文章编号：1003-0077(2019)01-0018-07

基于 Lattice-LSTM 的多粒度中文分词

张文静[1,2]，张惠蒙[1,2]，杨麟儿[1,2]，荀恩东[1,2]

(1. 北京语言大学 语言资源高精尖创新中心，北京 100083；
2. 北京语言大学 信息科学学院，北京 100083)

摘 要：中文分词是中文信息处理领域中的一项关键基础技术，而多粒度分词是中文分词领域较新的研究方向。针对多粒度中文分词任务，该文提出一种基于 Lattice-LSTM 的多粒度中文分词模型，在传统基于字的多粒度中文分词模型基础上，加入了多分词粒度的词典信息。与传统模型相比，所提出的模型在网格结构的辅助下，对不同粒度的分词标准都有较强的捕捉能力，且不局限于单一的分词标准。实验表明，该文提出的方法在多粒度中文分词方向取得了目前最好的结果。

关键词：中文分词；多粒度；Lattice-LSTM

中图分类号：TP391 **文献标识码**：A

Multi-grained Chinese Word Segmentation with Lattice-LSTM

ZHANG Wenjing[1,2], ZHANG Huimeng[1,2], YANG Liner[1,2], XUN Endong[1,2]

(1. Beijing Advanced Innovation Center for Language Resources, Beijing Language and Culture University, Beijing 100083, China; 2. School of Information Science, Beijing Language and Culture University, Beijing 100083, China)

Abstract: Chinese word segmentation is crucial to Chinese information processing. To achieve the multi-grained word segmentation, we proposed a model based on the lattice-LSTM. Multi-granularity dictionary information is added into our method comparing with the traditional character based LSTM model. With the help of lattice structure, our model has a strong ability to capture word segmentation standards with different granularities, without being confined to any word segmentation standard. Experiments show that the method proposed in this paper has reached the state-of-the-art performance in the field of multi-granularity Chinese word segmentation.

Keywords: Chinese word segmentation; multi-grained; Lattice-LSTM

0 引言

中文分词任务是中文信息处理领域中最为经典且关键的课题之一。在过去的几十年中，研究者们对其进行了不断的探索。早期，基于词表的最大长度匹配方法被广泛用来解决该问题。从 2003 年起，分词问题逐渐被视为序列标注任务[1]。最大熵[2]、条件随机场[3-5]等模型被用来解决序列标注任务。然而，传统机器学习方法一般采用特征工程的方式，依赖专家经验，成本较高。由于神经网络模型可以自动隐式提取特征，因此近年来被广泛用于解决分词任务。Zheng 等[6]提出神经网络中文分词的方法，将字向量作为输入，用一个简单的神经网络模型替代了最大熵模型[2]。Chen 等[7]提出用长短期记忆神经网络(long short-term memory neural networks，LSTM)来对句子进行建模，捕捉字与字之间的长距离依赖关系。Zhang 等[8]将基于转移的思想应用于分词任务中并取得了很好的效果。Yang 等[9]使用了大量的外部训练语料进行预训练，利用外部知识提高了分词效果。

中文分词任务在单粒度人工标注语料下取得了不错的成绩，然而，中文分词存在分词标准不统一的问题，比如，对于不同的人工标注语料，如 Microsoft

收稿日期：2018-09-29 **定稿日期**：2018-10-17

基金项目：语言资源高精尖创新中心项目(TYR17001)；国家社会科学基金(16AYY007)

Research(MSR)数据集[10]，Peking University People Daily(PPD)数据集[11]，Penn Chinese Treebank(CTB)数据集[12]，分词标准是有所区别的。表 1 给出了三种分词标准下示例句子的分词结果。

表 1　三种分词标准下示例句子的分词结果

分词标准	例句								
MSR	在	和平共处五项原则					的	基础	上
PPD	在	和平共处		五	项	原则	的	基础	上
CTB	在	和平	共处	五	项	原则	的	基础	上

