doi:10.3772/j.issn.2095-915x.2017.03.003

面向科技语料的短语结构句法分析器

哈尔滨工业大学机器智能与翻译研究室 哈尔滨 150001

王亚楠 马春鹏 曹海龙 赵铁军

摘 要 本文介绍了一个由哈尔滨工业大学设计和开发的面向科技语料的短语结构句法分析器。与传统的短语结构句法分析器不同，本句法分析器不需要对输入语料进行预处理。给定未经预处理的语料，本句法分析器可以联合地进行分词、词性标注以及短语结构的句法分析。这可以看成是多任务学习的一个实例。此外，针对科技语料的特点，本句法分析器对所使用的特征模板进行了优化，同时构建了面向科技语料的单词内部结构树库。实验结果表明，我们的句法分析器在通用领域的测试集以及科技领域的测试集上均取得了较好的效果。

关键词： 短语结构句法分析，科技语料，多任务学习

中图分类号： G35，TP39

A Constituent Parser for Science and Technology Corpus

Machine Intelligence and Translation Laboratory, Harbin Institute of Technology, Harbin 150001, China

WANG YaNan MA ChunPeng CAO HaiLong ZHAO TieJun

Abstract In this paper, we proposed a constituent parser for science and technology corpus, which was designed and developed by Harbin Institute of Technology. Compared with traditional constituent parsers, the parser of this study does not need to pre-processed corpus. Given a raw text as the input, this parser can do the tasks of word segmentation, POS-tagging and constituent parsing simultaneously. This can be regarded as an instance of multi-task learning. Furthermore, based on the characteristics of science and technology corpus, we optimized the feature templates used in our parser, and constructed a new tree-bank of the inner

基金项目： 本文受国家自然科学基金项目（91520204，61572154），863项目（2015AA015405），和微软亚洲研究院合作研究计划的资助。

作者简介： 王亚楠（1993–），硕士研究生，研究方向：句法分析，机器翻译，Email：ynwang@mtlab.hit.edu.cn；马春鹏（1992–），博士研究生，研究方向：句法分析、语义表示与推理、机器翻译；曹海龙（1976–），博士，讲师，研究方向：句法分析、机器翻译；赵铁军（1962–），博士，教授，博士生导师，研究方向：机器翻译、机器学习。

structures of the words in the science and technology corpora. The results of the experiments indicated that our parser performed well both on the corpus of general domain and on the corpus of science/technology domain.

Keywords: Constituent parsing, science and technology corpus, multi-task learning

1 引言

句法分析（parsing）是自然语言处理领域的一个重要问题。对一个句子进行句法分析，就是分析出这个句子中各个单词（或汉字）之间的句法关系。句法分析是语块分析思想的一个直接实现，它通过识别出高层次的结构单元来简化句子的描述[1]。

传统的句法分析器[2]在进行句法分析之前通常需要使用其他的工具对语料进行分词、词性标注的预处理，因此句法分析的精度会受到分词和词性标注结果的影响。与此相对，本文的句法分析器通过联合地进行分词、词性标注和句法分析，以多任务学习的方式，在一个统一的框架下训练了一个统一的模型。通过这个模型，本句法分析器可以同时完成分词、词性标注和句法分析三个任务，降低了误差传播带来的影响。

目前通用的中文句法分析器[2-4]通常在中文树库（CTB）[5]上进行训练。此语料库的文本主要来自新闻文本，因此在此语料库上训练得到的句法分析器通常在新闻文本的句法分析上表现较好，而在其他领域文本（例如科技语料）的句法分析上表现较差。针对这一问题，本文构建了一个人工标记的面向科技语料的内部结构树库。通过使用这一树库，本文的句法分析器能够同时在科技文献语料以及新闻语料上取得较好的句法分析效果。

本文将首先介绍目前关于句法分析的研究现状（第2节），然后介绍本项目①实现的句法分析器的基本原理（第3节），以及针对科技文献语料所构建的单词内部结构树库（第4节）。最后，本文将给出句法分析器在通用的新闻语料以及科技文献语料上的句法分析结果（第5节）。

2 研究现状

经典的短语结构句法分析方法大致可以分为两类：基于图的句法分析方法、基于转移的句法分析方法。近年来，随着深度学习技术的发展，基于神经网络的句法分析方法正逐渐受到研究者的关注。

