文章编号：1003-0077(2018)12-0048-09

基于转移的中文篇章结构解析研究

孙 成,孔 芳

(苏州大学 计算机科学与技术学院,江苏 苏州 215006)

摘 要：篇章结构解析作为篇章分析的子任务,对于篇章理解和下游篇章应用至关重要。该文基于中文连接依存树篇章标注语料,利用转移系统和深度学习的方法,给出了一个完整的从平文本到树形结构的篇章结构自动解析框架。该文统计了中文篇章语料的基本特点,提出了针对树形篇章结构的评测方法,并采用不同的方法对篇章解析过程的篇章子结构进行分布式表示,对比了不同方法下篇章结构解析的性能。

关键词：篇章分析;中文篇章结构;转移系统

中图分类号：TP391 **文献标识码**：A

A Transition-based Framework for Chinese Discourse Structure Parsing

SUN Cheng, KONG Fang

(School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China)

Abstract: As a subtask of discourse analysis, generating a proper discourse structure is critical for discourse comprehension and downstream discourse applications. Based on Chinese discourse treebank annotated under connective-driven dependency tree schema, a complete Chinese discourse structure generating framework is proposed. A statistical result on Chinese discourse corpus is given along with an evaluation protocol to measure the performance of discourse parser. The effectiveness in encoding discourse substructure is also compared between different distributed representation approaches.

Keywords: discourse parsing; Chinese discourse structure; transition-based system

0 引言

篇章是句子层级之上的自然语言单位。篇章分析技术旨在研究篇章内部的语义关系,从整体上理解篇章[1]。篇章分析技术由底向上可以分为:篇章理论的建立、篇章解析器的自动构建和篇章分析技术的上层应用。其中,篇章解析器的自动构建研究是篇章分析技术的核心。近年来,随着中文篇章理论的不断完善和篇章资源构建工作的进展[2-5],数据驱动的中文篇章解析器的自动构建成为可能。

本文使用汉语连接依存树篇章结构语料(Chinese connective-driven discourse treebank, CDTB)[5](以下简称汉语篇章结构语料),其结合了 RST[6] 的树形层次结构、PDTB[7] 的连接词以及汉语复句理论[8]的单复句的理论优点,自底向上地将子句(即篇章基本单元,以下简称子句)在连接词的驱动下组合为一棵篇章结构树,用连接词表征篇章关系的类型,并在内部节点中标出篇章关系和核心位置。针对该语料构建自动篇章解析器可以大致划分为三个子任务:①子句的自动分割,也就是篇章基本单元(EDU)的识别;②篇章树形结构的解析,即以段落为基本单元,迭代地将 EDU 构建成上层节点,并最终组织成一棵独立的篇章树;③篇章关系识别与分类,即判别联接形成上层节点的 EDU 间表达的逻辑语义关系。其中,中文篇章关系识别的相关研究和工作较多,方法也更为完善[9-11],本文并不关注这一任务。相对而言,篇章结构对于篇章级别的上层应用非常有帮助[12-13],本文研究的重点是中文篇章树形结构自动生成。此外,我们认为篇章树形结构对于篇章

收稿日期：2018-09-29 **定稿日期**：2018-10-29

基金项目：国家自然科学基金(61472264,61751206);国家重点研发计划子课题(2017YFB1002101);国家自然科学基金(61502149)

关系类型的判断也起着至关重要的约束作用。

本文的主要工作包括：详细分析了中文篇章结构语料的统计特征；提出了一个完整的从篇章纯文本到篇章树形层次结构的解析框架，及可供参考的性能评价标准；对比使用不同方式对子句进行特征表示对最终性能的影响表明，浅层词法特征仍然在篇章结构解析中发挥主要作用；最后，本文分析了结构解析中误差传播对于最后性能的影响。文章的组织如下：第 1 节首先介绍实验使用的中文篇章语料以及相关的统计结果；第 2 节介绍相关工作，并详述本文使用的篇章结构解析框架；第 3 节给出实验配置、实验结果以及对实验结果的分析；最后一节是对当前工作的总结和展望。