Sproat 等[13]的研究表明以汉语为母语者在未提供明确标准的情况下对于词边界的识别度约为 76%。可见，单粒度中文分词对人工数据标注的工作提出了很大的挑战。并且，模型在对单粒度标注语料进行训练时，会更多地学习标注标准的特征，而不是词语的特征。Gong 等[14]构建了多粒度中文分词语料库，并利用短语句法分析和序列标注的思想来解决多粒度中文分词任务，取得了很好的结果。这一工作为中文分词任务开创了新的思路，为研究者们提供了新的研究方向。多粒度中文分词的优势在于其应用于信息检索和机器翻译等任务时具有一定的容错性，可减少分词错误对后续任务的影响。而且不同粒度的分词结果可以起到互补作用：一方面，粗粒度词语可以使模型更准确地捕获信息从而进行分析；另一方面，细粒度词语可以减少数据稀疏性并体现出对语言更深层次的理解，为后续任务打下良好基础。

基于字的多粒度中文分词模型的缺点之一是没有充分利用不同粒度的词语信息。为了弥补这一缺陷，我们把词语信息作为特征输入到模型中。由于这些词语中包含多种粒度，可以为模型提供更多的知识引导，使得模型可以更好地生成多粒度中文分词结果。本文在基于字的 LSTM 模型中加入潜在的粒度多样化的词语信息作为特征，并构建 Lattice-LSTM 来对这些词语进行建模。如图 1 所示，我们利用自动获取的词表来构建网格结构。例如，“和平共处”“和平”“共处”表示两种分词标准得到的多粒度词语结果。Zhang 等[15]的工作表明混合词典信息的网格 LSTM 结构可以建模句子从开始字到结束字的信息流动过程。我们将本文提出的模型在 Gong 等[14]构建的多粒度标注数据上进行了训练。实验表明，模型可以从上下文中自动寻找到多粒度中文分词结果，并取得很好的效果。与基于字的 LSTM 多粒度中文分词模型相比，我们的模型利用了多粒度的词语信息作为模型特征。实验结果表明，我们的模型结果要好于基于字的 LSTM 多粒度中文分词模型和基于句法分析的多粒度中文分词模型，并且在多粒度中文数据集上取得了目前最好的结果。

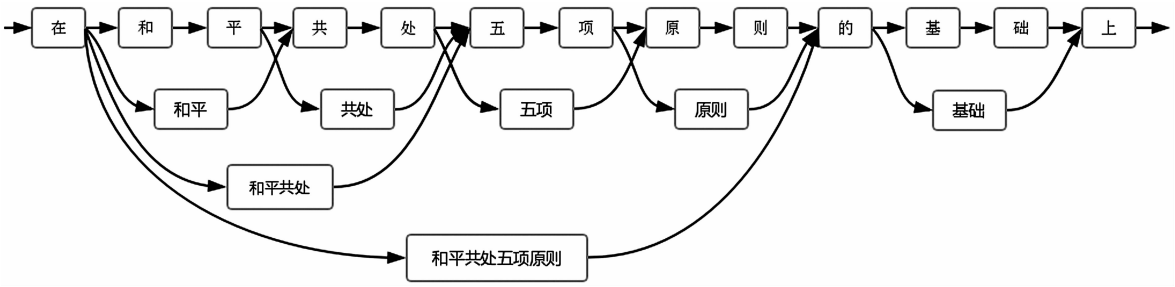


图 1　网格结构

1　模型

我们在基于字的 LSTM 模型中加入潜在的粒度多样化的词语信息并构建 Lattice-LSTM 模型。给定句子 $s=c_1,c_2,\cdots,c_m$ 作为输入，其中，c_j 代表这个句子中的第 j 个字。输出是粒度从粗到细的多粒度标签，如表 2 所示。每个词在单粒度上采用 BIES 标签体系。

表 2　多粒度标签

输入	在	和	平	共	处	五	项	原	则	的	基	础	上
字标签	S	BBB	IIE	IIB	IEE	IS	IS	IB	EE	S	B	E	S

1.1 基于字的 LSTM-CRF 模型

循环神经网络(RNN)利用隐藏状态来保存历史信息,是解决序列标注问题的一种有效方法。然而,由于梯度消失的原因,RNN 不能很好地学习到长距离依赖关系。长短期记忆网络(LSTM)在 RNN 的基础上引入记忆单元来记录状态信息,并通过三种名为输入门、遗忘门和输出门的门结构来更新隐藏状态和记忆单元。