2.1 基于图的句法分析

基于图的短语结构句法分析的基础是概率上下文无关文法（PCFG）[6,7]，它被定义为一个五元组 $G=(N, \Sigma, P', S, P)$，其中 N 是非终结符的有穷集合，Σ 是终结符的有穷集合，P' 是形如 $\alpha \rightarrow \beta$ 的产生式的集合，其中 $\alpha \in N$，$\beta \in \{N \cup E\}^*$，S 是

①即上文提到的863项目（2015AA015405），下文同。本项目的任务是面向科技语料的中日机器翻译。使用本文提出的句法分析器，机器翻译的效果得到了大幅度的提升。

doi:10.3772/j.issn.2095-915x.2017.03.003

起始符号，且$S \in N$。P 是产生式概率的集合，其中的每一个元素 $P(\alpha \to \beta|\alpha)$ 均与集合 P' 中的元素 $\alpha \to \beta$ 相对应。我们规定，产生式只能是下面两种形式之一：

- $X \to x$，其中为非终结符，x 为终结符。
- $X \to Y_1 \cdots Y_n$，其中 X 和 $Y_1 \cdots Y_n$ 都是非终结符。

于是，语法分析的过程即可转化为下面的最优化问题：

$$T_{best}(s) = \underset{T \in \tau(s)}{\operatorname{argmax}} P(T)$$

其中，s 为待分析的句子，$\tau(s)$ 是句子 s 可能的句法树的集合，$T_{best}(s)$ 是最优的句法树，$P(T)$ 可以按照如下的公式计算：

$$P(T) = \prod_{i=1...n} P(\alpha_i \to \beta_i | \alpha_i)$$

通过上述分解，即可计算出概率最高的句法树，从而完成句法分析。

2.2 基于转移的句法分析

基于转移的短语结构句法分析方法将句法分析的过程转化为一系列的状态转移的过程。通过一系列的移进 - 归约动作，一棵完整的短语结构句法树被生成。早期的基于转移的短语结构句法分析方法在思想上与编译原理的语法分析技术类似[8]，然而真正得到广泛使用的基于转移的短语结构句法分析方法的思想是将转移动作的决策转化为多类分类问题[9]，从而在英文以及包括中文[10]在内的多种语言上取得了较好的句法分析结果。

与基于图的句法分析方法相比，基于转移的句法分析方法具有以下优点：（1）句法分析的时间复杂度与输入句子的长度之间具有线性关系，（2）可以方便地添加比较复杂的特征，从而使得句子的各种信息可以较充分地被利用，（3）句法分析的性能与基于图的句法分析方法基本相当。

2.3 基于神经网络的句法分析

近年来，随着深度学习技术的发展，基于神经网络的各种算法已经广泛地被应用于众多自然语言处理任务中。针对句法分析任务，被广泛使用的神经网络模型包括递归神经网络模型[11]，以及基于循环神经网络的序列生成模型[12]。

与基于图的句法分析方法以及基于转移的句法分析方法不同，基于神经网络的句法分析不需要进行人为的特征模板选择。同时，借助诸如 Theano[13]、Tensorflow[14] 等面向神经网络的函数库，基于神经网络的句法分析器的实现十分简单。然而，基于神经网络的句法分析器的一个缺点是，为了达到较好的性能，通常这些句法分析器需要大量的有标注或者无标注数据进行训练。

2.4 总结

根据上面的分析可以看到，不同的句法分析方法有着各自的优点和缺点。对于本项目来说，由于句法分析的结果要服务于后续的基于句法树的科技文本机器翻译任务，因此本项目的句法分析器对于句法分析的速度有着较高的要求。同时，由于本项目的句法分析器涉及不同领域文本之间的转移，因此句法分析所使用的特征模板应该能够较方便地进行修改。针对这些要求与特点，本文实现的句法分析器采用的是基于转移的句法分析方法。

3 汉字粒度的短语结构句法分析方法

本文实现的句法分析器的总体框架如图 1 所示。这个框架由文献 [15] 提出。

句法分析器的输入为未经处理的中文句子，输出为汉字粒度的句法树。句法分析通过一系列的状态转移完成，每个状态由一个栈和一个队列组成。栈中保存着已经生成的句法树片段，队列中保存着尚未处理的汉字。初始状态下，栈为空，队列中元素的个数与句子中汉字的个数相同。终止状态下，队列为空，栈中只有唯一的结点，这个结点对应着一棵完整的句法树。每个状态转移的动作从一个预先定义好的动作集合中选择。此外，本文的句法分析器采用了柱搜索、平均感知器以及提前更新等策略。实验表明，采用这些策略会使得句法分析的精度得到提升。

