北京理工大學

自然语言理解课程报告

中文分词技术研究

Research on Chinese word segmentation technology

学　　院：计算机学院

专　　业：计算机科学与技术

学生姓名：赖昱行

学　　号：1120192236

指导教师：宋大为

2021 年 12 月 27 日

中文分词技术研究

摘　要

本文基于课程所学内容的向外拓展，选取了中文自然语言处理中的基础问题，对中文分词领域的近期研究热点和技术方法进行了研究。本文以近两年内的 3 篇顶会文章为例进行具体说明，分别讨论了文章关注的问题重点，使用的方法，实验内容，并将它们进行横向比较。

Improving Chinese Word Segmentation with Wordhood Memory Networks [1]

Improving Chinese Word Segmentation with Wordhood Memory Networks

Yuanhe Tian[♡*], **Yan Song**[♠†], **Fei Xia**[♡], **Tong Zhang**[◇], **Yonggang Wang**[♠]
[♡]University of Washington, [♠]Sinovation Ventures
[◇]The Hong Kong University of Science and Technology
[♡]{yhtian, fxia}@uw.edu [♠]clksong@gmail.com
[◇]tongzhang@ust.hk [♠]wangyonggang@chuangxin.com

图表 1 文章一

1.1 问题描述

中文语言因其特殊性，在分词时面临着以下两个主要难点。

一是歧义问题，由于中文存在大量歧义，一般的分词工具在切分句子时可能会出错。例如，“部分居民生活水平”，其正确的切分应为“部分/居民/生活/水平”，但存在“分居”、“民生”等歧义词。“他从小学电脑技术”，正确的分词是：他/从小/学/电脑技术，但也存在“小学”这种歧义词。

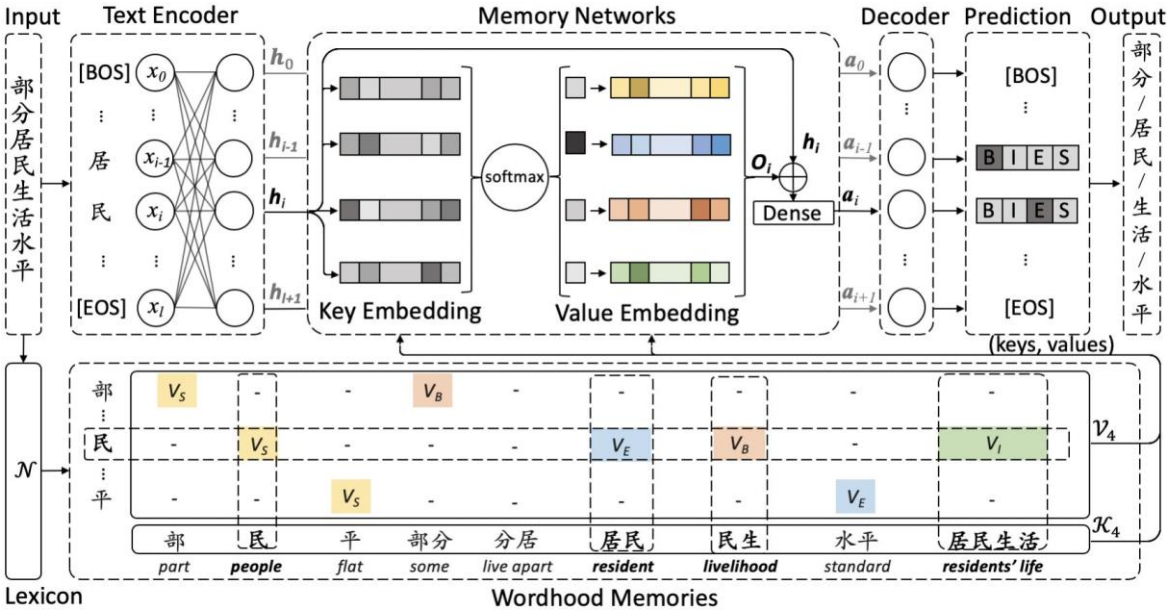
二是OOV(out of vocabulary)问题，即未登录词问题。未登录词指的是不在词表，或者是模型在训练的过程中没有遇见过的词。例如经济、医疗、科技等科学领域的专业术语或者社交媒体上的新词，或者是人名。这类问题在跨领域分词任务中尤其明显。