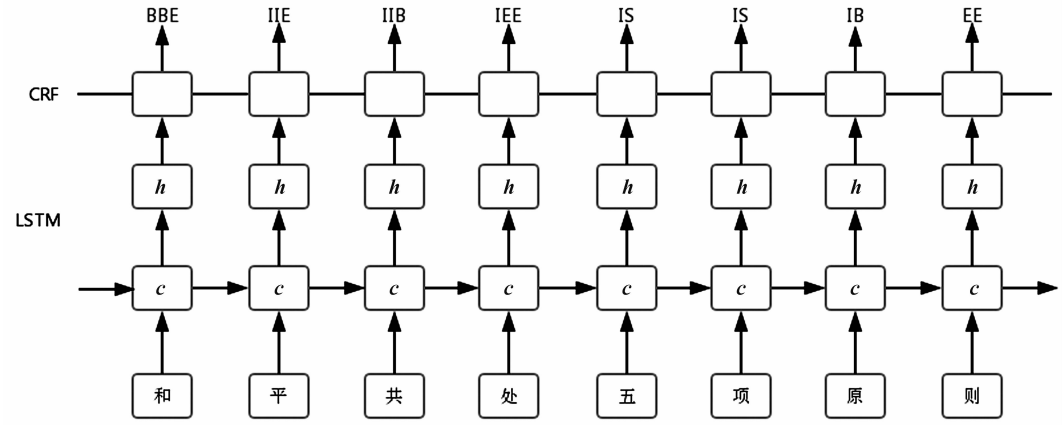
基于字的 LSTM-CRF 模型如图 2(a)所示。将字序列 $c_1, c_2, \cdots, c_m$ 作为 LSTM 的输入。我们用 $\boldsymbol{x}_j^c$ 来表示句中第 j 个字的字向量,如式(1)所示。

$$\boldsymbol{x}_j^c = \boldsymbol{e}^c(c_j) \tag{1}$$

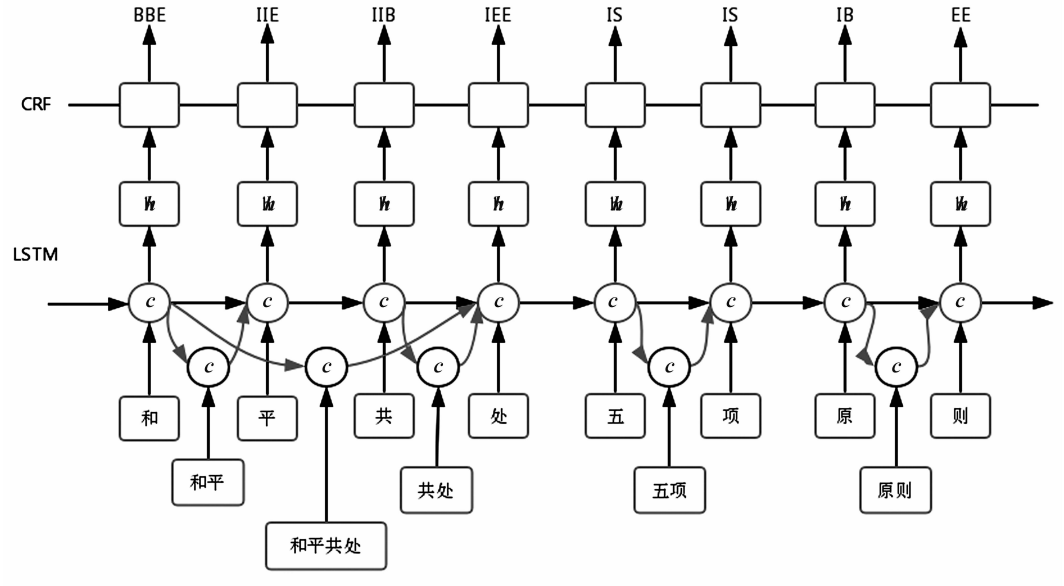
其中,$\boldsymbol{e}^c$ 表示字向量映射表。在每一时刻,LSTM 通过式(2)输入 $\boldsymbol{x}_j^c, \boldsymbol{h}_{j-1}^c, \boldsymbol{c}_{j-1}^c$,产生 $\boldsymbol{h}_j^c, \boldsymbol{c}_j^c$。