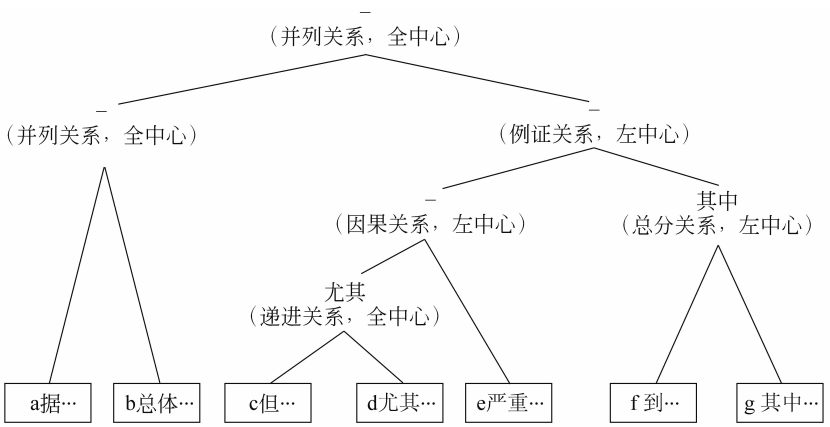
1　汉语连接依存树篇章语料

1.1　连接依存树篇章结构理论

图 1 给出了 CDTB 语料中文件名为 chtb_0083.xml 文档中编号 401 的篇章结构树。其中叶子节点用于表示子句。连接依存理论[14]中，子句定义为复句理论中的单句或者复句的分句，其至少包含一个谓语结构和表示一个完整主题。篇章结构树的中间节点为篇章关系节点，关系节点将下一级的篇章子结构组合为上一级的篇章结构，最终形成一棵完整的篇章树。篇章子结构的组合可由连接词驱动，形成的篇章关系称为显式关系；而缺乏显式连接词指导，由语义关系的组合则称为隐式篇章关系。和修辞结构理论相似，连接依存理论中，篇章关系也有主次之分，核心部分表达文章主体意图，次要部分起对核心部分修饰和补充的功能。

1.2　语料统计

汉语篇章结构语料以段落为单位，在宾州中文树库 6.0 版本(CTB 6.0)的基础上标注了 500 篇文档，包含段落数 2 342 个，每个段落独立标注为篇章结构树。共包含 10 638 个子句，平均每个篇章结构树包含 4.54 个子句。篇章结构树的平均高度为 2.43，平均包含 3.12 个关系节点。语料中，篇章关系按核心位置分为三个类别：核心在左、核心在右和多个核心(所有子节点都为核心)。表 1 给出了篇章关系中，各个核心位置的数量和占比。



a. 据卢仁法介绍，经过全国税务系统八十万干部的努力，今年全国工商税收收入任务已提前十一天超额完成，

b. 总体情况比较好。

c. 但企业欠缴税款的情况比较严重，

d. 尤其是企业欠缴增值税和消费税这“两税”的问题突出，

e. 严重影响了今年“两税”任务的完成。

f. 到十一月底，全国企业欠缴的工商税款累计达到了四百一十九亿元，

g. 其中欠缴国内增值税和消费税就达三百零七亿元。

图 1　CDTB 语料中文件名为 chtb_0083.xml 文档编号 401 的连接依存篇章结构树图示

下划线表示连接词，图例中“_”表示缺省连接词

表 1　篇章关系核心位置数量和占比

核心位置	数量	占比/%
核心在左	2 108	28.85
核心在右	1 446	19.79
多个核心	3 753	51.36

不同于 RST-DT 将整个篇章标注为一棵篇章结构树的策略，汉语篇章结构语料以段落为单位，包含子句更少，树的高度也更低，而且子句边界都有标点符号作为标识，给篇章结构解析中的树形结构的建立带来帮助。然而，中文篇章结构语料中隐式篇章关系占比达 75.2%。另外，核心位置分布不平衡，多个核心的关系占比达 51.3%，使得篇章核心位置的自动判定成为挑战。