这篇文章站在前人的肩膀上进一步解决了上述两个问题，在发表时的2020年达到了中文分词的SOTA(state-of-the-art)水平。

1.2 模型介绍

文章提出的模型采用了中文分词的经典方案，把分词看成对token的序列化标注任务，并采用了Encoder-Decoder的传统NER模型。这里的Encoder可以是BERT或者LSTM，Decoder可以是Softmax或者CRF等。文章的创新点在于，作者在Encoder和Decoder间加入了Wordhood Memory Networks。模型的整体形式可以写成：

$$\hat{y} = \underset{y \in \tau^l}{argmax}\, p(y|X, M(X, N))$$



图表 2 WMSEG 的结构

总体来说，该模型利用n-gram提供的每个字的构词能力，通过加（降）权重实现特定语境下的歧义消解。并通过非监督方法构建词表，实现对特定领域的未标注文本的利用，进而提升对未登录词的识别。

1.2.1 Lexcion 的构建

文章中利用了一种Accesor Variety[2]的方法构建了一个n-gram词表，该词表中包含了句子中所有可能的n-gram。以“部分居民生活水平”分词为例，构建的Lexcion如下：

{“部”，“民”，“平”，“部分”，“分居”，“居民”，“民生”，“生活”，“水平”，“居民生活”}

1.2.2 Wordhood Memory Networks

这一部分是这篇文章的重点内容，该部分使用key-Value记忆网络通过键和值之间的转换来对这种成对的知识建模。

Key Addressing：这里的Key就是n-gram，对每一个汉字，首先对该句子构建Lexicon，有可能存在很多包含该汉字的n-gram。比如上面的句子中的第四个字“民”构建Lexicon，可以表示为：

K_i ={“民”，“居民”，“民生”，“居民生活”}

将这些n-gram输入embedding层之后与Encoder的输出h_i相乘，再经过softmax即可得到概率分布，概率大小即表明相关程度。

$$p_{i,j} = \frac{exp(h_i \cdot e^k_{i,j})}{\sum_{j=1}^{m_i} exp(h_i . e^k_{i,j})}$$

Value Reading：先将每个k_{ij}映射到一个值V上去，因为每个字在不同的n-gram中的位置不同，所以需要映射的值也不同，文章使用经典的BIES标记法：(B:begin，I:inside，E:end，S:single)，还是上面的例子，与K_i 对应的value集合为：

$$V_i = \{V_S, V_E, V_B, V_I\}$$

同样的，将每个value送入embedding层中，将输出与Key Addressing中的概率输出再累加来计算每个字的wordhood memory值

$$o_i = \sum_{j=1}^{m_i} p_{i,j} e^v_{i,j}$$

之后，将o_i与h_i相加接入全连接层，然后再经过Decoder解码即可得到序列标注结果，即分词结果。

1.3 实验描述

数据集：公开数据集SIGHAN 2005 Bakeoff（包含MSR、PKU、AS、CITYU）。

1.3.1 消融实验

对比模型:在目前主流的Encoder-Decoder分词模型中分别加入Wordhood Memory Networks进行实验并对比。

Config		MSR		PKU		AS		CityU		CTB6	
En-Dn	WM	F	R_{OOV}	F	R_{OOV}	F	R_{OOV}	F	R_{OOV}	F	R_{OOV}
BL-SM	×	95.53	62.96	91.85	48.84	94.52	62.21	93.79	67.26	93.56	67.39
	√	**95.61**	**63.94**	**91.97**	**49.00**	**94.70**	**64.18**	**93.88**	**69.20**	**93.70**	**68.52**
BL-CRF	×	95.80	66.17	92.35	52.04	94.39	61.59	93.96	67.84	93.84	70.81
	√	**95.98**	**68.75**	**92.43**	**56.80**	**95.07**	**68.17**	**94.20**	**69.91**	**94.03**	**71.88**
BT-SM	×	97.84	86.32	96.20	84.43	96.33	77.86	97.51	**86.69**	96.90	**88.46**
	√	**98.16**	**86.50**	**96.47**	**86.34**	**96.52**	**78.67**	**97.77**	86.62	**97.13**	88.30
BT-CRF	×	97.98	85.52	96.32	85.04	96.34	77.75	97.63	86.66	96.98	87.43
	√	**98.28**	**86.67**	**96.51**	**86.76**	**96.58**	**78.48**	**97.80**	**87.57**	**97.16**	**88.00**
ZEN-SM	×	98.35	**85.78**	96.27	84.50	96.38	77.62	97.78	90.69	97.08	86.20
	√	**98.36**	85.30	**96.49**	**84.95**	**96.55**	**78.02**	**97.86**	**90.89**	**97.22**	**86.83**
ZEN-CRF	×	98.36	**86.82**	96.36	84.81	96.39	77.81	97.81	**91.78**	97.13	87.08
	√	**98.40**	84.87	**96.53**	**85.36**	**96.62**	**79.64**	**97.93**	90.15	**97.25**	**88.46**