$$\begin{aligned}
\boldsymbol{i}_j^c &= \sigma(\boldsymbol{W}_i^c \boldsymbol{x}_j^c + \boldsymbol{U}_i^c \boldsymbol{h}_{j-1}^c + \boldsymbol{b}_i^c) \\
\boldsymbol{f}_j^c &= \sigma(\boldsymbol{W}_f^c \boldsymbol{x}_f^c + \boldsymbol{U}_f^c \boldsymbol{h}_{j-1}^c + \boldsymbol{b}_f^c) \\
\boldsymbol{o}_j^c &= \sigma(\boldsymbol{W}_o^c \boldsymbol{x}_j^c + \boldsymbol{U}_o^c \boldsymbol{h}_{j-1}^c + \boldsymbol{b}_o^c) \\
\tilde{c}_j^c &= \tanh(\boldsymbol{W}_{\tilde{c}}^c \boldsymbol{x}_j^c + \boldsymbol{U}_{\tilde{c}}^c \boldsymbol{h}_{j-1}^c + \boldsymbol{b}_{\tilde{c}}^c) \\
\boldsymbol{c}_j^c &= \boldsymbol{f}_j^c \odot \boldsymbol{c}_{j-1}^c \odot \tilde{c}_j^c \\
\boldsymbol{h}_j^c &= \boldsymbol{o}_j^c \odot \tanh(\boldsymbol{c}_j^c)
\end{aligned} \tag{2}$$

其中,$\boldsymbol{i}_j^c, \boldsymbol{f}_j^c, \boldsymbol{o}_j^c$ 分别代表输入门,遗忘门和输出门。$\sigma()$代表 sigmoid 函数,$\boldsymbol{W}_i^c, \boldsymbol{w}_f^c, \boldsymbol{W}_o^c, \boldsymbol{f}_{\tilde{c}}^c, \boldsymbol{U}_i^c, \boldsymbol{U}_f^c, \boldsymbol{U}_{\tilde{C}}^c, \boldsymbol{b}_i^c, \boldsymbol{b}_f^c, \boldsymbol{b}_o^c, \boldsymbol{b}_{\tilde{c}}^c$ 都是模型的参数。将 $\boldsymbol{x}_1^c, \boldsymbol{x}_2^c, \cdots, \boldsymbol{x}_m^c$ 作为模型的输入,得到 $\boldsymbol{h}_1^c, \boldsymbol{h}_2^c, \cdots, \boldsymbol{h}_m^c$。该向量序列通过式(10)被用作条件随机场(CRF)的输入,产生预测标签序列。



(a) 基于字的LSTM-CRF模型



(b) Lattice-LSTM-CRF模型

图 2 本文提出的模型

• Char+bichar

双字特征在中文分词任务中对于字特征的表示起到了很好的作用[7,9,16]。于是，通过将单字向量和双字向量进行拼接的方式，在基于字的 LSTM-CRF 模型中加入双字特征，如式(3)所示。

$$\boldsymbol{x}_j^c = [\boldsymbol{e}^c(c_j); \boldsymbol{e}^b(c_j, c_{j+1})] \tag{3}$$

其中，$\boldsymbol{e}^b$ 表示双字向量映射表。

1.2 Lattice-LSTM-CRF 模型

模型的全部结构如图 2(b)所示。我们的模型可以看作是在基于字的 LSTM-CRF 模型基础上加入词特征，并利用门结构引导信息的流动。

如图 2(b)所示，我们可以看到，模型的输入包括字序列和词序列两部分。我们用 $w_{b,e}^d$ 来表示词，其中，b 表示这个词的第一个字在句中的位置，e 表示这个词的最后一个字在句中的位置。比如，在图 1 中，$w_{2,3}^d$ 代表“和平”，$w_{2,5}^d$ 代表“和平共处”。字向量是这个模型的输入之一。我们用 $\boldsymbol{x}_j^c$ 来表示句中第 j 个字的字向量，见式(1)。循环神经网络的基本结构在式(2)中已经给出。