2 汉语篇章结构解析框架

2.1 相关工作

篇章结构解析是一类结构预测(structured prediction)任务。对于给定的子句序列，模型需要预测出组织这些子句的最合适的篇章结构。目前的篇章解析工作大多参照句法解析的方法，可以分为基于图(如 Cocke-Younger-Kasami，CYK 算法)的解析方法和基于转移系统的解析方法。其中，基于转移系统的方法将篇章树的构建过程看成是一个状态转移路径的搜索过程。一个确定的状态转移序列对应一棵确定的篇章树形结构，从而将结构预测问题转换成为对每个状态正确预测下一个局部转移动作的分类问题。基于转移的解析策略因其具有线性的时间复杂度并取得了与 CYK 等基于图的解析方法相当的性能，因而被广泛应用。

相较于中文，英文篇章解析研究工作开展较早，其中关于篇章结构解析任务主要围绕 RST-DT 展开。其中具有代表性的有：Hernault 等开发的基于线性 SVM、采用自底向上贪婪策略的 HILDA 篇章解析器[15]；Feng 和 Hirst 提出了一种两步解析的方法，分别使用两个线性链条件随机场模型依次构建句内和句间的篇章结构[16]；Ji 和 Eisenstein 等尝试通过线性变换，将稀疏的子句特征转换为低维度的向量表示，并通过线性 SVM 预测转移系统的动作[17]；Jiwei Li 等基于递归神经网络对篇章子结构进行建模，使用表解析的方法完成自动篇章分析[18]。中文方面，由于篇章理论不够完善，篇章自动分析技术发展缓慢。可供参考的工作有：孔芳等基于 CDTB 语料采用流水线的方式构建的端到端的中文篇章解析器[19]；吴永芃等为简化篇章分析难度，提出了篇章依存树的篇章结构体系[20]。

2.2 基于转移的结构预测方法

本文使用的 Shift-Reduce 转移系统可以由式(1)所示的五元组定义：

$$SR = (C, T, C_s, C_t, o) \quad (1)$$

C：所有可行的状态，每个状态包含一个栈和一个队列结构。队列中存放待处理的子句，栈中存放正在处理的篇章子结构(包含子句和篇章树的子树)。

T：每个状态对应的可行动作，在表 2 中给出。Shift 动作执行是会从队列中弹出顶部子句，压入栈中；Reduce 动作将栈顶的头两个树节点组合成一个节点，并将合并的两个节点作为新节点的子节点。

C_s：初始状态集合，栈为空，队列中依次存放篇章所有的子句。

C_t：终止状态集合，队列为空，栈中之包含篇章结构树的根节点，终止状态不能执行任何转移动作。

o：Oracle 函数，给定状态，始终返回该状态生成正确篇章结构的下一个转移动作。从初始状态，不断执行 Oracle 函数返回的动作，最后到达终止状态的状态转移序列对应一棵正确的篇章结构树。

表 2 Shift-Reduce 状态转移动作

动作	转移前状态	转移后状态	条件
Shift	$([S],[q,Q])$	$([S,q],[Q])$	$Q \notin \varphi$
Reduce	$([S,m,n],[Q])$	$([S,(m,n)],[Q])$	栈中至少有两个元素

图 1 的篇章树形结构可以由初始状态在 Oracle 函数的指导下执行多次状态转移确定。表 3 给出了该正确的状态转移序列。可以看出，Shift-Reduce 转移系统自底向上地构建出篇章结构树。给定一棵二元的篇章树，Oracle 函数的输出动作序列可通过从标准二元篇章结构树的根节点开始后序遍历变形实现，遍历中遇到叶子节点则输出 Shift 动作，遇到关系节点则输出 Reduce 动作。由于 Shift-Reduce 生成的树结构为二叉树，所以状态转移次数为 $2n-1$，其中 n 为篇章中包含的子句数目。

