FLAT：中文NER屠榜之作！

原创 JayLou娄杰 夕小瑶的卖萌屋 2020-09-07 22:20

收录于话题
#卖萌屋@自然语言处理

69个

星标/置顶小屋，带你解锁
最萌最前沿的NLP、搜索与推荐技术

文 | JayLou娄杰
编 | YY

近年来，引入词汇信息逐渐成为提升中文NER指标的重要手段。ACL2020中一篇来自复旦大学邱锡鹏老师团队的 *FLAT: Chinese NER Using Flat-Lattice Transformer* 刷新了中文NER任务的新SOTA。

View F1 Models not using extra training data Edit

RANK	MODEL	F1	PRECISION	RECALL	EXTRA TRAINING DATA	PAPER	CODE	RESULT	YEAR
1	FLAT+BERT	96.09			×	FLAT: Chinese NER Using Flat-Lattice Transformer			2020
2	Glyce + BERT	95.54	95.57	95.51	×	Glyce: Glyph-vectors for Chinese Character Representations			2019
3	ZEN (Init with Chinese BERT)	95.25			×	ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations			2019
4	ERNIE 2.0 Large	95			×	ERNIE 2.0: A Continual Pre-training Framework for Language Understanding			2019
5	FLAT	94.12			×	FLAT: Chinese NER Using Flat-Lattice Transformer			2020
6	ERNIE	93.8			×	ERNIE: Enhanced Representation through Knowledge Integration			2019

如上图所示，在MSRA-NER任务中，FLAT+BERT登顶榜首；而单独的FLAT（1层TransFormer）也超越了预训练模型ERNIE。相比于之前引入词汇信息的中文NER工作，FLAT主要创新点在于：

- 基于Transformer设计了一种巧妙position encoding来融合Lattice结构，可以无损的引入词汇信息。
- 基于Transformer融合了词汇信息的动态结构，支持并行化计算，可以大幅提升推断速度。

下面让我们看看FLAT是如何登顶榜首的～

论文链接：

https://arxiv.org/pdf/2004.11795.pdf

开源代码：

https://github.com/LeeSureman/Flat-Lattice-Transformer

Arxiv访问慢的小伙伴也可以在**【夕小瑶的卖萌屋】**订阅号后台回复关键词**【0907】**下载论文PDF~

背景

中文NER为什么要引入词汇信息？

不同于英文NER，中文NER通常以字符为单位进行序列标注建模。这主要是由于中文分词存在误差，导致 **基于字符** 通常要好于 **基于词汇** （经过分词）的序列标注建模方法。

那中文NER是不是就不需要词汇信息呢？答案当然是否定的。引入词汇信息的好处在于：

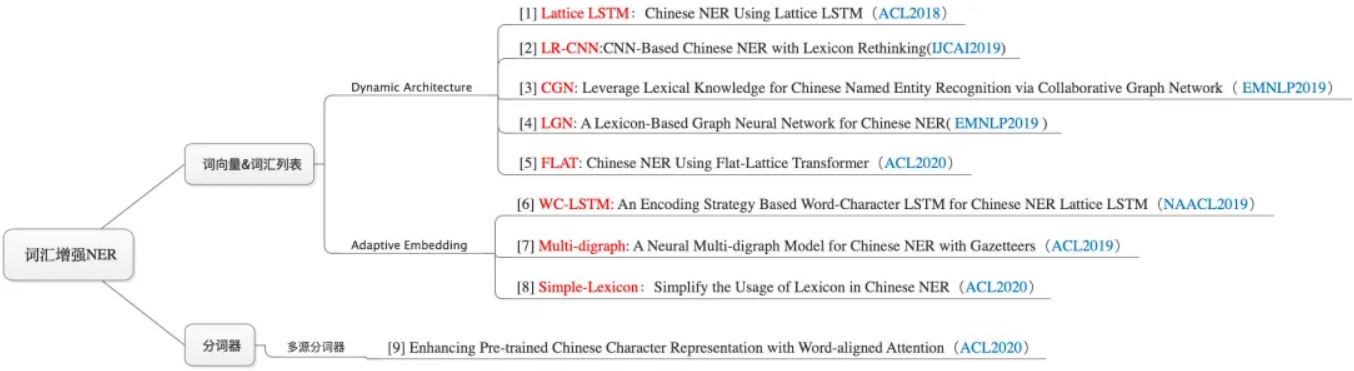
1. 引入词汇信息往往可以强化实体边界，特别是对于span较长的实体边界更加有效。
2. 引入词汇信息也是一种增强方式。对于NLP分类任务增益明显的数据增强方法，往往不能直接应用于NER任务，并且指标增益也极为有限。相反，引入词汇信息的增强方式对于小样本下的中文NER增益明显。