同基于字的 LSTM 模型相比，对于 $\boldsymbol{c}_j^c$，我们的模型不仅考虑字向量，还考虑网格中的词向量。每一个词向量可以表示为 $\boldsymbol{x}_{b,e}^w$，即：

$$\boldsymbol{x}_{b,e}^w = \boldsymbol{e}^w(w_{b,e}^d) \tag{4}$$

其中，$\boldsymbol{e}^w$ 为词向量映射表。另外，我们用 $\boldsymbol{c}_{b,e}^w$ 记录 $\boldsymbol{x}_{b,e}^w$ 的状态。$\boldsymbol{c}_{b,e}^w$ 计算方法如式(5)所示。

$$\begin{aligned}
\boldsymbol{i}_{b,e}^w &= \sigma(\boldsymbol{W}_i^w \boldsymbol{x}_{b,e}^w + \boldsymbol{U}_i^w \boldsymbol{h}_b^c + \boldsymbol{b}_i^w) \\
\boldsymbol{f}_{b,e}^w &= \sigma(\boldsymbol{W}_f^w \boldsymbol{x}_{b,e}^w + \boldsymbol{U}_f^w \boldsymbol{h}_b^c + \boldsymbol{b}_f^w) \\
\tilde{c}_{b,e}^w &= \tanh(\boldsymbol{W}_{\tilde{c}}^w \boldsymbol{x}_{b,e}^w + \boldsymbol{U}_{\tilde{c}}^w \boldsymbol{h}_b^c + \boldsymbol{b}_{\tilde{c}}^w) \\
\boldsymbol{c}_{b,e}^w &= \boldsymbol{f}_{b,e}^w \odot \boldsymbol{c}_b^c + \boldsymbol{i}_{b,e}^w \odot \tilde{c}_{b,e}^w
\end{aligned} \tag{5}$$

其中，$\boldsymbol{i}_{b,e}^w$ 和 $\boldsymbol{b}_{b,e}^w$ 代表输入门和遗忘门。

对于 $\boldsymbol{c}_j^c$，在此模型中有更多的信息输入其中。比如，在图 1 中，对于 $\boldsymbol{c}_5^c$ 的输入包括 $\boldsymbol{x}_5^c$(处)，$\boldsymbol{c}_{4,5}^w$(共处)，$\boldsymbol{c}_{2,5}^w$(和平共处)。我们先利用式(6)构建一个额外的门 $\tilde{c}_{b,e}^c$。

$$\tilde{c}_{b,e}^c = \sigma(\boldsymbol{W}_{\tilde{c}}^c \boldsymbol{x}_b^c + \boldsymbol{U}_{\tilde{c}}^c \boldsymbol{c}_{b,e}^w + \boldsymbol{b}_{\tilde{c}}^c) \tag{6}$$

之后，我们将所有的 $\boldsymbol{c}_{b,e}^w$ 以及 $\tilde{c}_j^c$ 用来计算 $\boldsymbol{c}_j^c$ 的值，计算过程如式(7)所示。

$$\boldsymbol{c}_j^c = \sum_{b \in \{b' | w_{b',j}^d \in D\}} \boldsymbol{\alpha}_{b,j}^c \odot \boldsymbol{c}_{b,j}^w + \boldsymbol{\alpha}_j^c \odot \tilde{c}_j^c \tag{7}$$

其中，对式(2)和式(6)中的 $\boldsymbol{i}_j^c$ 和 $\tilde{c}_{b,j}^c$ 进行归一化操作，使其和为 1。

$$\boldsymbol{\alpha}_{b,j}^c = \frac{\exp(\tilde{c}_{b,j}^c)}{\exp(\boldsymbol{i}_j^c) + \sum_{b' \in \{b'' | w_{b'',j}^d \in D\}} \exp(\tilde{c}_{b',j}^c)} \tag{8}$$

$$\boldsymbol{\alpha}_j^c = \frac{\exp(\boldsymbol{i}_j^c)}{\exp(\boldsymbol{i}_j^c) + \sum_{b' \in \{b'' | w_{b'',j}^d \in D\}} \exp(\tilde{c}_{b',j}^c)} \tag{9}$$

