文章编号：1003-0077（2011）00-0000-00

基于古文语料的新词发现方法*

刘昱彤[1]，吴斌[1]，谢韬[1]，王柏[1]

（1.北京邮电大学 智能通信软件与多媒体北京市重点实验室，北京 100876）

摘要：新词发现，作为自然语言处理的基本任务，是用计算方法研究中国古代文学必不可少的一步。该文提出一种基于古文语料的新词识别方法，称为 AP-LSTM-CRF 算法。该算法分为三个步骤。第一步，基于 Apache Spark 分布式并行计算框架实现的并行化的 Apriori 改进算法，能够高效地从大规模原始语料中产生候选词集。第二步，用结合循环神经网络和条件随机场的切分概率模型对测试集文档的句子进行切分，产生切分概率的序列。第三步，用结合切分概率的过滤规则从候选词集里过滤掉噪声词，从而筛选出真正的新词。实验结果表明，该新词发现方法能够有效地从大规模古文语料中发现新词，在宋词和史记数据集上分别进行实验，F1 值分别达到了 89.68%和 81.13%，与现有方法相比，F1 值分别提高了 8.66%和 2.21%。

关键词：Apriori 的改进算法；双向长短时记忆网络；条件随机场；过滤规则；并行化

中图分类号：TP391　　**文献标识码：**A

New Word Detection in Ancient Chinese Corpus

Yutong Liu[1], Bin Wu[1], Tao Xie[1], Bai Wang[1]

(1. Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: New word detection, as a basic task in natural language processing, is an indispensable step in the study of ancient Chinese literature with computational methods. In this work, we propose AP-LSTM-CRF model to discover new words in ancient Chinese literature. This model consists of three steps. First, the parallelized improved-Apriori algorithm, implemented on Apache Spark which is a distributed parallel computing framework, is used to efficiently generate candidate character sequences from large-scale raw corpus. Second, a segmentation probability model which combines recurrent neural network and conditional random field is used to segment sentences in documents of test set to generate sequences of segmentation probability. Third, we design a filter rule based on the results of the segmentation probability model to filter out noise words in the set of candidate character sequences. Experimental results demonstrate that the method is capable of detecting new words in large-scale ancient Chinese corpus effectively. The F-measure value is up to 89.68% and 81.13% in Song Poetry dataset and History of the Song Dynasty dataset, increased by 8.66% and 2.21% compared to state-of-the-art results.

Key words: improved-Apriori algorithm; Long Short-Term Memory Networks; Conditional Random Field; filter rules; parallelizing

1 引言

挖掘语料中的新词对整个自然语言处理领域都具有十分重要的意义，并且它是中文分词、命名实体识别和其他任务的必不可少的部分。根据黄昌宁等人[1]的研究，60%的中文分词错误来源于未登录词。

如今，中文新词发现的研究主要集中在现代文语料，在古文语料上的研究鲜有涉及。然而，古文与现代文在很多方面有很大的区别，比如词汇、短语、语法结构等。此外，很多现

* **收稿日期：**　　**定稿日期：**

基金项目：国家“九七三”重点基础研究发展计划基金项目(No.2013CB329606)；国家自然科学基金项目(No.61772082)；国家社会科学基金项目(No.16ZDA055)

作者简介：刘昱彤（出生年 1996），女，本科生，主要研究领域为自然语言处理、数据挖掘、机器学习、深度学习等； 吴斌（出生年 1969），男，教授，主要研究领域为基于图的数据挖掘、智能信息处理、复杂网络、海量数据并行处理等； 谢韬（出生年 1993），男，硕士，主要研究领域为机器学习、数据挖掘、自然语言处理、社交网络分析等； 王柏（出生年 1962），女，教授，主要研究领域为通信系统软件、分布式计算、数据挖掘、智能信息处理等。

代文的词汇，在古文语料中可能不是一个词语。举例来说，“可以”在现代文语境中就表示“可以”，但是在古文语境中，这两个字是单独的词语，当“可”和“以”组合在一起时，表示“可以凭借”的意思。因此，将针对现代文的新词发现工具直接套用在古文语料中是不合理的，对古文语料的新词发现研究十分有必要。

在古文语料中，Deng 等人[2]提出了一种无监督的方法来同时挖掘新词和切分中文文本。尽管这种方法能够比较有效地切分古文语料，但仍存在一些问题。第一，分词的切分粒度不均匀，分词结果存在很多歧义。第二，这种方法对于低频新词的挖掘效果不太理想。然而，在古文语料中，很少会出现分词歧义，而且很多新词属于低频新词。基于这些原因，本文提出了一种新的古文语料的新词发现算法。

本文提出的 AP-LSTM-CRF 古文新词发现算法融合了改进的类 Apriori 算法和 Bi-LSTM-CRF 切分概率模型。改进的类 Apriori 算法能够有效地挖掘低频新词。Bi-LSTM-CRF 模型能够获得连续两个字之间的切分概率。基于切分概率的序列，改进的类 Apriori 算法产生的候选词可以被划分为新词和噪声词。

本文的主要贡献包括 3 个方面：

1) 提出了古文的新词发现算法 AP-LSTM-CRF，融合了改进的类 Apriori 算法和 Bi-LSTM-CRF 切分概率模型，利用数据挖掘的关联规则算法和深度学习的方法有效地挖掘古文语料中的新词。
2) 基于 Apache Spark 分布式并行计算框架将改进的类 Apriori 算法并行化，提高了挖掘新词的效率。提出了新的较为严格的过滤规则，能够过滤掉更多的噪声词。
3) 在宋词和史记数据集上验证了 AP-LSTM-CRF 古文新词发现算法的有效性，与现有最好方法比较，F1 值分别提高了 8 个百分点和 2 个百分点。