> 下文将**引入词汇信息增强中文NER性能**的方法称为 **词汇增强**。

词汇增强的方式有哪些？

1. **词向量&词汇列表**：利用一个具备良好分词结果的词向量；亦或者不再利用词向量，仅利用词汇或者实体边界信息，通常可通过图网络提取相关信息。这种增强方式，主要有2大范式：
   - **Dynamic Architecture**：设计一个动态抽取框架，能够兼容词汇输入；本文所介绍的FLAT就属于这一范式。
   - **Adaptive Embedding**：基于词汇信息，构建自适应Embedding；与模型框架无关。ACL2020中的 *Simplify the Usage of Lexicon in Chinese NER*[1] 就属于这一范式，仅仅在embedding层融合词汇信息，对于词汇信息的引入更加简单有效，采取静态加权的方法可以提前离线计算。
2. **分词器**：单一的分词器会造成边界错误，可以引入多源分词器并pooling不同分词结果。ACL2020中有篇处理中文预训练的文章[2]就将多种分词结果中词汇信息pooling对齐到字符编码中。

如何在中文NER模型中引入词汇信息，是近年来中文NER的一个研究重点。下图展示了各大顶会中**词汇增强NER**的主要进展：

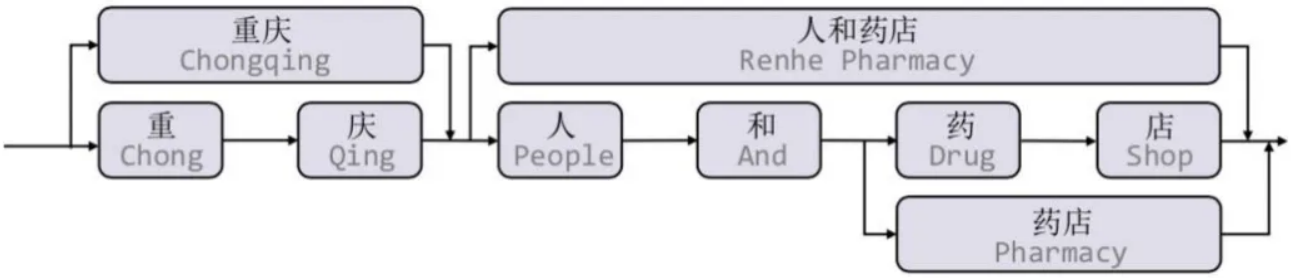


由于篇幅所限，本文将包含FLAT在内多种词汇增强方式进行了对比，感兴趣的同学可以进一步阅读有关文献。

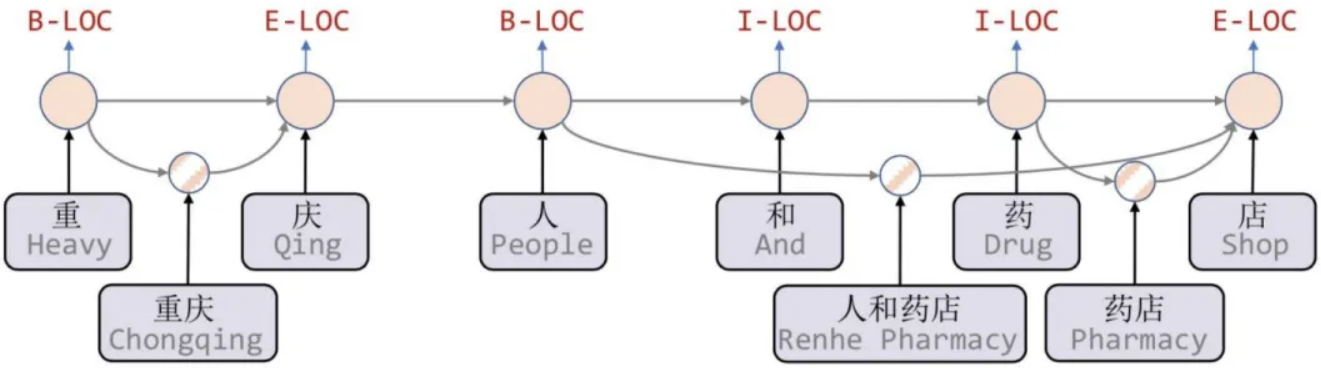
词汇增强范式	方法	特点	存在问题
Dynamic Architecture 设计相应结构以融入词汇信息	Lattice LSTM	开篇之作，设计兼容的LSTM将词汇信息引入中文NER任务	推断效率低，无法捕捉长距离依赖，存在一定程度的信息损失问题。
	LR-CNN	采取CNN进行堆叠编码，采取rethink机制解决词汇冲突问题	
	CGN	构建基于协作的图网络（GAN），充分利用词汇信息	将NER任务转化为node分类任务，但需要RNN作为底层编码器来捕捉顺序性，结构复杂。
	LGN	构建局部和全局聚合的图网络，充分利用词汇信息	
	FLAT	通过设计位置向量引入词汇信息，利用transformer捕捉长距离依赖、提高推断效率。	
Adaptive Embedding 模型无关，具备可迁移性	WC-LSTM	通过四种encoding策略对Lattice LSTM输入静态编码	存在信息损失，仍然采取LSTM进行编码