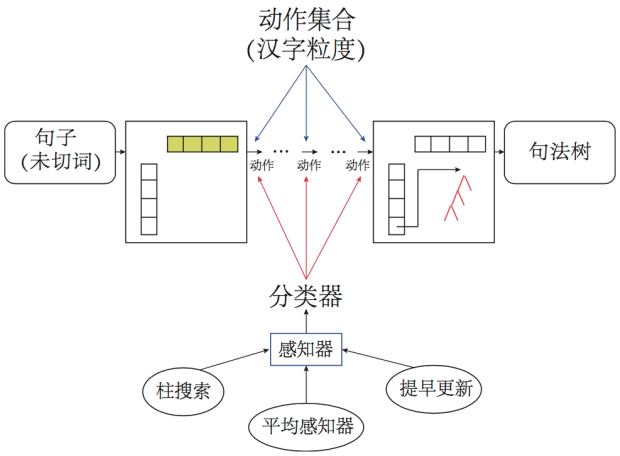

图1 句法分析器总体框架

3.1 动作集合

状态的转移通过一系列的动作完成，每个动作都是基于汉字粒度的。假设状态 ST 由栈 S 和队列 Q 组成，其中 $S=(\ldots, S_1, S_0)$ 包含了构建了一部分的句法树，$Q=(Q_0, Q_1,\ldots, Q_{n-j})=(c_j, c_{j+1},\ldots,c_n)$ 是尚未处理的输入汉字序列。我们形式化地定义动作集合如下：

• 移进 - 分裂（t）：把队列 Q 的队首的汉字弹出，在 S 中压入一个子单词结点 $\frac{S'}{c_j}$，并且令 $S'.t=t$。注意句法树 S_0 必须对应一个全单词结点或者一个短语结点，汉字 c_j 是下一个单词的汉字，参数 t 表示 S' 的词性。

• 移进 - 附着：把队列 Q 的队首的汉字弹出，在 S 中压入一个子单词结点。汉字 c_j 最终会与 S 顶端的所有子单词结点结合，组成一个单词，因此一定有 $S'.t=S_0.t$。

• 归约 - 子单词（d）：把 S_0 和 S_1 从 S 中弹出，在 S 中压入一个新的子单词结点 $\frac{S'}{S_1S_0}$。参数 d 表示 S' 的头方向，值可以是“左”、“右”或者“并列”。S_0 和 S_1 都必须是子单词结点，并且 $S'.t=S_0.t=S_1.t$。

• 归约 - 单词：把 S_0 从 S 中弹出，在 S 中压入一个全单词结点 $\frac{S'}{S_0}$。这种归约操作从 S_0 产生了一个全单词结点。S_0 必须是子单词结点。

• 归约 - 二元（d,l）：把 S_0 和 S_1 从 S 中弹出，在 S 中压入一个新的子单词结点 $\frac{S'}{S_1S_0}$。参

doi:10.3772/j.issn.2095-915x.2017.03.003

数 l 表示 S' 的句法标签，参数 d 表示 P' 的词汇头方向，值可以是“左”或“右”。S_0 和 S_1 都必须是全单词结点或者短语结点。

• 归约 - 一元（l）：把 S_0 从 S 中弹出，在 S 中压入一个全单词结点 $\frac{S'}{S_0}$。参数 l 表示 S' 的句法标签。

• 停顿：栈 S 和队列 Q 的状态不变。

• 终结：表示句法分析结束。

3.2 感知器算法与精度提升策略

影响句法分析精度的最主要部分是每个状态下转移动作的决策过程。这一决策过程由分类器执行，因此分类器是句法分析器的核心部分。本文的句法分析器使用感知器模型进行训练[16]。算法的伪代码如图 2 所示[17]。

为了提升句法分析的精度，在上述感知器算法的基础上，本文的句法分析器采用了如下的策略：柱搜索、提早更新、平均感知器。

Algorithm 1 感知器算法

```
1: procedure PERCEPTRON((x_i, y_i))
2:     w⃗ = 0
3:     for t = 1...T, i = 1...N do
4:         z_i = parse(x_i, w⃗)
5:     end for
6:     if z_i ≠ y_i then
7:         w⃗ = w⃗ + Φ(y_i) − Φ(z_i)
8:     end if
9: return w⃗
10: end procedure
```