图表 3 WMSEG 在 SIGHAN 上的实验结果

仔细观察不难发现，在6种模型组合上分别加入Wordhood Memory Networks，5个数据集上均有提升；即使baseline的表现已经足够好，加入Wordhood Memory Networks后仍然有较大提升；在使用ZEN[3]作为Encoder时提升并不大，因为ZEN在预训练时就已经使用了n-gram的关系信息。

1.3.2 对比往年 SOTA

	MSR		PKU		AS		CITYU		CTB6	
	F	R_{OOV}	F	R_{OOV}	F	R_{OOV}	F	R_{OOV}	F	R_{OOV}
ZHANG ET AL. (2013)	97.5	-	96.1	73.1	-	-	-	-	-	-
PEI ET AL. (2014)	97.2	-	95.2	-	-	-	-	-	-	-
MA AND HINRICHS (2015)	96.6	**87.2**	95.1	76.0	-	-	-	-	-	-
CHEN ET AL. (2015)	97.4	-	96.5	-	-	-	-	-	96.0	-
XU AND SUN (2016)	96.3	-	96.1	-	-	-	-	-	95.8	-
ZHANG ET AL. (2016)	97.7	-	95.7	-	-	-	-	-	95.95	-
CHEN ET AL. (2017)	96.04	71.60	94.32	72.64	94.75	75.34	95.55	81.40	-	-
WANG AND XU (2017)	98.0	-	96.5	-	-	-	-	-	-	-
ZHOU ET AL. (2017)	97.8	-	96.0	-	-	-	-	-	96.2	-
MA ET AL. (2018)	98.1	80.0	96.1	78.8	96.2	70.7	97.2	87.5	96.7	85.4
GONG ET AL. (2019)	97.78	64.20	96.15	69.88	95.22	77.33	96.22	73.58	-	-
HIGASHIYAMA ET AL. (2019)	97.8	-	-	-	-	-	-	-	96.4	-
QIU ET AL. (2019)	98.05	78.92	96.41	78.91	96.44	76.39	96.91	86.91	-	-
WMSEG (BERT-CRF)	98.28	86.67	96.51	**86.76**	96.58	78.48	97.80	87.57	97.16	88.00
WMSEG (ZEN-CRF)	**98.40**	84.87	**96.53**	85.36	**96.62**	**79.64**	**97.93**	**90.15**	**97.25**	**88.46**

图表 4 WMSEG **的比较实验**(F-score)

实验比对了在SIGHAN上的F-score，结果表明，该文章提出的模型刷新了中文分词的记录，并且在OOV召回率上的提升非常明显。

Attention Is All You Need for Chinese Word Segmentation [4]

Attention Is All You Need for Chinese Word Segmentation

Sufeng Duan[1,2,3], **Hai Zhao**[1,2,3]*
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
[3]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
1140339019dsf@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