	Multi-digraph	引入实体词典，通过多图结构更好地显示建模字符和词典的交互	
	Simple-Lexicon	通过Soft-lexicon方法引入词汇信息，简单直接	

Lattice LSTM

要想更系统的理解FLAT，就必须掌握Lattice LSTM[3]这篇论文，这是针对中文NER任务引入词汇信息的开篇之作。文中提出了一种Lattice LSTM用于融合词汇信息。如下图所示，当我们通过词汇信息（词典）匹配一个句子时，可以获得一个类似Lattice的结构。



Lattice是一个有向无环图，词汇的开始和结束字符决定了格子位置。Lattice LSTM结构则融合了词汇信息到原生的LSTM中：



如上图所示，Lattice LSTM引入了word cell结构，对于当前的字符，融合以该字符结束的所有word信息，如「店」融合了「人和药店」和「药店」的信息。对于每一个字符，Lattice LSTM采取注意力机制去融合个数可变的word cell单元，其主要的数学形式化表达为：

$$\boldsymbol{c}_j^c = \sum_{b \in \{b' | w_{b',j}^d \in \mathbb{D}\}} \boldsymbol{\alpha}_{b,j}^c \odot \boldsymbol{c}_{b,j}^w + \boldsymbol{\alpha}_j^c \odot \tilde{\boldsymbol{c}}_j^c$$

本文不再堆砌繁杂的数学公式，具体看参考原论文。需要指出的是，当前字符有词汇融入时，则采取上述公式进行计算；如当前字符没有词汇时，则采取原生的LSTM进行计算。虽然Lattice LSTM有效提升了NER性能，但也存在一些缺点：

- **信息损失**：
  - 每个字符只能获取以它为结尾的词汇信息。如对于「药」，并无法获得'inside'的「人和药店」信息。
  - 由于RNN特性，采取BiLSTM时其前向和后向的词汇信息不能共享。
  - Lattice LSTM并没有利用前一时刻的记忆向量 c_{j-1}^c ，即不保留对词汇信息的持续记忆。
- **计算性能低下，不能batch并行化**：究其原因主要是每个字符之间的增加word cell（看作节点）数目不一致；不过，后续也有工作[4] 将Lattice LSTM进行batch化。
- **可迁移性差**：只适配于LSTM，不具备向其他网络迁移的特性。