图2 感知器算法伪代码

3.2.1 柱搜索

上述感知器算法是一种贪心搜索算法，每次只选择得分最高的状态与正确的状态进行比较，决定是否进行权值的更新。这会使得正确的搜索状态被淘汰掉，从而使得句法分析的精度下降。柱搜索策略在一定程度上缓解了这一现象的发生。柱搜索算法的伪代码如图 3 所示[18]。

Algorithm 2 柱搜索算法

```
1: procedure BEAM SEARCH(item = (S, Q), agenda, list)
2:     for item ∈ agenda do
3:         if item.score == agenda.bestScore ∧ item.isFinished then
4:             rval = item
5:             break
6:         end if
7:         next = []
8:         for move ∈ item.legalMoves do
9:             next.push(item.TakeAction(move))
10:        end for
11:        agenda = next.getBBest()
12:    end for
13: return rval
14: end procedure
```

图3 柱搜索算法伪代码

3.2.2 提早更新

提早更新策略[16]一方面能够减少训练时间，另一方面能够提升分类器的精度。

直观地讲，如果模型的权值与高精度分类器所使用的模型的权值比较接近，则在柱搜索的过程中，正确状态的得分应该始终处于最高的几位，即正确的状态应该始终在“柱”当中。然而一个错误的权值不会有这样的效果。当模型的权值与高精度分类器所使用的模型权值偏差较大时，在柱搜索的过程中，很可能错误状态的得分较高，而正确状态因为得分排名过低而被淘汰，一旦正确的状态在柱搜索的过程中

被淘汰，那么无论“柱”中的状态如何进行状态转移，均无法得到正确的解码结果。

“提早更新”策略正是基于上述思想而提出的。当正确的状态被从“柱”中淘汰后，解码过程停止，立即进行权值更新。这样，当模型的权值与高精度分类器所使用的模型的权值偏差较大时，解码过程进行很少的步骤即可终止，从而减少了训练的时间。另一方面，采用这一策略时，权值更新发生在错误产生之处，因此更新之后的权值能够更好地反映训练集的语言特征，为模型提供更有效的信息，从而提升了句法分析器的精度。

3.2.3 平均感知器

平均感知器策略 [19] 在一定程度上可以避免过拟合的发生。设迭代总轮数为 T，每轮迭代的索引为 t，其中 $0<t<T+1$，语料库中的句子总数为 N，句子的索引为 n，其中 $0<n<N+1$。设第 t 轮迭代时，处理了第 n 句之后，模型的权值为 $w_{t,n}$，则传统的感知器算法训练得到的模型的权值为 $w_{T,N}$。

此权值可以使得模型在训练集上取得较高的预测精度，但是容易造成过拟合现象，使得模型在测试集上的预测精度不高。平均感知器策略为了防止过拟合，并不使用 $w_{T,N}$ 作为最终权值，而是使用 $\frac{1}{NT}\left(\sum_{t=1\ldots T,n=1\ldots N} w_{t,n}\right)$ 作为模型的权值。

3.3 特征模板的设计

基于转移的句法分析方法的一个重要的优点在于，可以在特征模板中使用任意复杂的特征来进行句法分析，因此如何设计特征模板直接影响到句法分析的精度。

原则上，使用的特征数量越多、组合越复杂，则模型的表达能力越强，句法分析的精度越高。然而实践中，过于复杂的特征具有很高的稀疏性，对于句法分析提供的信息十分有限。同时，特征数量过多会导致特征提取的时间过长，进而使得训练和解码的速度均减慢。

对于汉字粒度的中文句法分析器，使用的特征可以分为两类：结构特征、字符串特征。前者主要反映了句子的句法结构信息，它深刻地影响着句法分析任务的精度。字符串特征主要反映了句子中单词的信息，它对分词和词性标注任务的精度影响较大。

3.3.1 结构特征

结构特征利用了生成句法树过程中各种与树结构相关的信息指导句法分析的决策，主要包括通用的结构特征 [17] 以及与汉字相关的特征 [15]，分别如图 4 和图 5 所示。