2 相关工作

最新的中文新词发现研究主要分为三个方面。第一方面是针对特定领域的新词发现。Chen 等人[3]提出一种联合统计模型可以同时发现特定领域的新词和在特定领域有特殊含义的词。霍帅等人[4]提出一种基于微博内容的新词发现方法，引入词关联性信息的迭代上下文熵算法，并通过上下文关系获取新词候选列表进行过滤。周霜霜等人[5]提出了一种融合人工启发式规则、C/NC-value 改进算法和条件随机场模型的微博新词抽取方法。雷一鸣等人[6]提出一种基于词语互信息模型和外部统计量的针对微博语料的新词发现方法。杜丽萍等人[7]提出的互信息(PMI)的改进算法，将 PMI^k 算法与少量基本规则相结合，能够从大规模语料中自动识别不同长度的网络新词。第二方面是针对开放领域的新词发现。陈飞等人[8]提出了一系列区分新词边界的统计特征，并采用 CRF 方法综合这些特征实现了开放领域新词发现的算法。第三方面是用新词发现来辅助情感分析任务。杨阳等人[9]提出了基于词向量的情感新词发现方法。万琪等人[10]将新词发现融入到微博情感表达抽取任务中，建立了基于 CRF 的联合模型，利用新词的信息提高了情感表达识别的效果。

然而上述的所有这些方法，都主要集中在现代文语料。

关于在古文语料中发现新词，最早的工作是 Deng 等人[2]提出的 TopWords 算法。这是一种无监督的面向特定领域的新词发现算法，在古文语料中发现新词只是其中的一个应用场景。这种算法并不是针对古文语料的特点来设计的，存在着分词歧义和无法有效挖掘低频新词的缺点。谢韬等人[11]的工作是在此基础上进行改进的。只保留了原先产生候选词集的 Apriori 算法，他们所提出的 AP-LSTM 是一种专门针对古文语料的有监督新词发现算法。

本文的工作与谢韬等人[11]的工作最为相似，都是先通过改进的类 Apriori 算法产生候选词集，然后使用过滤规则去掉候选词集里的噪声词。而主要区别分为 3 个方面。第一，为改进的类 Apriori 算法添加了并行化实现，极大地提高了产生候选词集的效率。第二，将 LSTM 切分概率模型改为了 Bi-LSTM-CRF 模型，提高了切分概率模型的准确率，召回率和 F1 值。

第三，对过滤规则进行改进，使其变得更加严格，在不影响新词的条件下过滤掉更多的噪声词，提高了发现新词的准确率。

3 模型描述

3.1 问题描述

本文的研究点主要集中在古文语料的新词上，所以需要首先明确在古文语料中，哪种词才算是新词。首先，它应该是一个含有明确语义的词语，其次，它很少出现在现代文中，最后，它应该具有鲜明的历史意义，譬如：古汉语、诗歌词汇、专有名词等等。所以，本文给出古文语料新词的定义：

定义 1. 古文语料新词

古文中的新词是一个包含了明确语义解释和历史特征的字符序列，同时它不包含在标准词典中。

基于上述定义，本文阐明了古文中新词的含义。并且本文设计了一个标准词典来过滤掉噪声词，标准词典主要包括现代词汇和停用词。进一步地，本文将古文新词发现问题形式化地描述为：

定义 2. 古文新词发现

给定古文语料 C 和标准词典 D，新词发现旨在找出所有不在 D 中的新词 W。这包括候选新词 A 的生成以及过滤掉噪声词 T。换句话说，新词集合$(A-T)$就是最终的结果。

3.2 总体流程介绍

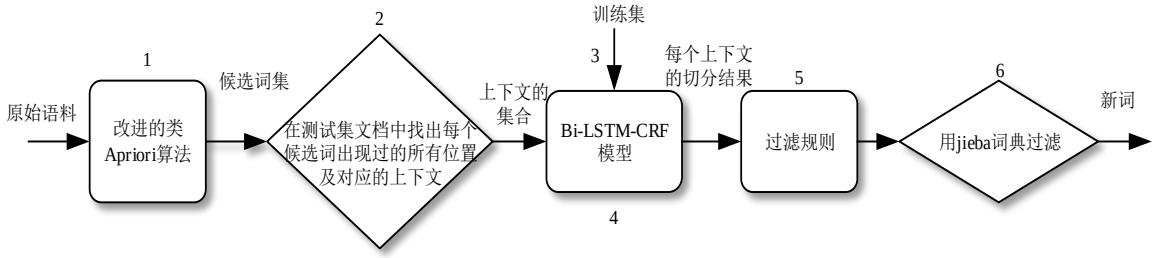

图 1 AP-LSTM-CRF 算法流程图

图 1 展示了本文提出的古文新词发现算法 AP-LSTM-CRF 的流程图，具体步骤如下：

1） 用改进的类 Apriori 算法产生候选词集。
2） 在测试集文档中找到候选词每次出现的位置和对应的上下文。
3） 用训练集文档训练 Bi-LSTM-CRF 模型。
4） 在测试集上用训练好的 Bi-LSTM-CRF 模型对句子进行切割。
5） 用设计好的结合切分概率的过滤规则把候选词集里的噪声词过滤掉。
6） 用 jieba[1]分词器的词典对上一步得到的结果进行过滤，得到真正的新词。

3.3 改进的类 Apriori 算法

改进的类 Apriori 算法来源于谢韬等人[11]的工作，现复述如下。

算法 1. 改进的类 Apriori 算法

输入：原始语料$D=\{s_1,s_2,\cdots,s_n\}$
输出：候选新词的集合
//产生 1 频繁项集

[1] https://pypi.org/project/jieba/

1 候选项集$C_1 = \{c_1, c_2, \cdots, c_m\}$
统计原始语料中单个字的频率，得到(字,频率)的二元组集合
$S_1 = \{(c_1, f_1), (c_2, f_2), \cdots, (c_m, f_m)\}$
FOR (c_i, f_i) IN S_1 DO
IF f_i>支持度 THEN
　　把c_i加入 1 频繁项集L_1
得到 1 频繁项集L_1

//产生 2 频繁项集
FOR s_i IN L_1 DO
　　FOR s_j IN L_1 DO
　　把s_i+s_j加入 2 候选项集C_2
统计原始语料中C_2的每个字符串s_i出现的频率，得到(字符串,频率)的二元组集合$S_2 = \{(s_1, f_1), (s_2, f_2), \cdots, (s_m, f_m)\}$
FOR (s_i, f_i) IN S_2 DO
　　IF f_i>支持度 THEN
　　把s_i加入 2 频繁项集L_2
　　IF f_i<低频阈值 THEN
　　把s_i加入低频项集 M
得到 2 频繁项集L_2