图表 5 文章二

传统的中文分词模型算法根据其原理和特点可以大致分为两类——基于词典的匹配方法和基于统计的分词方法。

2.1 问题描述

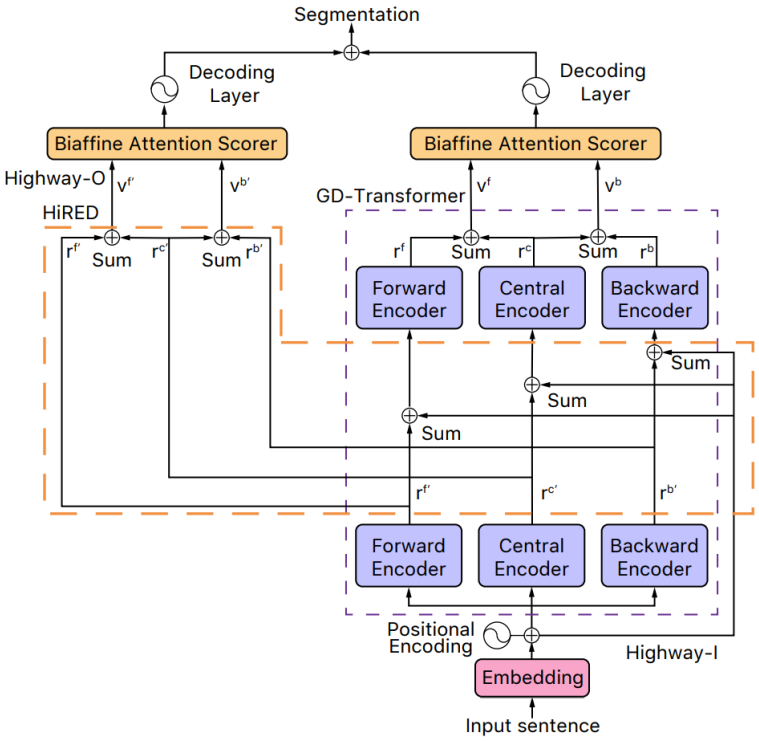
这篇文章和第一篇的不同点在于，并不是基于目前中文分词领域的某一个难点问题提出针对性的改进，而是创新性地提出了一种新型的Transformer变体——高斯掩码定向Transformer编码器（Gaussian-masked Directional Transformer encoder，GD）。并基于GD Transformer设计了一种新的中文分词模型，在其内部堆叠了注意力机制模块。作者希望用这种方式适应中文词语内部的语义联系。

作者对比了之前的工作，更多的研究人员愿意寻求多任务标注的联合模型，结合词典知识的学习方法，使用预训练模型的方法，从训练集中抽取更多信息等。但只有少数研究把重点放在增强模型本身的结构上。这部分研究提供了更高效的学习方法，可以用更少的计算复杂度从简单的数据集中训练出好的效果。

2.2 模型介绍

2.2.1 高斯掩码定向 Transformer

作者提出的GD Transformer与原版的Transformer[5]相比，做了两大改进，一是用三种平行的Encoder代替了原Transformer中的Encoder，二是采用高斯掩码定向注意力机制代替了标准的多头自注意力机制。



图表 6 基于 GD Transformer 的模型结构

如图6所示，Encoder部分每层共有三个彼此平行的编码器：前向编码器、中心编码器、后向编码器。前、后向编码器分别用于捕捉gap前边、后边的信息，中心编码器与原Transformer中的编码器一样，可以同时捕捉gap前后文的信息。作者还提到了一些技术上的细节来优化模型效果。

在Transformer中，Attention的计算公式如下：

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

而在GD Transformer中，对应的GD Attention的计算公式被改进为：

$$g_{ij} = \Phi(dis_{ij}) = \sqrt{\frac{2}{\sigma^2\pi}} \int_{-\infty}^{-dis_{ij}} exp(-\frac{x^2}{2\sigma^2})dx$$

$$AG(Q,K,V) = softmax(\frac{QK^T * G}{\sqrt{d_k}})V$$

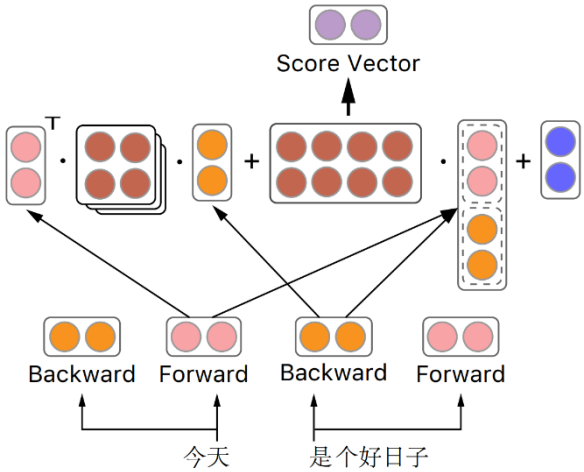
其中，高斯权重矩阵$G = g_{ij}$表示位置i和位置j的两个字符之间的关系。这样使得一个字符对其相邻字符的影响大于对不相邻字符的影响。

值得注意的是，由于作者采用了三种编码器并行，因此在把高斯权重融入注意力的时候也需要分三步进行，即针对前后向编码和中心编码采用不同的掩码矩阵，前向编码模块中，位置i应当能够见到它前方的字符序列信息，看不到它后边的信息；后

向编码则完全相反。

2.2.2 Biaffine Attention Scorer

这篇文章利用Biaffine Attention Scorer[6]来标记词间gap是否为单词边界。



图表 7 Biaffine Attention Scorer

用s_{ij}表示字符i和字符j之间是否应该分词，计算公式如下：

$$s_{ij} = Biaffinal Scorer(v_i^f, v_j^b)$$
$$= (v_i^f)^T W v_j^b + U(v_i^f \oplus v_j^b) + b$$