图 4 中，S_0,S_1,S_2,S_3 分别表示位于栈顶的四个结点；S_{0l},S_{0r},S_{0u} 分别表示结点 S_0 的左子结点、右子结点和唯一子结点；Q_0,Q_1,Q_2,Q_3 分别表示位于输入队列队首的四个汉字。n 是一个布尔变量，表明这个结点是否是单词内部结构树的结点；w 表示这个结点的单词；t 表示这个结点的词性标签，c 表示这个结点的句法标签；j 随着结点类型的不同而不同，当结点的句法标签为空时，j 表示结点的词性标签，否则 j 表示结点的句法标签。这些特征是基于转移的短语结构句法分析经常使用的经典特征，是基于转移的短语结构句法分析的基础。

doi:10.3772/j.issn.2095-915x.2017.03.003

图 5 中，z 表示输入队列的对应位置的汉字，isNumOrAlpha 是一个函数，返回一个布尔值。当传递给它的变量为数字或拉丁字母时，这个函数返回 1，否则这个函数返回 0。这些特征主要利用了单词内部汉字之间的结构关系指导句法分析。

S_0nwt	S_0nc	S_0nct	S_0nwc	S_1nwt	S_1nc	S_1nct	S_1nwc
S_2nct	S_2nwc	S_3nct	S_3nwc	Q_0w	Q_0t	Q_1w	Q_1t
Q_2w	Q_2t	Q_3w	Q_3t	$Q_0w.Q_1w$		$Q_1w.Q_2w$	
$Q_2w.Q_3w$		$Q_0w.Q_1w.Q_2w$			$Q_1w.Q_2w.Q_3w$		
$S_0nwc.S_1nwc$		$S_0nc.S_1nwc$		$S_0nwc.S_1nc$		$S_0nc.S_1nc$	
$S_{0l}nct$	$S_{0l}nwc$	$S_{0r}nct$	$S_{0r}nwc$	$S_{0u}nct$	$S_{0u}nwc$	$S_{1l}nct$	$S_{1l}nwc$
$S_{1r}nct$	$S_{1r}nwc$	$S_{1u}nct$	$S_{1u}nwc$	$S_0wc.Q_0w$		$S_0c.Q_0w$	
$Q_0t.Q_1t$		$S_0w.S_1c.Q_0t$			$S_0c.S_1w.Q_0t$		
$S_0c.S_1c.Q_0w$			$S_0c.S_1c.Q_0t$			$Q_0w.Q_1t$	
$S_0c.S_1t.Q_0t$			$S_0wc.Q_0t.Q_1t$			$Q_0t.Q_1w$	
$S_0c.Q_0wt.Q_1t$			$S_0c.Q_0t.Q_1wt$			$S_1c.Q_0t$	
$S_0c.Q_0t.Q_1t$			$S_0t.Q_0t.Q_1t$			$S_1w.Q_0t$	
$S_0w.S_1c.S_2c$			$S_0c.S_1w.S_2c$			$S_1c.Q_0w$	
$S_0c.S_1c.S_2w$			$S_0c.S_1c.S_2c$			$S_1wc.Q_0w$	
$S_0t.S_1t.S_2t$			$S_0c.S_{0r}c.Q_0w$			$S_0c.Q_0t$	
$S_0c.S_{0l}c.S_1c$			$S_0c.S_{0l}j.S_1j$			$S_0w.Q_0t$	
$S_0c.S_{0l}c.S_1w$			$S_0c.S_1c.S_{1r}c$				
$S_0j.S_1c.S_{1r}j$			$S_0w.S_1c.S_{1r}c$				

图4 通用的结构特征

S_0nz	S_0nzc	S_0nzct	S_1nz	S_1nzc	S_1nzct	$S_0nzc.S_1nc$
isNumOrAlpha(S_0z).isNumOrAlpha(S_1z)					S_2nz	S_2nzc S_2nzct
isNumOrAlpha(S_0z).isNumOrAlpha(S_1z).S_1t						
$S_0nzt.S_1nzt$		$S_0nz.S_1nz$		$S_0nzc.S_1nzc$		$S_0nc.S_1nzc$
$S_0nz.S_1nz.S_2nz$			$S_0nz.S_1nc.S_2nc$			$S_0nz.Q_0z$
$S_0nc.S_1nz.S_2nc$			$S_0nc.S_1nc.S_2nz$			$S_0nzc.Q_0z$
$S_0nz.Q_0z.Q_1z$			$S_1nz.Q_0z.Q_1z$			$S_1nz.Q_0z$
$S_0nz.Q_0z.Q_1z.Q_2z$				$S_1nz.Q_0z.Q_1z.Q_2z$		
$S_1nzc.Q_0z$		$S_1nz.S_0nz.Q_0z$				
$S_1nz.S_0nz.Q_0z.Q_1z$						