FOR k=3 TO K DO
//产生 k 频繁项集
FOR P$(p_1 p_2 \cdots p_{k-1})$ IN L_{k-1} DO
　　FOR Q$(q_1 q_2 \cdots q_{k-1})$ IN L_{k-1} DO
　　IF $p_2 p_3 \cdots p_{k-1} = q_1 q_2 \cdots q_{k-2}$THEN
　　把$p_1 p_2 \cdots p_{k-1} q_{k-1}$加入$C_k$
统计原始语料中C_k的每个字符串s_i出现的频率，得到(字符串,频率)的二元组集合$S_k = \{(s_1, f_1), (s_2, f_2), \cdots, (s_m, f_m)\}$
FOR (s_i, f_i) IN S_k DO
　　IF f_i>支持度 THEN
　　把s_i加入 k 频繁项集L_k
　　If f_i<低频阈值 THEN
　　把s_i加入低频项集 M
得到 k 频繁项集L_k
Return $\boldsymbol{L_2 \cup L_3 \cup \cdots \cup L_K \cup M}$

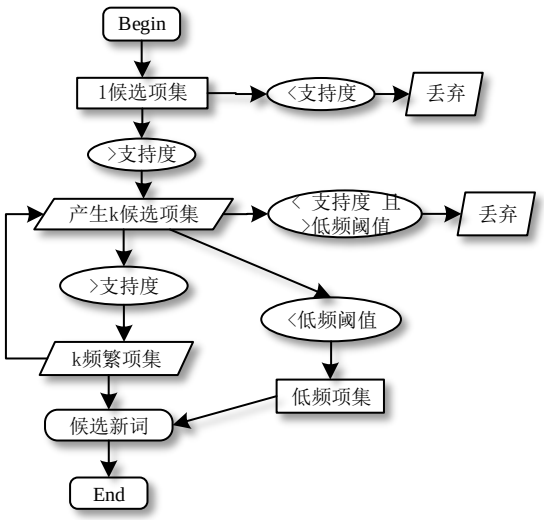

图 2 改进的类 Apriori 算法流程图

算法 1 描述了改进的类 Apriori 算法，图 2 展示了该算法的流程图。假设原始语料 *D* 由 *n* 个句子组成，*s* 表示字符串，*c* 表示字符，*f* 表示频率。

产生 1 频繁项集的过程可分为统计频率，过滤两个步骤。1 候选项集是语料中出现的每个字。统计原始语料中每个字出现的频率，把频率大于支持度的字加入 1 频繁项集。产生 2 频繁项集的过程可分为组合，统计频率，过滤三个步骤。把 1 频繁项集中的字两两组合后的字符串加入 2 候选项集，在原始语料中统计 2 候选项集的每个字符串出现的频率，把频率大于支持度的字符串加入 2 频繁项集，把频率小于低频阈值的字符串加入低频项集。产生 3 频繁项集到 K 频繁项集的过程可分为组合，统计频率，过滤三个步骤。把 k-1 频繁项集的两两字符串经过特殊的方式组合后产生的字符串加入 k 候选项集，在原始语料中统计 k 候选项集的每个字符串出现的频率，把频率大于支持度的字符串加入 k 频繁项集，把频率小于低频阈值的字符串加入低频项集。最后把频繁项集$L_2 \cdots L_K$和低频项集 M 取并集成为候选新词的集合。

改进的类 Aprior 算法相比传统的 Apriori 算法的区别有两点。第一，加入了低频阈值，使之能够挖掘古文中的低频新词。因为中国古代文学中的大部分词汇只在整个语料库中出现过一次或两次，但是频繁项集的生成是基于支持度（频率）的，因此传统的 Apriori 算法无法挖掘出古文中的低频新词。第二，对产生 3 频繁项集到 k 频繁项集过程的组合步骤提出改进。传统 Apriori 算法的组合步骤虽然也能挖掘出频繁候选字符串，但会产生大量的噪声词，而且没有考虑词语里字符的顺序关系。

算法 2. 改进的类 Apriori 的并行化算法

输入：原始语料D = $\{s_1, s_2, \cdots, s_n\}$

输出：候选新词的集合

① 将原始语料文件以 RDD 变量的形式读入内存，假设该 RDD 变量为 *input_rdd*。

//产生 1 频繁项集

② 对 *input_rdd* 的每一行，把出现的每一个字符存到数组中，并将每个字符映射到频率(也就是 1)。最后用 reduce 操作把相同字符的频率相加，得到每个元素是(字符，频率)的 RDD 变量f_1。

③ 对RDD变量f_1做filter操作，将频率大于支持度的元素保留，得到新的RDD变量L_1，即 1 频繁项集。

④ FOR k=2 TO K DO

//产生 k 频繁项集

⑤ IF k==2 THEN

⑥ 对L_1和L_1做笛卡尔积（cartesian 操作），产生 2 候选项集C_2(C_2也是 RDD 变量)。

⑦ ELSE

⑧ 用 map 操作把L_{k-1}的每个元素$\mathrm{P}=p_1p_2\cdots p_{k-1}$映射到$(p_1p_2\cdots p_{k-2},P)$，得到 RDD 变量 *pre*。用 map 操作把L_{k-1}的每个元素$\mathrm{Q}=q_1q_2\cdots q_{k-1}$映射到$(q_2q_3\cdots q_{k-1},Q)$，得到 RDD 变量 *post*。然后把 *pre* 和 *post* 做 join 操作，也就是把前缀和后缀相同的两个字符串关联在一起，然后把这两个字符串拼接在一起，得到候选词集C_k。

⑨ 对 *input_rdd* 的每一行，用长度为 k 的窗口滑动，将产生的所有长度为 k 的字符串存入数组，并将每个字符串映射到频率(也就是 1)。最后用 reduce 操作把相同字符串的频率相加，得到每个元素为(字符串，频率)的 RDD 变量 *scan_trans*。把C_k和 *scan_trans* 做 join 操作，得到每个元素为(字符串，频率)的 RDD 变量f_k，每个元素为在原始语料中出现过的候选词和对应的出现频率。