对于 $\boldsymbol{h}_j^c$，仍然用式(7)来计算。在进行训练的过程中，根据损失函数来不断优化模型参数。

2.3 CRF 层

一个标准的 CRF(层)作用在 $\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_m$ 上。标签序列 $y = l_1, l_2, \cdots, l_m$ 的生成概率如式(10)所示。

$$P(y \mid s) = \frac{\exp\left(\sum_i (\boldsymbol{w}_{\mathrm{CRF}}^{l_i} \boldsymbol{h}_i + b_{\mathrm{CRF}}^{l_{i-1}, l_i})\right)}{\sum_{y'} \exp(\boldsymbol{w}_{\mathrm{CRF}}^{l'_i} \boldsymbol{h}_i + b_{\mathrm{CRF}}^{l'_{i-1}, l'_i})} \tag{10}$$

其中，y' 是任意可能的标签。我们采用维特比算法来得到最可能的标签序列。

2 实验

2.1 数据集

对于多粒度中文分词任务，实验所用的训练语料和测试语料来源于 Gong 等[14]构建的多粒度标注数据集。该数据集由 MSR，PPD，CTB 这三个分词标准不一致的数据集以及人工标注测试数据集构成，详细信息如表 3 所示。

表 3　训练集、开发集和测试集统计信息

	训练集	开发集	测试集
句子数	138 000	5 000	1 500
字数	6 031 000	254 000	65 000

2.2 评测指标

多粒度中文分词任务的目标是对于给定的句子，尽可能地获得所有的不同分词粒度的词。因此，本文利用精确率 $\left(P = \frac{\sharp\,\mathrm{Word}_{\text{正确词个数}}}{\sharp\,\mathrm{Word}_{\text{预测词个数}}}\right)$、召回率 $\left(R = \frac{\sharp\,\mathrm{Word}_{\text{正确词个数}}}{\sharp\,\mathrm{Word}_{\text{黄金标准词个数}}}\right)$ 和 F_1 值 $\left(F_1 = \frac{2PR}{P+R}\right)$ 来进行评测。

2.3 超参数及训练设置

超参数的选择对神经网络模型有很大的影响。在我们的模型中，超参数的设置如表 4 所示。字向量、双字向量及网格中的词向量都采用随机初始化的方式，且向量维度都设为 50。LSTM 模型的隐藏层维度设为 200，层数为 1。对于字向量和网格中的

词向量都使用 Dropout[17]，且值设为 0.5。优化算法使用 SGD(随机梯度下降法)对模型优化，且初始学习率设为 0.015 并以 0.05 的速率进行衰减。我们选择在开发集上效果最好的模型来对测试集进行评测。由于时间和计算资源的限制，之后我们将会继续探索预训练词向量和词向量表示对模型的影响。

表 4 超参数值表

参数	值	参数	值
字向量维度	50	LSTM 层数	1
网格词向量维度	50	LSTM 隐藏层单元数	200
双字向量维度	50	学习率	0.015
字 Dropout	0.5	学习衰减率	0.05
网格 Dropout	0.5	参数更新方法	SGD

2.4 实验结果及分析

表 5 给出了本文提出的基于 Lattice-LSTM 的多粒度中文分词模型在多粒度中文标注数据开发集和测试集上的实验结果，且同 Gong 等[14]的实验结果进行了对比。本文提出的方法是在基于字的 LSTM 多粒度中文分词模型基础上，加入了网格结构。从表 5 中我们可以看出，我们的模型效果要好于 Gong 等[14]基于字的 LSTM 模型的效果，开发集的 F_1 值从 95.41%提高到了 96.78%。而且，表 5 的实验结果表明，同 Lattice-LSTM 模型相比，引入双字特征之后的 Lattice-LSTM 模型并没有使得开发集的 F_1 值很大的提升。这是因为虽然双字向量在一定程度上可以消除单字歧义，但是双字向量也是模棱两可的。