图5 汉字相关的结构特征

3.3.2 字符串特征

字符串特征利用了当前处理的汉字及其前后汉字的信息，以及当前处理的单词及其前后单词的信息，指导分词和词性标注。在本文的基于汉字的句法分析器中，分词、词性标注和句法分析是在同一个框架下完成的，因此与分词和词性标注相关的字符串特征对于生成的最终的汉字粒度句法树的正确性有着深刻的影响。与结构特征不同，字符串特征只在特定的动作被执行时使用。图 6、图 7 和图 8 分别给出了

在“移进-分裂”、“移进-附着”和“归约-单词”动作被执行时使用的字符串特征。

图 6 中，z_0 表示当前输入队列队首的汉字，t_0,t_{-1},t_{-2} 分别表示当前单词和它之前两个单词的词性标签，w_0,w_{-1},w_{-2} 分别表示当前单词和它之前的两个单词，l_{-2} 表示当前单词和之前单词的长度，start 函数和 end 函数分别返回单词的第一个汉字和最后一个汉字。图 7 和图 8 中符号的含义与图 6 相同。这些特征利用了汉字的信息，使得分词和词性标注能够取得较高的精度。

t_{-1}	w_{-1}	t_0	z_0	$t_{-1}.l_{-1}$
$t_{-2}.t_{-1}$		$z_0.t_{-1}$		
$t_{-2}.t_{-1}.l_{-1}$			$z_0.t_{-1}.t_{-2}$	
$w_{-1}.t_{-2}$		$w_{-2}.t_{-1}$		
$t_{-1}.\text{end}(w_{-1}).\text{start}(w_0)$				

图6 “移进–分裂”时使用的字符串特征

$z_{-1}.z_0$	$z_{-1}.z_0.t_{-1}$	$z_0.t_{-1}$	$\text{start}(w_{-1}).z_0.t_{-1}$

图7 “移进–附着”时使用的字符串特征

$w_{-2}.w_{-1}$	$\text{start}(w_{-1}).\text{end}(w_{-1})$	$\text{start}(w_{-1}).\text{len}(w_{-1})$	$\text{end}(w_{-1}).\text{len}(w_{-1})$
$w_{-1}.\text{end}(w_{-2})$	$\text{end}(w_{-1}).\text{end}(w_{-2})$	$w_{-2}.\text{len}(w_{-1})$	$w_{-1}.\text{len}(w_{-2})$
$w_{-1}.t_{-1}$	$w_{-1}.t_{-2}$	$\text{end}(w_{-1}).t_{-1}$	$\text{start}(w_{-1}).w_{-1}$
$\text{start}(w_{-1}).z_{-1}$	$z_{-1}.z_0$	$w_{-1}.z_0$	$\text{start}(w_{-1}).z_0$
$w_{-1}.\text{end}(w_{-2}).t_{-1}$	$t_{-2}.t_{-1}.\text{len}(w_{-1})$		w_{-1}
$w_{-1}.t_{-1}.z_0$		$w(-1)$, where $\text{len}(w(-1))=1$	
$z(-2).z(-1).z(0).t(-1)$, where $\text{len}(w(-1))=1$			
$c.t_{-1}.\text{end}(w_{-1})$, where $c \in w_{-1} \wedge c \neq end(w_{-1})$			

图8 “归约–单词”时使用的字符串特征

4 语料库构建

对于本文的句法分析任务，语料库的构建包括两个部分：二叉化短语结构树库的构建、科技领域单词内部结构树的构建。

4.1 二叉化短语结构树库的构建

根据句法分析的原理可以看到，我们的句法分析器生成的句法树均为二叉树的形式。为了训练这个句法分析器，我们同样需要对训练语料中的句法树进行二叉化处理。

对于一棵句法树中每个子结点数大于 2 的结点，我们首先使用一个预先制定的规则 [20] 寻找头结点，之后分别处理头结点右侧结点与左侧结点。二叉化算法的伪代码如图 9 所示。