⑩ 对 RDD 变量f_k做 filter 操作，将频率大于支持度的元素保留，得到新的 RDD 变量L_k，即 k 频繁项集。再重新对 RDD 变量f_k做 filter 操作，这次将频率小于低频阈值的元素保留，将得到的 RDD 变量与低频项集 M 合并。

⑪ Return $L_2 \cup L_3 \cup \cdots \cup L_K \cup M$

本文基于 Spark 平台为改进的类 Apriori 算法实现了并行化，详见算法 2。

3. 4 Bi-LSTM-CRF 切分概率模型

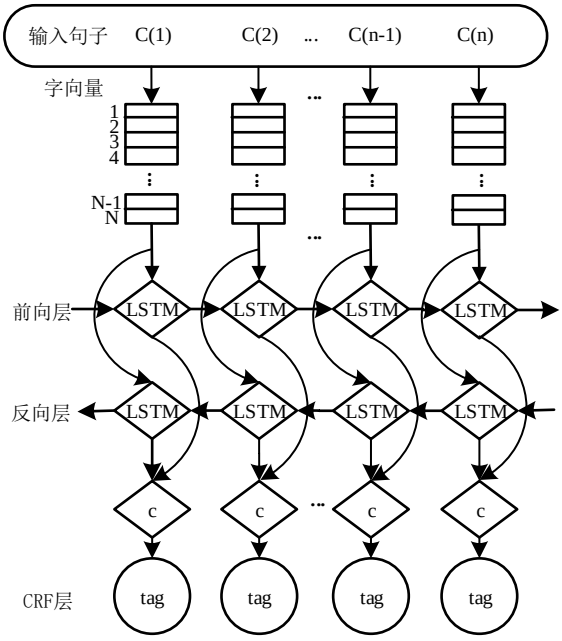

图 3 Bi-LSTM-CRF 切分概率模型结构图

图 3 展示了 Bi-LSTM-CRF 切分概率模型的结构图，接下来的几小节将分别解释该神经网络结构的每一层。

3.4.1 Embedding 层

在大规模古文语料上使用 word2vec[2]训练字向量。假设古文语料中的不同字构成的字典为 C，则字典大小为$|C|$，字向量的维度为 d。那么可以得到字向量矩阵$\mathrm{M} \in R^{d\times|c|}$。在 Embedding 层，把输入句子的每个字映射到它们对应的字向量。

3.4.2 Bi-LSTM 层

循环神经网络(Recurrent Neural Network, RNN)是庞大的神经网络家族中的一员，主要用于编码序列数据。它以一个向量序列$(x_1, x_2, \cdots, x_n)$作为输入，返回另一个序列$(h_1, h_2, \cdots, h_n)$来表示输入中每个步骤中有关序列的一些信息。尽管在理论上，RNN 能够学习到长距离的依赖信息，但是在实际应用中，却因为梯度爆炸或梯度消失的问题难以学习到长距离信息。

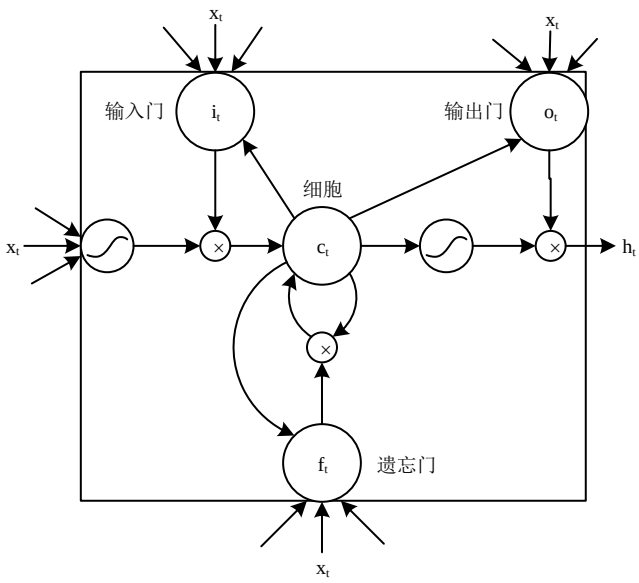

图 4 LSTM 单元结构

因此，长短时记忆网络(Long Short-term Memory Network, LSTM)通过添加了一个存储单元来克服这个问题，以此能够捕捉到长距离的依赖信息。LSTM 单元结构如图 4 所示。LSTM 单元拥有三个特殊的“门”结构，分别是输入门，遗忘门和输出门。“遗忘门”会根据当前的输入x_t、上一时刻状态c_{t-1}、上一时刻输出h_{t-1}共同决定哪一部分记忆需要被遗忘。“输入门”会根据x_t、c_{t-1}和h_{t-1}决定哪些部分将会被记忆。“输出门”会根据最新的状态c_t、上一时刻的输出h_{t-1}和当前的输入x_t来决定该时刻的输出。具体通过以下表达式来说明：

$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad \text{式(3-1)}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \quad \text{式(3-2)}$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad \text{式(3-3)}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad \text{式(3-4)}$$

$$h_t = o_t \tanh(c_t) \quad \text{式(3-5)}$$

其中σ是 *sigmoid* 函数；$\mathrm{i, f, o, c}$分别代表输入门、遗忘门、输出门和记忆细胞的启动向量。从权值矩阵的下标可以看出每一个权值矩阵的具体含义，例如W_{hi}代表的就是隐藏输入门的权值矩阵。b_i, b_f, b_c, b_o分别代表对应门的偏置向量。

但是，从前往后的 LSTM 层（前向 LSTM 层）只能编码每个字的上文信息。若要表达出每个字的上下文信息，还需要一个从后往前的 LSTM 层（反向 LSTM 层）编码每个字的下文信息，然后把这两层结合起来。如公式所示。

$$\overrightarrow{h_t} = \overrightarrow{LSTM}\left(w_t, \overrightarrow{h_{t-1}}\right) \quad \text{式(3-6)}$$

