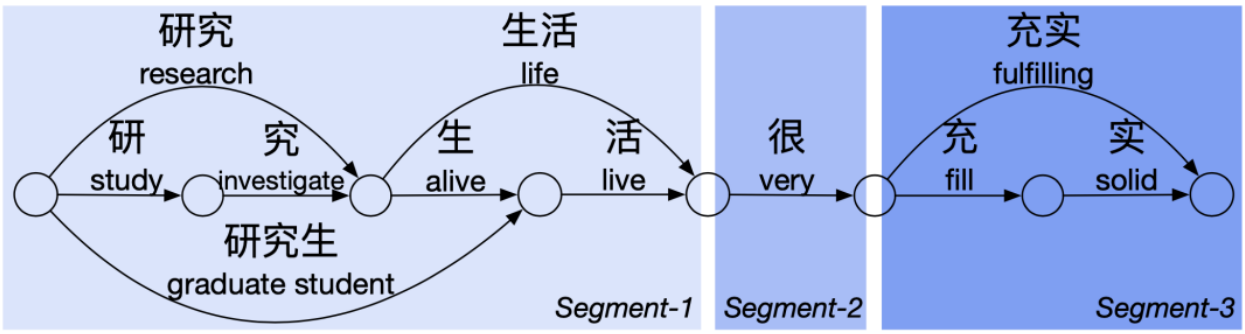
最后一个问题：原来的 MLM 任务在词格输入的形式上，似乎并不适用。

还是用“研究生活很充实”来举个例子。这句话的词格输入将是这样：

> 研 究 生 活 研究 研究生 生活 很 充 实 充实

词格的输入带来了冗余，在 MLM 任务中，我们随机 mask 掉一些 token，是希望通过其上下文预测这些 token。但是在词格输入里，比如我们随机 mask 掉了“研究”，但是模型会直接通过前面的“研”“究”和后面的“研究生”来预测这个 mask token，这样走捷径，最终一定得不到好结果。

于是，这篇文章设计了整段预测任务（masked segment prediction）：在词格图中，一句话将被切成多个段（segment），每个段之间不会有重叠的 token，同时也要使段的长度最小。“研究生活很充实”这句话就可以切成下图的三段：



在整段预测任务中，直接 mask 掉一段里的所有 token，并预测这些 token。这样就可以避免输入的冗余让模型“作弊”。

实验

这篇文章使用句子里所有可能的词来构建词格图，这样尽管会带来错误的分词，但是让模型自己学习降噪，还能提升模型的鲁棒性。

这篇文章在 11 个任务上进行了实验，11 个任务包括:

- 6 个文本分类任务：长文本分类、短文本分类、关键词提取、指代消解、自然语言推断和文本匹配；
- 2 个序列标注任务：分词和命名实体识别；
- 3 个问答任务：机器阅读理解（答案段选取）、选择题、完形填空。

总体性能如下图所示：

Task	CLUE-Classification							CLUE-MRC				Seq. Labeling		avg.
	NLI	TC		SPM	CoRE	KwRE	avg.	MRC			avg.	CWS	NER	
	CMNLI	TNEWS	IFLY.	AFQMC	WSC.	CSL		CMRC	ChID	C^3		MSR	MSRA	
base-size settings														
RoBERTa	80.5	67.6	60.3	74.0	76.9	84.7	74.0	**75.2**	83.6	66.5	75.1	98.2	96.8	78.5
NEZHA	81.1	67.4	59.5	74.5	-	83.7	-	72.2	84.4	71.8	76.1	-	-	-
BERT-word	80.0	68.2	60.0	73.5	75.5	85.2	73.7	41.3	80.9	67.0	63.1	-	-	-
AMBERT	**81.9**	**68.6**	59.7	73.9	78.3	**85.7**	74.7	73.3	**86.6**	69.6	76.5	-	-	-
BERT-Our	80.3	67.7	62.2	74.0	79.3	81.6	74.2	72.7	84.1	68.6	75.1	98.4	96.5	78.7
LBERT	81.1	68.4	**62.9**	**74.8**	**82.4**	84.0	**75.6**	74.0	**86.6**	**72.7**	**77.8**	**98.6**	**97.1**	**80.2**
lite-size settings														
BERT-Our	77.9	66.7	60.7	72.1	62.4	78.7	69.7	68.3	78.7	61.6	69.5	98.1	95.5	74.6
LBERT	79.1	68.2	61.9	72.4	70.0	81.9	72.3	69.9	81.3	63.6	71.6	98.4	96.2	76.6

其中，RoBERTa 是哈工大的 roberta-base-wwm-ext；NEZHA 是最好的字级别中文预训练模型，来自华为诺亚方舟研究院；AMBERT 是曾经多粒度中文预训练模型的 SOTA，是字节跳动李航组的工作；BERT-word 是使用词作为输入的 BERT；LBERT 是本文的方法；BERT-our 是本文使用相同语料重新预训练的 BERT。