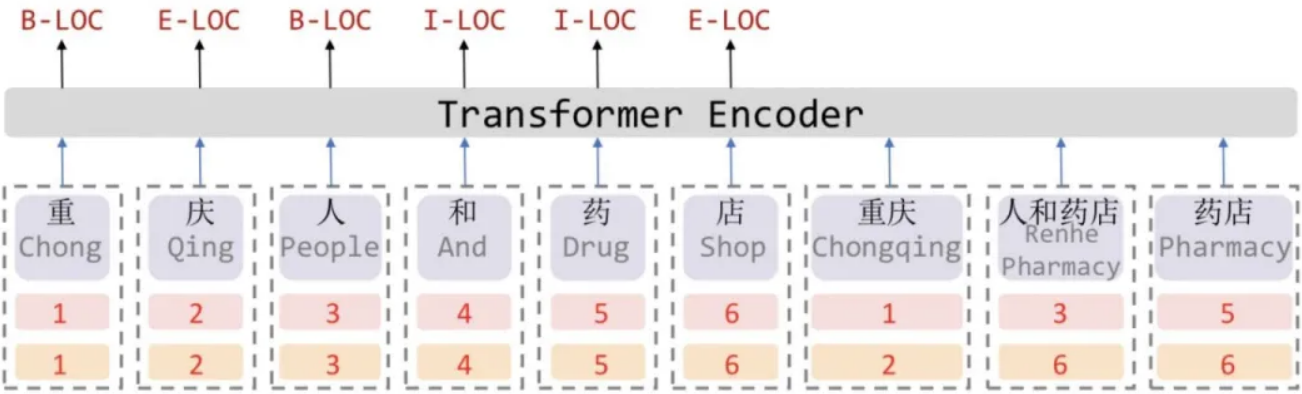
FLAT

由上文分析，Lattice-LSTM采取的RNN结构无法捕捉长距离依赖，同时引入词汇信息是有损的，同时动态的Lattice结构也不能充分进行GPU并行。

此外，有一类图网络（如CGN[5] 和LGN[6]）通过采取图网络来引入词汇信息，虽然可以捕捉对于NER任务至关重要的顺序结构，但它们通常需要RNN作为底层编码器来捕捉顺序性，模型结构更为复杂。