如图7所示，“天”的位置为i，“是”的位置为j，通过三个编码器得到v_i^f与v_j^b，再由模型计算即可得到s_{ij}，判断字符i与字符j是否应该进行分词操作。

2.2.2 Highway Connections via Hidden Layer

作者为了充分利用GD Transformer隐层状态的信息，不仅仅使用了最后一层的输出，还将每一层编码器的输出信息都进行了提取，将所有层的s_{ij}汇总得到了最终的输出。

2.3 实验描述

数据集：公开数据集SIGHAN 2005 Bakeoff（包含MSR、PKU、AS、CITYU）。

2.3.1 封闭测试结果

本文所提出的模型在保证了F-score高于baseline的情况下，在训练时长和测试时长上都体现出了较大的优势。

Models	PKU			MSR			AS			CITYU		
	F_1	Tr. (hours)	Test (sec.)	F_1	Tr. (hours)	Test (sec.)	F_1	Tr. (hours)	Test (sec.)	F_1	Tr. (hours)	Test (sec.)
(Chen et al., 2015)	**95.7**	58	105	96.4	117	120	-	-	-	-	-	-
(Cai and Zhao, 2016)	95.2	48	95	96.4	96	105	-	-	-	-	-	-
(Cai et al., 2017)	95.4	**3**	25	97.0	**6**	30	95.2	-	-	95.4	-	-
(Zhou et al., 2017)	95.0	-	-	97.2	-	-	-	-	-	-	-	-
(Ma et al., 2018)	95.4	-	-	97.5	-	-	95.5	-	-	95.7	-	-
(Wang et al., 2019a)	**95.7**	-	-	97.4	-	-	95.6	-	-	**95.9**	-	-
Our results	95.5	33	**4**	**97.6**	15	**4**	**95.7**	**67**	**10**	95.4	**17**	**1.5**

图表 8 GD Transformer 在 SIGHAN 上的封闭测试结果

2.3.2 开放测试结果

同时，在开放集合的测试中，模型取得的F-score也保持了和SOTA相一致的水平。

	PKU	MSR	AS	CITYU
(Cai et al., 2017)	95.8	97.1	95.3	95.6
(Chen et al., 2017)	94.3	96.0	94.6	95.6
(Wang and Xu, 2017)	95.7	97.3	-	-
(Zhou et al., 2017)	96.0	97.8	-	-
(Ma et al., 2018)	96.1	**98.1**	96.2	97.2
(Wang et al., 2019a)	96.1	97.5	-	-
(Huang et al., 2019)	**96.6**	97.9	**96.6**	**97.6**
Our Method	95.5	97.7	95.7	96.4

图表 9 GD Transformer 在 SIGHAN 上的开放测试结果

2.3.3 消融实验

这篇文章同样研究了去掉模型中的创新结构对结果的影响，验证了结构的合理性。

	PKU		MSR	
GD-Transformer	95.4		97.6	
-Gaussian mask	94.6	-0.8	97.1	-0.5
-Directional mask	95.1	-0.3	97.4	-0.2
Transformer	94.1	-1.3	96.5	-1.1

图表 10 Gaussian mask 和标准 mask 的对比实验

	PKU		MSR	
Our full model	95.5		97.6	
-Forward encoder	95.3	-0.2	97.4	-0.1
-Center encoder	95.3	-0.2	97.5	-0.1
-Backward encoder	95.4	-0.1	97.5	-0.2