通过对实验结果的对比与分析，本文提出的 Lattice-LSTM 多粒度中文分词模型好于 Gong 等[14]提出的基于字的 LSTM 多粒度中文分词模型和基于句法分析的多粒度中文分词模型，并且对于引入双字特征后的 Gong 等[14]的模型，本文模型的实验效果依然显得更好。基于 Lattice-LSTM 的多粒度中文分词模型充分利用了蕴含着分词粒度多样化特征的词语信息，对多粒度中文分词任务实验效果的提升起到了一定的帮助作用。

表 5 本文模型与 Gong 等[14]模型实验结果对比

模型		开发集			测试集		
		精确率	召回率	F_1 值	精确率	召回率	F_1 值
Gong 等[14]	基于字的 LSTM 模型	95.88	94.94	95.41	96.56	94.18	95.35
	+bichar	96.86	96.26	96.59	97.01	94.96	95.97
	短语句法模型	95.58	95.04	95.51	96.37	94.11	95.22
	+bichar	96.55	96.40	96.48	97.00	95.16	96.07
我们的模型	Lattice-LSTM	97.17	96.39	96.78	97.36	95.23	96.29
	+bichar	97.24	96.64	96.94	97.16	95.22	96.39

3 相关工作

最大长度匹配方法是中文分词的经典方法，此方法基于合适的搭配词典就可以取得一定程度上可以使人接受的性能。Xue[1]是最先将分词问题转化成基于字的序列标注问题。Peng 等[4]的工作表明将条件随机场应用于基于字的序列标注模型中可以取得不错的效果。研究者们将这些方法用于神经网络模型中[6-7,18-19]。除了字特征外，词特征以及将词特征与字特征相结合的方法也被应用于神经网络模型[8,20-23]。

研究者们对树结构循环神经网络进行不断改进，形成了网格结构循环神经网络。网格 RNNs 被应用于解决机器翻译[24]、同声传译[25]、命名实体识别[15]等问题。Su 等[24]提出基于词网格的 RNN 编码器用于解决机器翻译问题。Sperber 等[25]将树结构 LSTM 转化为网格 LSTM 并融合了语音翻译的词典信息，取得了不错的效果。Zhang 等[15]提出将词典特征融入到网格结构 RNNs 中以解决中文命名实体识别问题。此方法在字序列信息的基础上充分利用了词的信息，减少了分词错误信息的传递，同时取得了很好的结果。

针对多粒度中文分词任务，我们在传统基于字

的 LSTM 模型基础上，加入了多种分词粒度的词典信息。与传统模型相比，我们把多种分词粒度的词语信息作为特征输入到模型中。这些词语中包含多种粒度的词语，可以在一定程度上为模型提供更多的知识引导。在网格结构的辅助下，本文提出的模型对不同粒度的分词标准都有较强的捕捉能力。

4　结论

本文针对多粒度中文分词任务，提出一种基于 Lattice-LSTM 的多粒度中文分词模型。在基于字的 LSTM 多粒度中文分词模型的基础上，融合了多种分词粒度的词语信息，取得了更好的效果，可以捕捉到不同粒度的分词标准。我们会探索多粒度中文分词在信息检索、机器翻译等任务上的应用。

参考文献

[1] Nianwen Xue. Chinese word segmentation as character tagging[J]. Computational Linguistics and Chinese Language Processing, 2003, 8(1), 29-47.

[2] JinKiat Low, HweeTou Ng, Wenyuan Guo. A maximum entropy approach to Chinese word segmentation[C]//Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing. Jeju Island, Korea: Association for Computational Linguistics, 2005: 448-455.

[3] John Lafferty, Andrew McCallum, Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data[C]//Proceedings of the 18th International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann, 2001: 282-289.

[4] Fuchun Peng, Fangfang Feng, Andrew McCallum. Chinese segmentation and new word detection using conditional random fields[C]//Proceedings of the 20th International Conference on Computational Linguistics. Geneva, Switzerland: Association for Computational Linguistics, 2004: 562-569.