4.2 科技领域单词内部结构树的构建

为了使用单词内部汉字之间的结构信息指导句法分析以及生成汉字粒度的句法树，我们需要对单词内部汉字之间的关系进行标注。为了与短语结构的二叉化结构树统一，我们在构建单词内部结构树时，也采用二叉化的结构。同时，为了反映各个汉字之间的关系，我们为每个结点增加了“方向”信息。方向有三种：左（l）、右（r）、并列（c），分别表示两个子结点中表示核心语义的结点为左子结点、右子结点、以及两个子结点的地位相同的情形。

doi:10.3772/j.issn.2095-915x.2017.03.003

Algorithm 3 将多叉句法树转化为二叉树的算法

```
1: procedure BINARIZATION(Tree T)
2:     for node Y = X_1 ... X_m ∈ T do
3:         if m > 2 then
4:             find the head node X_k(1 ≤ k ≤ m) of Y
5:             m' = m
6:             while m' > k ∧ m' > 2 do
7:                 new node Y* = X_1 ... X_{m'-1}
8:                 Y ← Y * X_{m'}
9:                 m' = m' − 1
10:            end while
11:            n' = 1
12:            while n' < k ∧ k − n' > 1 do
13:                new node Y* = X_{n'} ... X_k
14:                Y ← X_{n'} Y*
15:                n' = n' + 1
16:            end while
17:        end if
18:    end for
19: end procedure
```

图9 句法树二叉化算法伪代码

为了使我们的句法分析器能够用于科技领域，我们基于科技领域语料完成了单词内部结构树库的构建。这个单词内部结构树库中，共有 2017 个科技领域的高频词。图 10 给出了科技领域单词内部结构树的一个实例。

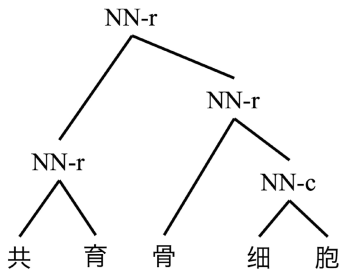

图10 一个科技领域单词内部结构树

5 实验结果

5.1 实验配置

我们在两个领域的语料上测试了本句法分析器的性能：新闻语料、科技语料。对于新闻语料，我们使用了中文句法分析任务的标准语料 Chinese TreeBank 5.0 (CTB5)[5] 进行训练和测试。其中，训练集、开发集和测试集的划分如表 1 所示。

表1 CTB5训练集、开发集、测试集的划分

	文件	句子数量
训练集	Sec. 001 – Sec. 815	16076
开发集	Sec.900 – Sec.931	804
测试集	Sec.860 – Sec.885	1905

对于科技语料，我们使用了面向中文 - 日文科技文献机器翻译任务的平行语料库的中文部分。考虑到人工标注句法树的成本，我们只对 1000 个句子进行了标注。同时，为了测试本句法分析器的领域迁移能力，在测试科技语料的句法分析性能时，我们使用 CTB5 的训练集进行训练，使用这 1000 个科技领域的句子进行测试。

在评估实验结果时，本句法分析器使用了与文献 [15] 相同的评估脚本[②]。该脚本可以输出分词、词性标注、单词内部结构、短语结构句法分析、树结构的精确率、召回率和 F-score。所有实验均在哈尔滨工业大学机器智能与翻译研究室的第 114 号服务器上进行。服务器的 CPU 为 48 核心的 Intel Xeon CPU E7-4807，主频 1.87GHz，内存 512GB。

5.2 句法分析结果

表 2 和表 3 分别给出了本句法分析器在新闻语料和科技语料上的句法分析统计结果。

表2　新闻语料句法分析结果

	精确率	召回率	F-score
分词	97.91	99.03	98.47
词性标注	93.80	94.88	94.33
单词结构	93.39	94.46	93.92
短语句法分析	82.46	82.73	82.60
整棵句法树	91.24	91.31	91.27

表3　科技语料句法分析结果

	精确率	召回率	F-score
分词	97.05	97.36	97.20
词性标注	94.29	94.59	94.44
单词结构	91.41	91.70	91.55
短语句法分析	89.75	92.07	90.89
整棵句法树	92.07	92.62	92.35

可以看到，我们的句法分析器在新闻语料和科技语料上均可以取得较好的句法分析效果。同时，作为一个副产品，本句法分析器可以同时完成分词和词性标注，并且在这两个任务上也可以取得相当好的结果。