为解决计算效率低下、引入词汇信息有损的这两个问题，FLAT基于Transformer结构进行了两大改进：

改进1：Flat-Lattice Transformer，无损引入词汇信息

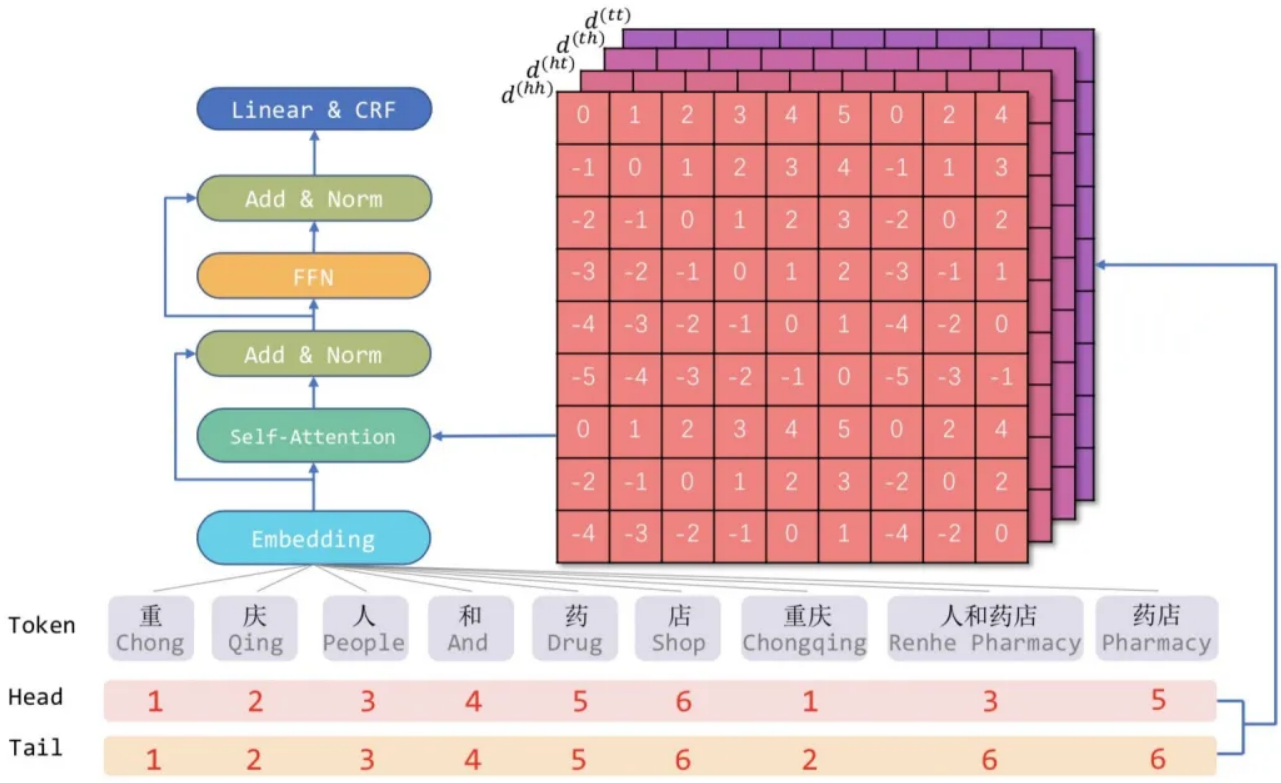


众所周知，Transformer采取全连接的自注意力机制可以很好捕捉长距离依赖，由于自注意力机制对位置是无偏的，因此Transformer引入位置向量来保持位置信息。

受到位置向量表征的启发，FLAT设计了一种巧妙position encoding来融合Lattice 结构，具体地情况如上图所示，对于每一个字符和词汇都构建两个head position encoding 和tail position encoding，这种方式可以重构原有的Lattice结构。

也正是如此，FLAT可以直接建模字符与所有匹配的词汇信息间的交互，例如，字符[药]可以匹配词汇[人和药店]和[药店]。

因此，我们可以将Lattice结构展平，将其从一个有向无环图展平为一个平面的Flat-Lattice Transformer结构，由多个span构成：每个字符的head和tail是相同的，每个词汇的head和tail是skipped的，如下图所示：



改进2：相对位置编码，让Transformer适用NER任务

FLAT使用了两个位置编码（head position encoding 和 tail position encoding），那么是否可以采用绝对位置编码呢？同样来自邱锡鹏老师组的论文*TENER: Adapting Transformer Encoder for Named Entity Recognition* [7]给出答案：**原生Transformer中的绝对位置编码并不直接适用于NER任务**。

TENER论文发现：**对于NER任务来说，位置和方向信息是十分重要的。** 如上图所示，在「Inc.」前的单词更可能的实体类型是「ORG」，在「in」后的单词更可能为时间或地点。而对于方向性的感知会帮助单词识别其邻居是否构成一个连续的实体Span。可见，对于「距离」和「方向性」的感知对于Transformer适用于NER任务至关重要。

但是，**原生Transformer的绝对位置编码本身缺乏方向性，虽然具备距离感知，但还是被self-attention机制打破了。**

仔细分析，BiLSTM在NER任务上的成功，一个关键就是BiLSTM能够区分其上下文信息的方向性，来自左边还是右边。而对于Transformer，其区分上下文信息的方向性是困难的。因此，要想解决Transformer对于NER任务表现不佳的问题，必须提升Transformer的位置感知和方向感知。

因此，FLAT这篇论文采取XLNet论文中提出相对位置编码计算attention score：

$$\boldsymbol{A}_{i,j}^{*}=\boldsymbol{W}_{q}^{T}\boldsymbol{E}_{x_i}^{T}\boldsymbol{E}_{x_j}\boldsymbol{W}_{k,E}+\boldsymbol{W}_{q}^{T}\boldsymbol{E}_{x_i}^{T}\boldsymbol{R}_{ij}\boldsymbol{W}_{k,R}+\boldsymbol{u}^{T}\boldsymbol{E}_{x_j}\boldsymbol{W}_{k,E}+\boldsymbol{v}^{T}\boldsymbol{R}_{ij}\boldsymbol{W}_{k,R}$$