[5] Hai Zhao, Chang-Ning Huang, Mu Li. An improved Chinese word segmentation system with conditional random field[C]//Proceedings of the 15th SIGHAN Workshop on Chinese Language Processing. Sydney, Australia: Association for Computational Linguistics, 2006: 162-165.

[6] Xiaoqing Zheng, Hanyang Chen, Tianyu Xu. Deep learning for Chinese word segmentation and POS tagging[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics, 2013: 647-657.

[7] Xinchi Chen, et al. Long short-term memory neural networks for Chinese word segmentation[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015: 1197-1206.

[8] Meishan Zhang, Yue Zhang, Guohong Fu. Transition-based neural word segmentation[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, Germany: Association for Computational Linguistics, 2016: 421-431.

[9] Jie Yang, Yue Zhang, Fei Dong. Neural word segmentation with rich pretraining[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, Canada: Association for Computational Linguistics, 2017: 839-849.

[10] Changning Huang, Yumei Li, Xiaodan Zhu. Tokenization guidelines of Chinese text (V5.0, in Chinese)[Z]. Microsoft Research Asia, 2006.

[11] Shiwen Yu, et al. Specification for corpus processing at Peking University: Word segmentation, POS tagging and phonetic notation[J]. Journal of Chinese Language and Computing, 2003, 13(2): 121-158.

[12] Nianwen Xue, et al. The Penn Chinese Treebank: Phrase structure annotation of a large corpus[J]. Natural Language Engineering, 2005, 11(2): 207-238.

[13] Richard Sproat, et al. A stochastic finite-state word-segmentation Algorithm for Chinese[J]. Computational Linguistics, 1996, 22(3): 377-404.

[14] Chen Gong, et al. Multi-grained chinese word segmentation[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, 2017: 692-703.

[15] Yue Zhang, Jie Yang. Chinese NER using lattice LSTM[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, Australia: Association for Computational Linguistics, 2018: 1554-1564.

[16] Meishan Zhang, Nan Yu, Guohong Fu. A simple and effective neural model for joint word segmentation and POS tagging[J]. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 2018, 26(9): 1528-1538.

[17] Nitish Srivastava, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 15(1): 1929-1958.

[18] Wenzhe Pei, Tao Ge, Baobao Chang. Max-margin tensor neural network for Chinese word segmentation [C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Baltimore, Maryland: Association for Computational Linguistics, 2014: 293-303.

[19] Jingjing Xu, Xu Sun. Dependency based gated recursive neural network for Chinese word segmentation [C]//Proceedings of the 54nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Berlin, Germany: Association for Computational Linguistics, 2016: 567-572.

[20] Yijia Liu, et al. Exploring segment representations for neural segmentation models[C]//Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2016: 2880-2886.

[21] Changning Huang, Hai Zhao. Which is essential for Chinese word segmentation: Character versus word [C]//Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation, 2006: 1-12.

[22] Weiwei Sun. Word-based and character-based word segmentation models: Comparison and combination [C]//Proceedings of the 23rd International Conference on Computational Linguistics. Beijing, China: Coling 2010 Organizing Committee, 2010: 1211-1219.

[23] Mengqiu Wang, Rob Voigt, Christopher D Manning. Two knives cut better than one: Chinese word segmentation with dual decomposition[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Baltimore, Maryland: Association for Computational Linguistics, 2014: 193-198.

[24] Jinsong Su, et al. Lattice-based recurrent neural network encoders for neural machine translation[C]// Proceedings of Association for the Advancement of Artificial Intelligence. 2017.

[25] Matthias Sperber, et al. Neural Lattice-to-Sequence Models for Uncertain Inputs[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, 2017: 1380-1389.

张文静(1994—),硕士研究生,主要研究领域为自然语言处理。
E-mail: 15600760036@163.con

杨麟儿(1983—),通信作者,博士,主要研究领域为句法分析。
E-mail: yangtianlin08@gmail.com

张惠蒙(1992—),硕士研究生,主要研究领域为自然语言处理。
E-mail: zinna.zhang0523@gmail.com