图表 11 去掉一种 Encoder 的对比实验

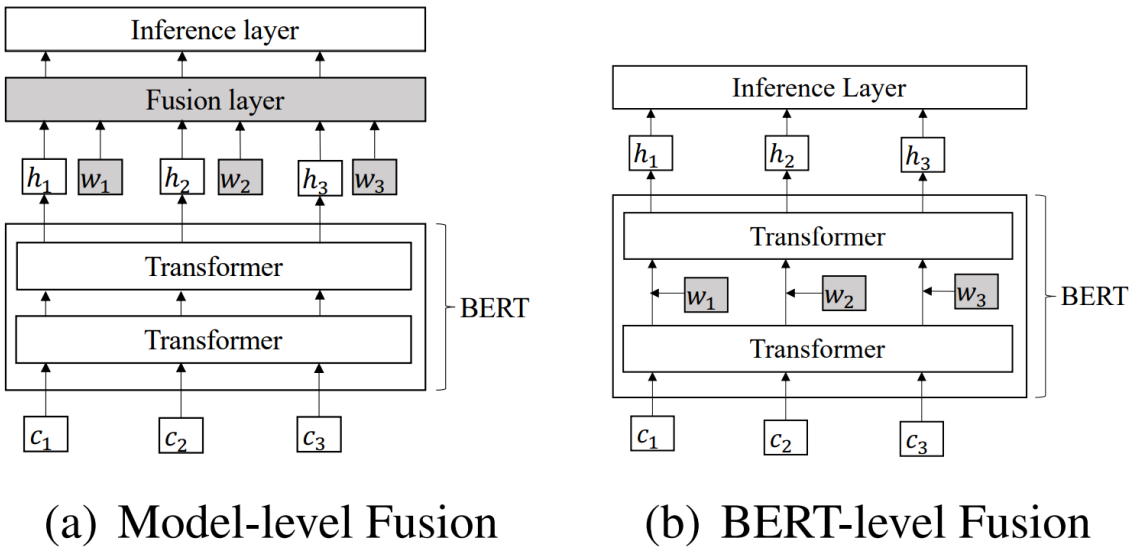
Lexicon Enhanced Chinese Sequence Labelling Using BERT Adapter [7]

Lexicon Enhanced Chinese Sequence Labeling Using BERT Adapter

Wei Liu[1], **Xiyan Fu**[2], **Yue Zhang**[3], **Wenming Xiao**[1]
[1]DAMO Academy, Alibaba Group, China
[2]College of Computer Science, Nankai University, China
[3]School of Engineering, Westlake University, China
[3]Institute of Advanced Technology, Westlake Institute for Advanced Study
hezan.lw@alibaba-inc.com, fuxiyan@mail.nankai.edu.cn,
yue.zhang@wias.org.cn, wenming.xiaowm@alibaba-inc.com

图表 12 文章三

3.1 问题描述



(a) Model-level Fusion (b) BERT-level Fusion

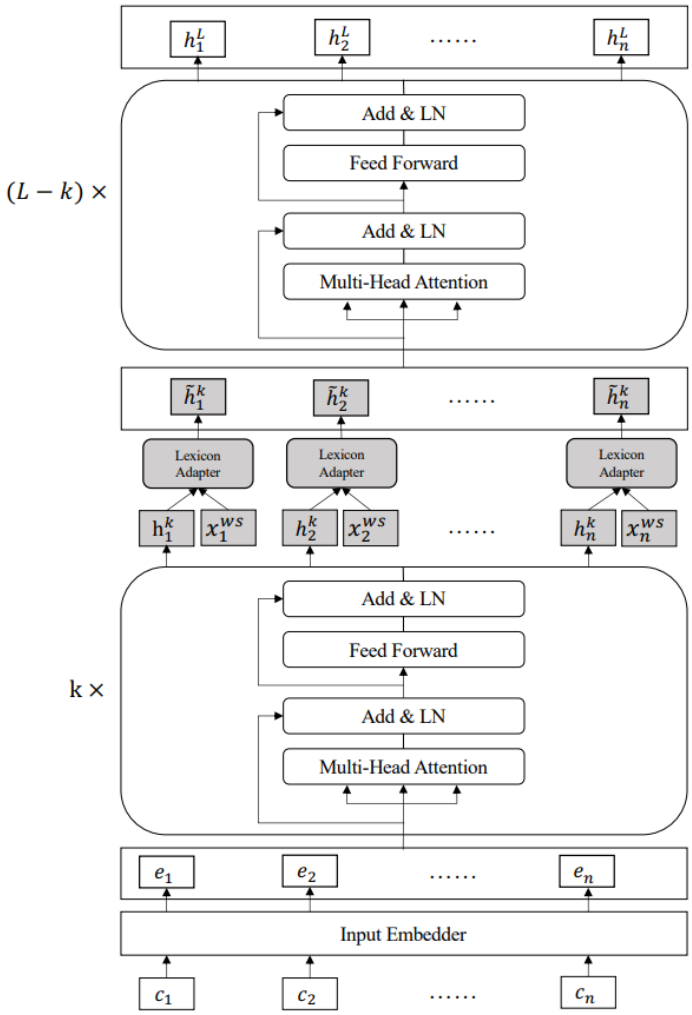
图表 13 模型层面融合和 BERT 内部融合的

词典信息和经过训练的模型（例如BERT）由于其各自的优势而经常被组合用于探索中文序列标记任务。但是，现有方法仅通过浅层和随机初始化的序列层融合词汇特征，而没有将其集成到BERT的底层。在本文中，我们提出了用于中文序列标签的Lexicon增强BERT（LEBERT），它通过Lexicon适配器层将外部词典知识直接集成到BERT层中。与已有的方法相比，我们的模型在ERT的较低层促进了深度词汇知识融合。在命名实体识别、分词和词性标注三个任务的十个中文数据集上进行实验，结果表明，LEBERT取得了最好的结果。