[2] https://code.google.com/archive/p/word2vec/

$$\overleftarrow{h_t} = \overleftarrow{LSTM}\left(w_t, \overleftarrow{h_{t+1}}\right) \quad \text{式(3-7)}$$

$$h_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right] \quad \text{式(3-8)}$$

这样，通过双向 LSTM 神经网络，就可以表示出一句话中每个字所有的上下文信息，为后续的标注任务奠定了基础。

3.4.3 全连接层

表 1 BMES 标注表

标签	含义
B	词的开头字
M	词的中间字
E	词的结尾字
S	独立成词

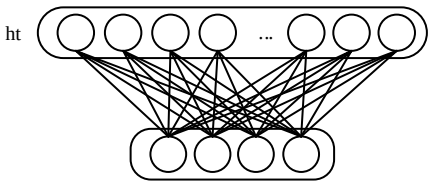

图 5 全连接层时刻 t 的网络结构

从 Bi-LSTM 层输出的是每个时刻 t（每个字）的隐状态向量h_t，接下来使每个时刻的输出映射到4个神经网络单元，分别代表该字标注为B,M,E,S 的分数，映射采用全连接的方式，如图 5 所示。字的标签 B,M,E,S 的含义见表 1。

3.4.4 CRF 层

在全连接层之后可直接加一个 *softmax* 层，对标注为 B,M,E,S 的分数做归一化，使之转化为概率，即每个时刻这四个神经网络单元的输出相加为 1。但是这种方法对每个字的标签独立地进行预测，无法捕捉到输出标签之间的依赖关系（例如：标签“M”后面不能接标签“S”），而条件随机场(Conditional Random Field, CRF)可以有效地解决这个问题。

对于一个输入句子（假设$x_1, x_2, \cdots, x_n$都是中文字符）

$$\mathrm{X} = (x_1, x_2, \cdots, x_n) \quad \text{式(3-9)}$$

假设预测的输出标签序列为

$$\mathrm{Y} = (y_1, y_2, \cdots, y_n) \quad \text{式(3-10)}$$

定义它的得分为：

$$\mathrm{s}(X, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \quad \text{式(3-11)}$$

其中 *A* 是转移分数矩阵，$A_{i,j}$表示从标签 *i* 转移到标签 *j* 的分数。*P* 是双向 LSTM 层输出的得分矩阵，所以 *P* 的大小是n × k，k 是不同的标签数，$P_{i,j}$表示句子的第 *i* 个字标注为标签 *j* 的分数。

在 CRF 层后，通过一个 *softmax* 层，得到给定输入句子 X 的条件下所有可能的标签序列的概率。

$$\mathrm{p}(y|X) = \frac{e^{s(X,y)}}{\sum_{y' \in Y_x} e^{s(X,y')}} \quad \text{式(3-12)}$$

然后在模型训练的过程中，极大化正确标签序列的 log 似然。

$$\log\left(p(y|X)\right) = s(X,y) - log\left(\sum_{y' \in Y_x} e^{s(X,y')}\right) = s(X,y) - \underset{y' \in Y_x}{logadd} s(X,y') \quad \text{式(3-13)}$$

在测试（解码）的过程中，将得分最高的标签序列作为预测输出序列：

$$y^* = \underset{y' \in Y_x}{argmax}\, s(X, y') \quad \text{式(3-14)}$$

3. 5 过滤规则

对于由改进的类 Apriori 算法产生的每个候选新词，它在测试集文档中可能出现一次或多次，并且在每个出现位置都存在一个输入上下文。测试集文档经过训练好的 Bi-LSTM-CRF 模型后，会得到每一个位置的切分结果。

接下来，给出过滤规则的定义：

定义 3. 过滤规则

对于一个候选新词，如果存在某个输入上下文窗口，它的左边界切分概率和右边界切分概率同时大于 0.5，并且它的内部切分概率全都小于 0.5，则将其划分为新词。

4 实验

4.1 数据集介绍

在本文中，主要用到两个很有代表性的古文数据集：宋词和史记。这两个数据集来源于谢韬等人[11]的实验语料，其中，分别随机抽取 3 万行进行了分词标注，并标注出有具体语义的新词。数据集的具体描述见表 2。

表 2 数据集统计信息

数据集	原始规模	分词标注	新词标注
宋词	139W 字	30000 行	14891 个
史记	525W 字	32000 行	12132 个

另外，本文将开源中文分词工具 jieba 分词器的词典作为算法里的标准词典，jieba 词典共包含 584429 个已知词。

4. 2 实验结果

4.2.1 有效性实验

4.2.1.1 分布式的类 Apriori 算法

改进的类 Apriori 算法采用和谢韬等人[11]工作的相同实验设置，将支持度设置为 5，低频阈值设置为 2，对于宋词语料，设置频繁项集的最大长度为 5，对于史记语料，频繁项集的最大长度设为 10。在测试集上运行改进的类 Apriori 算法，分别在宋词语料和史记语料中得到了 13905 和 12265 个候选词。

表 3 计算节点配置信息

项目	配置说明
CPU	12 Core Intel Genuine 2.10GHz
Memory	48GB
Disk	1T
OS	Ubuntu 14.04
JVM Version	Java 1.8.0
Hadoop Version	Hadoop 2.7.5
Spark Version	Spark 2.1.2

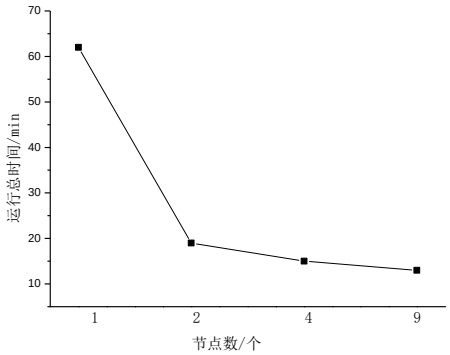

图 6 改进的类 Apriori 算法随节点数目变化运行时间图

本文采用的并行化实验集群是由 1 个主控节点和 9 个计算节点组成，计算节点的配置信息参见表 3。实验结果如图 6 所示，并行化后的算法相比于串行的算法，效率提升显著。而且节点数越多，算法耗费的时间越少。