表 4 比较了本文的句法分析器与目前普遍使用的句法分析器在科技语料上的短语结构句法分析结果。

表4　本句法分析器与普遍使用句法分析器的比较

句法分析器	F-score
[10]	85.23
[17]	87.82
[15]	89.56
[2]	89.67
[4]	90.32
本句法分析器	90.89

可以看到，与其他的句法分析器相比，本项目的句法分析器在科技语料上取得了最好的句法分析结果。

6 结论

本文介绍了由哈尔滨工业大学机器智能与翻译研究室开发的一个面向科技语料的中文短语结构句法分析器。本句法分析器的处理粒度为汉字，不需要进行分词和词性标注等预处理，从而减少了误差传播。此外，根据科技语料的特点，我们对特征模板进行了优化，同时构建了一个基于科技语料的单词内部结构树。实验结果表明，我们的句法分析器在通用的新闻语料以及科技领域语料上均取得了较好的效果。

参考文献

[1] Manning C D, Schutze H. Foundations of Natural Language Processing [M]. USA MIT Press, 2005: 258.

②脚本地址：https://github.com/zhangmeishan/wordstructures

doi:10.3772/j.issn.2095-915x.2017.03.003

[2] Zhu M, Zhang Y, Chen W, et al. Fast and Accurate Shift-Reduce Constituent Parsing[C]// Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. 2013: 434-443.

[3] Zhang Y, Clark S. A Fast Decoder for Joint Word Segmentation and POS-Tagging Using a Single Discriminative Model[C]// Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, Mit Stata Center, Massachusetts, USA, A Meeting of Sigdat, A Special Interest Group of the ACL. DBLP, 2010: 843-852.

[4] Wang Z, Xue N. Joint POS Tagging and Transition-based Constituent Parsing in Chinese with Non-local Features[C]// Meeting of the Association for Computational Linguistics. 2014: 733-742.

[5] Xue N, Xia F, Chiou F D, et al. The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus[J]. Natural Language Engineering, 2005, 11(2): 207-238.

[6] Baker J K. Trainable Grammars for Speech Recognition[C]// Speech Communication Papers for the, Meeting of the Acoustical Society of America. 1979: 547-550.

[7] Booth T L, Thompson R A. Applying Probability Measures to Abstract Languages[J]. IEEE Transactions on Computers, 1973, C-22(5): 442-450.

[8] Briscoe T, Carroll J. Generalized Probabilistic LR Parsing of Natural Language (Corpora) with Unification-Based Grammars[J]. Computational Linguistics, 1993, 19(1): 25-59.

[9] Sagae K, Lavie A. A Classifier-based Parser with Linear Run-time Complexity[C]// International Workshop on Parsing Technology. Association for Computational Linguistics, 2005:125-132.

[10] Wang M, Sagae K, Mitamura T. A fast, Accurate Deterministic Parser for Chinese[C]// ACL 2006, International Conference on Computational Linguistics and, Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July. DBLP, 2006: 425-432.

[11] Socher R, Bauer J, Manning C D, et al. Parsing with Compositional Vector Grammars[C]// Meeting of the Association for Computational Linguistics. 2013: 455-465.

[12] VINYALS O, KAISER L, KOO T, et al. Grammar as a Foreign Language [J]. Advances in Neural Information Processing Systems, 2015: 2773-2781.

[13] Team T D, Alrfou R, Alain G, et al. Theano: A Python Framework for Fast Computation of Mathematical Expression [M]. arXiv e-prints, 2016.

[14] Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems [EB/OL]. [2016-03-16]. https://arxiv.org/pdf/1603.04467v1.

[15] Zhang M, Zhang Y, Che W, et al. Chinese Parsing Exploiting Characters[C]// Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. 2013: 125-134.

[16] Collins M, Roark B. Incremental Parsing with the Perceptron Algorithm[C]// Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain. DBLP, 2004: 111-118.

[17] Zhang Y, Clark S. Transition-based Parsing of the Chinese Treebank using a Global Discriminative Model[C]// International Conference on Parsing Technologies. Association for Computational Linguistics, 2009: 162-171.

[18] Hogan D. Coordinate Noun Phrase Disambiguation in a Generative Parsing Model[J]. Association for Computational Linguistics, 2007(2007).

[19] Martí, Nez C, Prodinger H. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms[C]// ACL-02 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2002: 1-8.

[20] Zhang Y, Clark S. A Tale of Two Parsers: Investigating and Combining Graph-based and Transition-based Dependency Parsing using Beam-search[C]// Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A Meeting of Sigdat, A Special Interest Group of the ACL. DBLP, 2008: 562-571.