3.2 模型介绍

3.2.1 主要架构

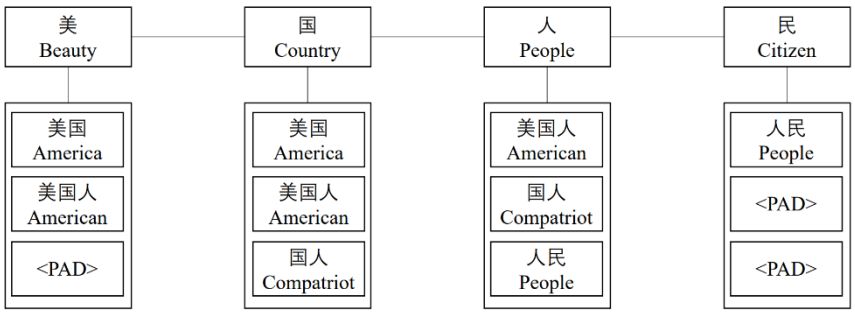
与BERT相比，LBERT有两个主要的不同之处。首先，在将中文句子转换为字词对序列的情况下，LBERT将汉字特征和词典特征都作为输入。其次，在转换器层之间附加一个词典适配器，使得词典知识能够有效地集成到BERT中。



图表 14 LEBERT 的结构

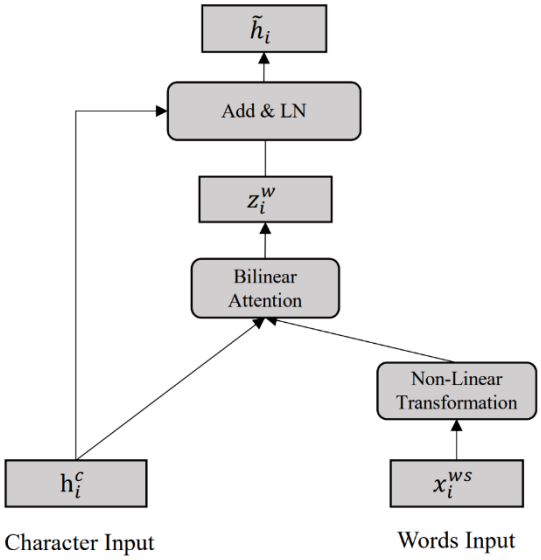
3.2.2 Char-Words Pair Sequence

类似文章一中的做法，根据已有词典，我们可以将一个句子分成很多词，给定一个句子 ’美国人民’，根据词典，我们可以得到 ’美国’,’美国人’,’国人’,’人民’ 这几个词，然后将这些词和对应的字配对起来。<PAD>用于填充矩阵空缺。



图表 15 Char-Words Pair Sequence

3.2.3 Lexicon Adapter



图表 16 Lexicon Adapter

句子中的每个位置由两类信息组成，即字级特征和词级特征。与现有的混合模型一致，作者的目标是将词典功能与BERT相结合。

如图16所示结构，h_i^c是模型的字向量，为BERT模型中某一层的输出，x_i^{ws}中的每一个元素都是一个词向量，词向量需要经过一个非线性转换，非线性转换为两层的线性层加上$tanh$激活函数。经转换后的词向量矩阵为$V_i = (v_{i1}^w, \cdots, v_{im}^w)$，通过双线性注意力层得到每个词向量对应的Attention score，a_i ，然后讲注意力权重和词向量相乘累加得到z_i^w，然后将词信息与字信息融合可得输出$\tilde{h}_i$。

3.3 实验描述

这篇文章对多个中文序列化标注的任务进行了测试，其中对于中文分词任务而言，采用了SIGHAN 2005 Bakeoff中的PKU和MSR数据集，以及CTB6数据集。