4.2.1.2 Bi-LSTM-CRF 切分概率模型

切分概率模型用来预测一个句子的每个位置是否应该被切分。假设一个句子表示为 $s = c_1 c_2 \cdots c_n$（$c_1, c_2, \cdots, c_n$表示句子中对应位置的字）。经过切分概率模型后，则产生由 0 和 1 组成的长度为 n-1 的序列，1 表示切分，0 表示不切分。该序列是预测的切分序列。带有分词标注的句子经过处理可以得到实际的切分序列。

在实验中，用准确率、召回率和 F1 值来评价切分概率模型的好坏。

$$\text{准确率} = \frac{\text{有效切分的次数}}{\text{预测切分的次数}} \times 100\% \quad \text{式(4-1)}$$

$$\text{召回率} = \frac{\text{有效切分的次数}}{\text{实际切分的次数}} \times 100\% \quad \text{式(4-2)}$$

$$\text{F1 值} = \frac{2 \times \text{准确率} \times \text{召回率}}{\text{准确率} + \text{召回率}} \quad \text{式(4-3)}$$

本文将经过分词标注的数据集划分为训练集和测试集，划分比例为 8:2。

神经网络模型中超参数的设置对实验结果有较大影响，本文选择的超参数的值详见表 4。表中所列超参数的值是凭经验选择的。其中，Embedding 层和 Bi-LSTM 层的 dropout 比率都设置为 0.5。设置 dropout 主要是为了防止模型的过拟合。

表 4 超参数设置

最大轮数	Epoch = 20
批大小	Batch_size = 64
学习速率	α = 0.002
正则化系数	λ = 0.001
Dropout 比率	P = 0.5

表 5 Bi-LSTM-CRF 切分概率模型实验结果

数据集		宋词			史记		
隐藏层单元个数	字向量维度	准确率(%)	召回率(%)	F1 值(%)	准确率(%)	召回率(%)	F1 值(%)
150	50	96.41	95.11	95.11	96.85	95.29	**96.07**
128	100	93.52	94.80	**96.12**	96.70	94.93	95.81
150	100	96.04	93.04	94.51	97.08	94.86	95.96
200	100	95.85	93.74	94.78	96.67	95.35	96.01
150	150	95.75	93.91	94.82	96.79	95.09	95.03
150	200	95.32	94.26	94.79	96.77	94.93	95.84

关于隐藏层单元个数和字向量维度这两个超参数的选择过程如下。

测试基于不同字向量维度和 LSTM 隐藏层单元个数的模型效果。如表 5 所示，对于宋词语料，当字向量的维度是 100，LSTM 隐藏层单元个数是 128 的时候，Bi-LSTM-CRF 有最高的性能。而对于史记语料，当把字向量维度设置成 50，LSTM 隐藏层单元个数设置为 150 时，模型能够有最好的效果。所以本文将这两组设置和表 4 的设置合起来作为后续实验的超参数设置。相比于谢韬等人[11]的切分概率模型，F1 值分别有 7.20%和 3.07%的提升。（注：谢韬等人[11]的切分概率模型是用来判断一个四字输入的中间是否应该被切分。它的网络结构是字向量层+Bi-LSTM 层+1 输出的全连接层，再接 sigmoid 函数把输出值映射到[0,1]。最终的输出表示四字输入的中间的切分概率，大于 0.5 表示切分，小于 0.5 表示不切分。）

4.2.1.3 过滤规则

假设改进的类 Apriori 算法产生的某个候选新词是字符序列C_tC_{t+1}，找到它在测试集文档中的每一个出现位置，然后获得它的左邻接字符和右邻接字符，所以可以得到一个上下文窗口$C_{t-2}C_{t-1}C_tC_{t+1}C_{t+2}C_{t+3}$，如果上下文中不存在邻接字符，则用一个默认字符“padding”替代。进而能够得到三个小窗口：$C_{t-2}C_{t-1}C_tC_{t+1}$，$C_{t-1}C_tC_{t+1}C_{t+2}$，$C_tC_{t+1}C_{t+2}C_{t+3}$。通过 Bi-LSTM-CRF 切分概率模型可以得到测试集文档中每个句子任意两个连续字之间的切分概率，从中找出这三个小窗口中间位置的切分概率，就得到了候选词的内部切分概率和边界切分概率。然后利用与内部切分概率和边界切分概率有关的过滤规则来判断候选新词是真正的新词还是噪声词。这样，利用过滤规则从候选词集中筛选出真正的新词，即 AP-LSTM-CRF 模型预测的新词。语料中标注的新词即是实际的新词。

在实验中，用准确率、召回率和 F1 值来对新词发现结果进行评价。

$$\text{准确率} = \frac{\text{有效的新词数}}{\text{模型预测的新词数}} \times 100\% \quad \text{式(4-4)}$$

$$\text{召回率} = \frac{\text{有效的新词数}}{\text{语料中的实际新词数}} \times 100\% \quad \text{式(4-5)}$$

$$\text{F1 值} = \frac{2\times\text{准确率}\times\text{召回率}}{\text{准确率}+\text{召回率}} \quad \text{式(4-6)}$$

使用谢韬等人[11]的过滤规则，新词发现的效果如表 6 所示（AP-LSTM 模型是谢韬等人[18]提出的古文新词发现算法，它的过滤规则是，对于一个候选新词，如果存在某个输入上下文窗口，它的左边界切分概率和右边界切分概率同时大于 0.5，则将其划分为新词）。

表 6 使用旧的过滤规则的新词发现实验结果

数据集	宋词		史记	
模型	AP-LSTM	AP-LSTM-CRF	AP-LSTM	AP-LSTM-CRF
语料中实际的新词数	3757	3757	2270	2270
有效新词数	3549	3583	1957	1975
预测的新词数	5911	3757	4987	4881
准确率(%)	60.04	66.69	39.24	40.46
召回率(%)	94.46	95.37	86.21	87.00
F1 值(%)	73.42	78.49	53.93	55.24