（向右滑动查看完整公式）

论文提出四种相对距离表示x_i和 x_j 之间的关系，同时也包含字符和词汇之间的关系：

$$\begin{aligned}d_{ij}^{(hh)}&=head[i]-head[j]\\d_{ij}^{(ht)}&=head[i]-tail[j]\\d_{ij}^{(th)}&=tail[i]-head[j]\\d_{ij}^{(tt)}&=tail[i]-tail[j]\end{aligned}$$

$d_{ij}^{(hh)}$ 表示 x_i 的head到 x_j 的head距离，其余类似。相对位置encoding表达式为：

$$R_{ij}=\mathrm{ReLU}(W_r(\boldsymbol{p}_{d_{ij}^{(hh)}}\oplus\boldsymbol{p}_{d_{ij}^{(ht)}}\oplus\boldsymbol{p}_{d_{ij}^{(th)}}\oplus\boldsymbol{p}_{d_{ij}^{(tt)}}))$$

（向右滑动查看完整公式）

p_d的计算方式与vanilla Transformer相同：

$$\begin{aligned}\boldsymbol{p}_{d}^{(2k)}&=\sin(d/10000^{2k/d_{model}})\\\boldsymbol{p}_{d}^{(2k+1)}&=\cos(d/10000^{2k/d_{model}})\end{aligned}$$