3.3.1 Overall Results

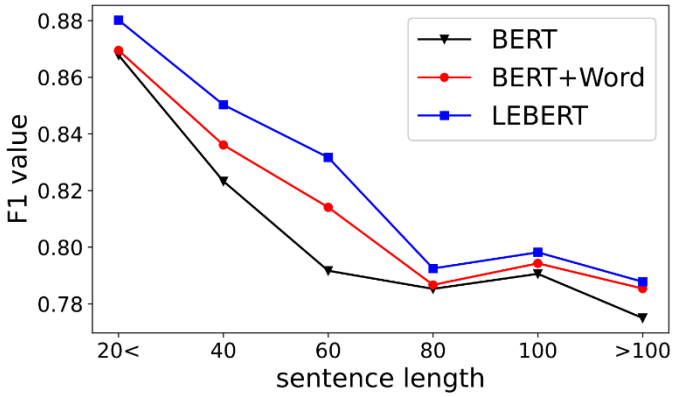
与以往中文分词的SOTA模型结果作比较，实验结果中，LEBERT表现出了优秀的效果，成功取得了中文分词领域新的SOTA。

Model	PKU	MSR	CTB6
Yang et al. (2017)	95.00	96.80	95.40
Ma et al. (2018)	96.10	97.40	96.70
Yang et al. (2019)	95.80	97.80	96.10
Qiu et al. (2020)	96.41	98.05	96.99
Tian et al. (2020c)(with BERT)	96.51	98.28	97.16
Tian et al. (2020c)(with ZEN)	96.53	98.40	97.25
BERT	96.25	97.94	96.98
BERT+Word	96.55	98.41	97.25
ERINE	96.33	98.17	97.02
ZEN	96.36	98.36	97.13
LEBERT	**96.91**	**98.69**	**97.52**

图表 17 LEBERT 在中文分词中的实验结果

3.3.2 Model-level Fusion 和 BERT-level Fusion 的对比实验

作者将模型层面之间的知识融合与BERT内部的知识融合进行了对比，结果显示LEBERT比前者更加优越。



图表 18 F1-value 的对比

分析和对比

4.1 数据集

三篇论文所使用的数据集均包括SIGHAN 2005 Bakeoff（MSR、PKU、AS、CITYU），其中文章三中还使用了CTB6。

4.2 验证指标

三篇论文均使用了F-score作为验证指标。同时，训练时长和测试时长作为文章二的研究要点，也被纳入了实验评价中。

4.3 横向比较

中文分词作为中文自然语言处理中较为成熟的领域，近期的论文基本都采用了序列化标注的方法作为基础。而后，正如文章二中所提到的，现在的研究重点在于多任务标注的联合模型，融合词典知识的学习方法，使用预训练模型的方法，从训练集中抽取更多信息的方法等，同时也有少部分如文章二直接对模型进行改进，提出新型模型。

在这三篇论文中，文章一和文章三都是想把词典知识融合到模型中。不同之处在于，文章一的融合更为浅层，是在Encoder-Decoder模型的模块间添加了一个模块；而文章三利用了预训练模型，并将融合知识的模块嵌入到了BERT中，使得知识能更充分地被模型所学习。

文章二则是利用了中文分词任务的序列化特点，以及词距与词间联系的相关性，对Transformer进行了创新性的改进，使其测试效果能在比拟SOTA的同时，大幅优化了训练时长和测试时长。

目前来看效果最好，潜力最大的应该是文章三的模型。它给了我们一种思考，能不能将多任务的联合模型，或者知识的融合，嵌入到预训练模型的过程中去？对于文章二，给我们的启发是，能不能针对特定任务的特点，对已有的基准模型进行针对性的改进？或者，上述思考也能放在同一个方法中被实现。

参考文献

[1] Tian Y, Song Y, Xia F, et al. Improving Chinese word segmentation with wordhood memory networks[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 8274-8285.

[2] Feng H, Chen K, Deng X, et al. Accessor variety criteria for Chinese word extraction[J]. Computational Lingus, 2004, 30(1): p. 75-93.

[3] Diao S， Bai J， Y Song, et al. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations[J]. 2019.

[4] Duan S, Zhao H. Attention Is All You Need for Chinese Word Segmentation[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020: 3862-3872.

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.

[6] Dozat T, Manning C D. Deep biaffine attention for neural dependency parsing[J]. arXiv preprint arXiv:1611.01734, 2016.

[7] Liu W, Fu X, Zhang Y, et al. Lexicon Enhanced Chinese Sequence Labelling Using BERT Adapter[J]. arXiv preprint arXiv:2105.07148, 2021.