使用本文提出的新的过滤规则的实验结果如表 7 所示。

表 7 使用新的过滤规则的新词发现实验结果

数据集	宋词		史记	
模型	AP-LSTM	AP-LSTM-CRF	AP-LSTM	AP-LSTM-CRF
语料中实际的新词数	3757	3757	2270	2270
有效新词数	2907	3436	1728	1832
预测的新词数	3197	3906	2129	2246
准确率(%)	90.93	87.97	81.16	81.56
召回率(%)	77.38	91.46	76.12	80.70
F1 值(%)	83.61	89.68	78.56	81.13

分别观察表 6 和表 7，可以看出，无论是使用谢韬等人[11]提出的过滤规则，还是使用新的过滤规则，AP-LSTM-CRF 算法的性能都比 AP-LSTM 好，证明了本文提出的 Bi-LSTM-CRF 切分概率模型的有效性。

通过对比表 6 和表 7 的实验结果，可以发现无论是 AP-LSTM 算法，还是 AP-LSTM-CRF 算法，使用新的过滤规则的实验结果都比使用谢韬等人[11]提出的过滤规则的实验结果要好，由此证明了本文提出的新的过滤规则的有效性。

4.2.2 AP-LSTM-CRF 算法和其它算法的对比实验

本文对比了古文新词发现算法 AP-LSTM-CRF 和当今主流的几种开源中文分词工具(Ansj[3],中科院 ICTCLAS[4],斯坦福中文分词工具[5])以及 Deng 等人[2]提出的 TopWords 模型的效果。

关于 AP-LSTM-CRF 模型，改进的类 Apriori 算法部分采用谢韬等人[11]的实验设置，Bi-LSTM-CRF 切分概率模型采用 4.2.1.2 节最佳的超参数设置，最后利用新的过滤规则。

关于几种中文分词工具，首先用分词器切分语料，然后把得到的词用 jieba 词典过滤，得到预测的新词。

关于 TopWords 模型，采用 Deng 等人[2]的实验设置。

经过实验，关于发现的有效新词个数的对比如图 7 所示，关于新词发现的 F1 值的对比如图 8 所示。

[3] https://github.com/NLPchina/ansj_seg
[4] http://ictclas.nlpir.org/
[5] https://nlp.stanford.edu/software/segmenter.shtml

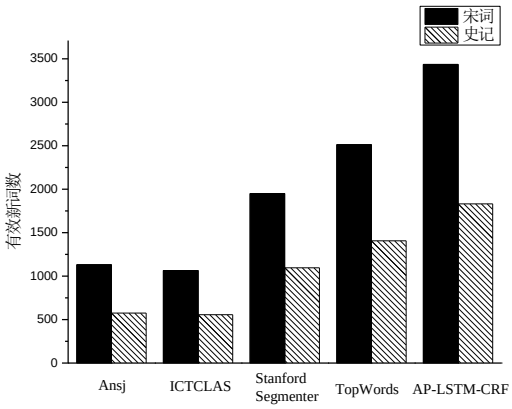

图 7 AP-LSTM-CRF 算法和其他算法发现的有效新词数对比

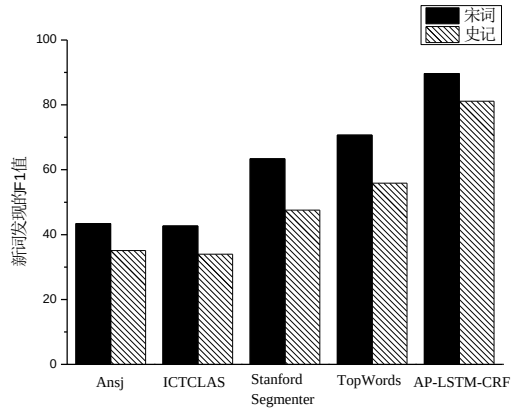

图 8 AP-LSTM-CRF 算法和其他算法发现新词的 F1 值对比

可以看出，AP-LSTM-CRF 算法的性能明显优于现有的主流分词器，这证明了现代文和古文在语法和构词规则上确实存在很大的差异，针对现代文的算法无法很好地运用于古文这一特殊领域。而 AP-LSTM-CRF 模型比 TopWords 模型的效果要好，证明了低频新词也是古文中新词的重要组成部分，通过找到更多的低频新词，可以减少切分歧义，使相同的词不再会有多种切分结果。

4.2.3 差错分析

有两种情况的识别错误。第一种情况是算法并未将实际的新词识别出来。如“江渚”，它在原文中出现过 1 次，“乘/兴/离/江渚”。经过 Bi-LSTM-CRF 切分概率模型，得到的切分序列为 011，而实际的切分序列应该是 101，根据过滤规则，把该词判定为噪声词，而实际上该词是新词。再如“山耸”，在原文中出现过 2 次，分别是“鳌/山耸”，“重叠/暮山/耸翠”。经过 Bi-LSTM-CRF 切分概率模型，得到的切分序列分别为 011 和 010，而实际的切分序列应该为 101 和 010，根据过滤规则，把该词判定为噪声词，而实际上该词是新词。在这两个例子中，若切分概率模型得到的切分序列是正确的，那么根据过滤规则可以得到正确的结果，所以问题出在切分概率模型。

第二种情况是算法将噪声词误判成了新词。如“渐遏”，它在原文中出现过 1 次，“渐/遏/遥天”。经过 Bi-LSTM-CRF 切分概率模型，得到的切分序列为 101，而实际的切分序列应该是 111，根据过滤规则，把该词判定为新词，而实际上该词为噪声词。再如“知我”，在原文中出现过 2 次，“争/得/知/我”，“争知/我”。经过 Bi-LSTM-CRF 切分概率模型，得到的切分序列为 101 和 011，而实际的切分序列应该为 111 和 011，根据过滤规则，把该词判

定为新词，而实际上该词为噪声词。在这两个例子中，问题同样在于切分概率模型，所以在后续的工作中切分概率模型有待进一步完善。

但是由这几个例子可以看出过滤规则的制定是合理的，为了包容切分概率模型可能产生的差错，过滤规则要求只要有一个上下文完全满足两边切分中间不切分的条件就判断成新词。事实证明，如果切分概率模型预测正确，使用这样的过滤规则是不可能判断失误的。