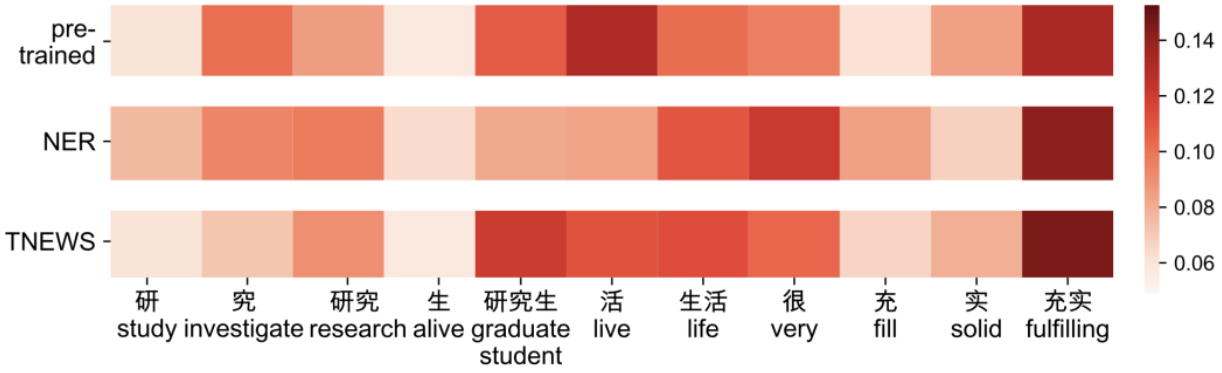
可以发现，LBERT 优于所有字级别的预训练模型，并在 7/11 个任务上取得 SOTA。

LBERT 在哪里强于字级别的 BERT 呢？作者对预测结果进行分析，得到如下结论：

- 在短文本分类任务上，LBERT 在更短的样本上有更大的性能提升，作者认为，词格输入的冗余信息为短文本提供了更丰富的语义信息；
- 在关键词提取任务上，LBERT 在词级别的关键词上性能提升更高，作者认为 LBERT 从词格输入中，理解了关键词的语义；

- 在命名实体识别任务上，LBERT 在重叠实体的样本上减少了 25% 的错误，这是词格输入带来的天然优势；

LBERT 是怎么运用多粒度的信息呢？作者对注意力分数进行了可视化，还用“研究生活很充实”这句话为例：



图中的三行分别为：

- 在预训练结束后，模型会关注句子的各个部分；
- 在命名实体识别任务上 fine-tune 之后，模型更关注“研究”“生活”“很”“充实”，这与正确的分词结果是一致的，对命名实体识别任务也是非常关键；错误分词的“研究生”就没有得到注意力；
- 在文本分类任务上 fine-tune 之后，模型更关注“研究生”“生活”“充实”，尽管这些词不能在一套分词中同时存在，但是对分类都是有用的。

总结

这篇文章解决了三个问题：

1. 怎么输入？使用词格（lattice）作为 BERT 的输入；
2. 位置编码？设计了词格注意力机制（lattice position attention），使模型真正习得词格整张图的信息；
3. MLM？设计了整段掩码预测任务（masked segment prediction），避免模型从词格的多粒度输入中使用捷径。

这样一来，就能在 BERT 中融合字和词信息，也在多个任务上拿到 SOTA。

另外，这种词格的输入看上去也是优点多多：对于短文本的任务，词格输入可以作为一种信息的增强；对于和词相关的任务，输入的词能让模型更好的理解语义；对于抽取的任务，词格能帮助定位抽取的边界。

这里还延伸出一个问题：英文是不是也可以利用多粒度的信息呢？中文的预训练模型可以使用字和词的信息，相似地，英文就可以使用 subword 和 word 信息，这样是不是有效呢？

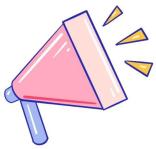
萌屋作者：ivem

在北大读研，目前做信息抽取，对低资源、图网络都非常感兴趣。希望大家在卖萌屋玩得开心 ヾ(=･ω･=)o

作品推荐

1. **老板让我用少量样本 finetune 模型，我还有救吗？急急急，在线等！**
2. **谷歌：CNN击败Transformer，有望成为预训练界新霸主！LeCun却沉默了...**

寻求报道、约稿、文案投放：
添加微信xixiaoyao-1，备注“商务合作”



后台回复关键词【**入群**】

加入卖萌屋NLP/IR/Rec与求职讨论群

后台回复关键词【**顶会**】

获取ACL、CIKM等各大顶会论文集！

参考文献

[1] Yuxian Meng, et al., "Is Word Segmentation Necessary for Deep Learning of Chinese Representations?", ACL 2019, http://arxiv-download.xixiaoyao.cn/pdf/1905.05526.pdf

[2] Guolin Ke, et al., "Rethinking Positional Encoding in Language Pre-training", ICLR 2021, http://arxiv-download.xixiaoyao.cn/pdf/2006.15595.pdf