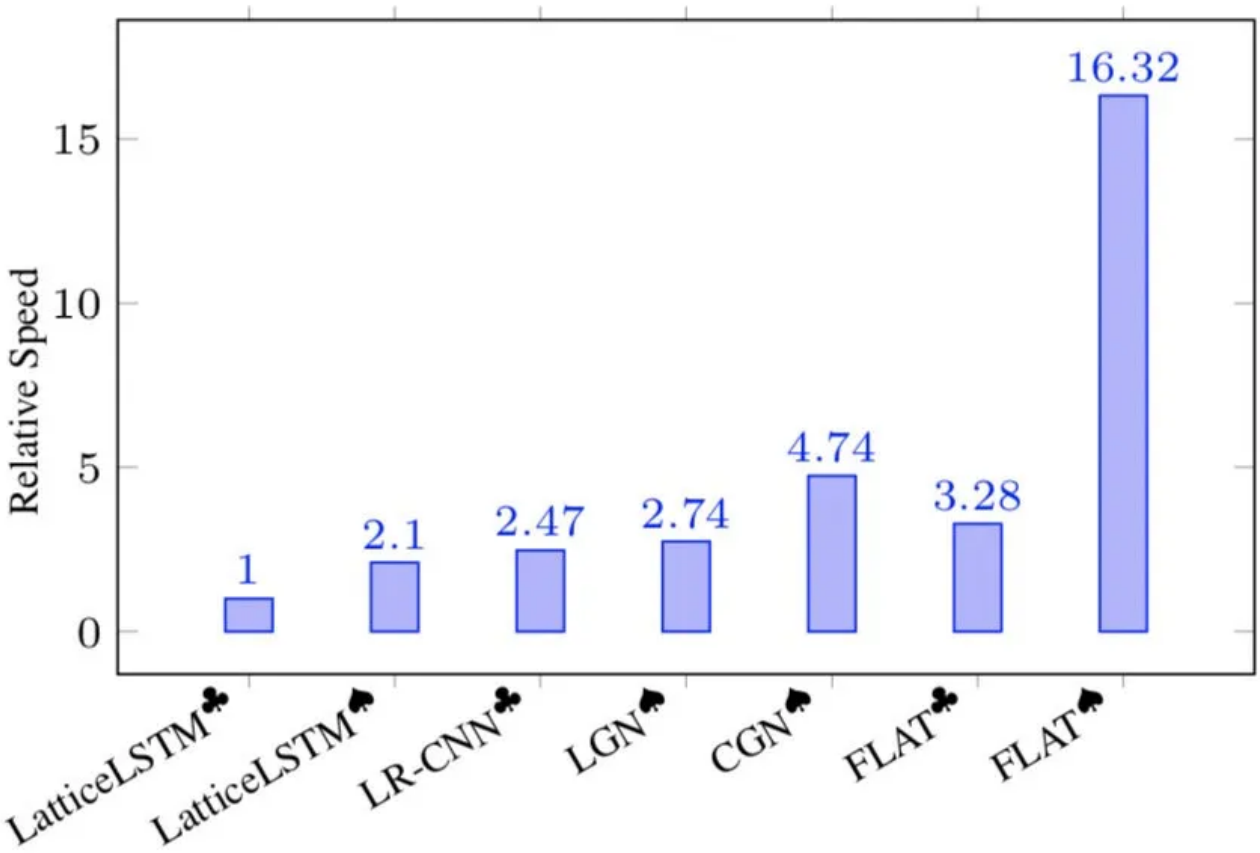
实验结果

	Lexicon	Ontonotes	MSRA	Resume	Weibo
BiLSTM	-	71.81	91.87	94.41	56.75
TENER	-	72.82	93.01	95.25	58.39
Lattice LSTM	YJ	73.88	93.18	94.46	58.79
CNNR	YJ	74.45	93.71	95.11	59.92
LGN	YJ	74.85	93.63	95.41	60.15
PLT	YJ	74.60	93.26	95.40	59.92
FLAT	YJ	**76.45**	**94.12**	**95.45**	**60.32**
FLAT_{msm}	YJ	73.39	93.11	95.03	57.98
FLAT_{mld}	YJ	75.35	93.83	95.28	59.63
CGN	LS	74.79	93.47	94.12*	63.09
FLAT	LS	**75.70**	**94.35**	**94.93**	**63.42**

	Lexicon	Ontonotes	MSRA	Resume	Weibo
BERT	-	80.14	94.95	95.53	68.20
BERT+FLAT	YJ	81.82	96.09	95.86	68.55

上图给出了论文的实验结果，具体地讲：

1. 引入词汇信息的方法，都相较于baseline模型biLSTM+CRF有较大提升。可见引入词汇信息可以有效提升中文NER性能。
2. 采用相同词表（词向量）时，FLAT好于其他词汇增强方法；
3. FLAT如果mask字符与词汇间的attention，性能下降明显，这表明FLAT有利于捕捉长距离依赖。
4. FLAT结合BERT效果会更佳。



如上图所示，在推断速度方面，FLAT论文也与其他方法进行了对比，FLAT仅仅采用1层Transformer，在指标领先的同时、推断速度也明显优于其他方法。

总结

近年来，针对中文NER如何更好地引入词汇信息，无论是Dynamic Architecture还是Adaptive Embedding，这些方法的出发点无外于两个关键点：

1. 如何更充分的利用词汇信息、最大程度避免词汇信息损失；
2. 如何设计更为兼容词汇的Architecture，加快推断速度；

FLAT就是对上述两个关键点的集中体现：FLAT不去设计或改变原生编码结构，设计巧妙的位置向量就融合了词汇信息，既做到了信息无损，也大大加快了推断速度。

本文介绍的词汇增强方式不仅应用于中文NER任务，也可进一步探索其在关系抽取、事件抽取中的有效性。

加入卖萌屋NLP/IR/Rec与求职讨论群

有顶会审稿人、大厂研究员、知乎大V和妹纸

等你来撩哦~

参考文献

[1] Simplify the Usage of Lexicon in Chinese NER:

https://arxiv.org/abs/1908.05969

[2] Enhancing Pre-trained Chinese Character Representation with Word-aligned Attention:

https://arxiv.org/abs/1911.02821

[3] Chinese NER Using Lattice LSTM:

https://arxiv.org/pdf/1805.02023.pdf

[4] Batch_Parallel_LatticeLSTM:

https://github.com/LeeSureman/Batch_Parallel_LatticeLSTM

[5] Leverage Lexical Knowledge for Chinese Named Entity Recognition via Collaborative Graph Network:

https://www.aclweb.org/anthology/D19-1396.pdf

[6] A Lexicon-Based Graph Neural Network for Chinese NER:

https://www.aclweb.org/anthology/D19-1096.pdf

[7] TENER: Adapting Transformer Encoder for Named Entity Recognition:

https://arxiv.org/abs/1911.04474