4.2.4 古文新词发现结果展示

表 8 分别列举了 AP-LSTM-CRF 算法在宋词和史记中发现的 20 个典型的新词。

表 8 古文中发现的典型新词展示

宋词	流霞	寒灯	洞户	槛菊	沙汀
	宝辇	残蝉	万灵	嫩香	苹花
	红茵	九仪	孤帏	岁华	画檐
	庭宇	万斛	井梧	幽闺	星闱
史记	亓氏	孫覺	浮漏	銀帛	是歲
	畿内	龜茲	冢宰	交州	暴骸
	賢妃	郊宫	升黜	臺諫	熟戶
	夏兵	岢嵐	徹樂	伏誅	熒惑

基于在语料中已经发现的新词，本文把这些新发现的词添加到分词器的词典中，并比较其与原分词器的性能，结果如下。

表 9 加入新词后分词器的性能提升效果

数据集	宋词		史记	
分词器	原始分词器	加入新词	原始分词器	加入新词
准确率(%)	76.63	94.15	76.42	94.53
召回率(%)	94.58	90.87	95.82	93.03
F1 值(%)	84.66	92.48	85.03	93.77

从表 9 可以看出，当向分词器里添加了新词之后，分词器的准确性得到了极大的提升。

5 结束语

本文提出了一种基于古文语料的新词发现算法 AP-LSTM-CRF。改进的类 Apriori 算法可以有效地挖掘低频新词，通过并行化该算法，极大地提升产生候选词集的效率。Bi-LSTM-CRF 切分概率模型可以更加准确地判断两个字之间的切分概率。基于候选词上下文的切分概率序列，本文提出的过滤规则可以更加有效地过滤掉噪声词，从而有效地从古文语料中挖掘新词。

将来，我们将尝试采用无监督或半监督的算法对古文的新词进行挖掘。因为如果采用有监督的算法，就需要研究者自己人工标注大量训练数据，这种代价是十分大的。此外，在大量的古文数据背景下，句子语法十分繁多，这给数据的标注和模型的训练带来了很大的挑战。因此，无监督地在古文语料中发现新词具有十分重大的现实意义。

参考文献

[1] Huang Chang-Ning, Zhao Hai. Chinese word segmentation: a decade review[J]. Journal of Chinese Information Processing, 2007, 21(3): 8-19. (in Chinese)
(黄昌宁, 赵海. 中文分词十年回顾[J]. 中文信息学报, 2007, 21(3): 8-19.)

[2] Ke Deng, Peter K. Bol, Kate J. Li, et al. On the unsupervised analysis of domain-specific Chinese texts[C]. //Proceedings of the National Academy of Sciences. 2016: pp. 6154-6159.

[3] Chen Ao, Sun Mao-Song. Domain-specific new words detection in Chinese[C]. //Proceedings of the 6th Joint Conference on Lexical and Computational Semantics. 2017: pp. 44-53.

[4] Huo Shuai, Zhang Min, Liu Yi-Qun, et al. New words discovery in microblog content[J]. Pattern Recognition and Artificial Intelligence, 2014, 27(2): 141-145. (in Chinese)
(霍帅, 张敏, 刘奕群, 等. 基于微博内容的新词发现方法[J]. 模式识别与人工智能, 2014, 27(2): 141-145.)

[5] Zhou Shuang-Shuang, Xu Jin-An, Chen Yu-Feng, et al. New words detection method for microblog text based on integrating of rules and statistics[J]. Journal of Computer Applications, 2017, 37(4): 1044-1050. (in Chinese)
(周霜霜, 徐金安, 陈钰枫, 等. 融合规则与统计的微博新词发现方法[J]. 计算机应用, 2017, 37(4): 1044-1050.)

[6] Lei Yi-Ming, Liu Yong, Huo Hua. New word discovery based on microblog corpus for network language[J]. Computer Engineering and Design, 2017, 38(3): 789-794. (in Chinese)
(雷一鸣, 刘勇, 霍华. 面向网络语言基于微博语料的新词发现方法[J]. 计算机工程与设计, 2017, 38(3): 789-794.)

[7] Du Li-Ping, Li Xiao-Ge, Yu Gen. New word detection based on an improved PMI algorithm for enhancing segmentation system[J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2016, 52(1): 35-40. (in Chinese)
(杜丽萍, 李晓戈, 于根. 基于互信息改进算法的新词发现对中文分词系统改进[J]. 北京大学学报(自然科学版), 2016, 52(1): 35-40.)

[8] Chen Fei, Liu Yi-Qun, Wei Chao, et al. Open domain new word detection using condition random field method[J]. Journal of Software, 2013, 24(5): 1051-1060. (in Chinese)
(陈飞, 刘奕群, 魏超等. 基于条件随机场方法的开放领域新词发现[J]. 软件学报, 2013, 24(5): 1051-1060.)

[9] Yang Yang, Liu Long-Fei, Wei Xian-Hui. New methods for extracting emotional words based on distributed representations of words[J]. Journal of Shandong University (Natural Science), 2014, 49(11): 51-58. (in Chinese)
(杨阳, 刘龙飞, 魏现辉. 基于词向量的情感新词发现方法[J]. 山东大学学报(理学版), 2014, 49(11): 51-58.)

[10] Wan Qi, Yu Zhong-Hua, Chen Li, et al. Improving emotion expression extraction in Chinese microblogs via new words detection[J]. Journal of University of Science and Technology of China, 2017, 47(1): 63-69. (in Chinese)
(万琪, 于中华, 陈黎, 等. 利用新词探测提高中文微博的情感表达抽取[J]. 中国科学技术大学学报, 2017, 47(1): 63-69.)

[11] Xie Tao, Wu Bin, Wang Bai. New word detection in ancient Chinese literature[C]. //Proceedings of the Asia-Pacific Web and Web-Age Information Management Joint Conference on Web and Big Data. 2017: pp. 260-275.

作者联系方式：刘昱彤　北京市海淀区西土城路 10 号北京邮电大学教三 616 房间
18600029812　443072618@qq.com