表 3 状态转移序列示例

时刻 t	状态 C_t	动作 $o(C_t)$
0	$[\varnothing],[a,b,c,d,e,f,g]$	SHIFT
1	$[a],[b,c,d,e,f,g]$	SHIFT
2	$[a,b],[c,d,e,f,g]$	REDUCE
3	$[r1=(a,b)],[c,d,e,f,g]$	SHIFT
4	$[r1,c],[d,e,f,g]$	SHIFT
5	$[r1,c,d],[e,f,g]$	REDUCE

续表

时刻 t	状态 C_t	动作 $o(C_t)$
6	$[r1,r2=(c,d)],[e,f,g]$	SHIFT
7	$[r1,r2,e],[f,g]$	REDUCE
8	$[r1,r3=(r2,e)],[f,g]$	SHIFT
9	$[r1,r3,f],[g]$	SHIFT
10	$[r1,r3,f,g],[\varnothing]$	REDUCE
11	$[r1,r3,r4=(f,g)],[\varnothing]$	REDUCE
12	$[r1,r5=(r3,r4)],[\varnothing]$	REDUCE
13	$[r6=(r1,r5)],[\varnothing]$	—

2.3　基于 SPINN 模型的状态动作预测

由 2.1 节可知，基于转移的篇章结构解析的关键在于构建模型模仿 Oracle 函数，因此可以看成对给定状态进行分类的问题。为了在确定结构的同时确定篇章关系的核心位置，我们将 Reduce 动作扩展为 Reduce-左核心、Reduce-右核心和 Reduce-全核心。总的分类标签数变成 4 个，是一个结构生成和核心关系判定的联合模型。

神经网络和文本分布式表示的应用使得在自然语言处理领域取得了突破性的进展。本文使用 SPINN（Stack-augmented Parser-Interpreter Neural Network）模型[21]，对历史转移状态序列、篇章子结构进行分布式表示，得到当前状态的表示，用于预测下一个转移动作。

2.3.1　SPINN 模型

图 2 给出了 SPINN 模型在表 3 中时刻 4 的模型图示。SPINN 用一个 tracking LSTM 网络维护整个解析过程的历史转移状态。每个时刻 tracking LSTM 输入栈的头两个元素和队列的头一个元素，

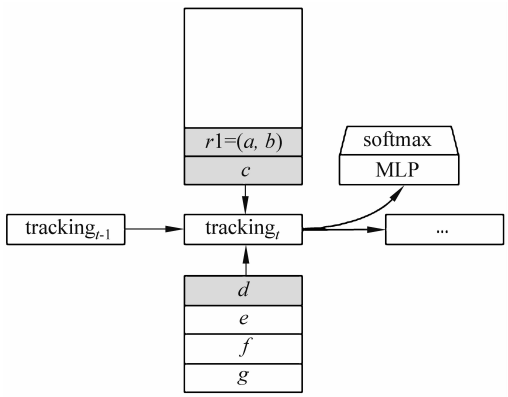


图 2　SPINN 模型示例

其中如果栈的头两个元素结合紧密则应该执行 Reduce 操作，而栈的头元素与队列的头元素决定是否将队列的头元素移入栈内。时刻 t 的解析状态 tracking$_t$ 用于预测时刻 t 下一个动作的概率分布，并且作为下一个时刻的输入表示解析状态的历史信息。在一个解析过程中，模型的状态随着状态转移不断推进。我们使用动态计算图而不是每次转移后重新生成计算图，以保留模型的状态。

转移状态序列的每一个状态包含一个栈和一个队列结构。栈和队列的元素对应篇章树的某个叶子节点（子句节点）或者中间节点（关系节点）。下文中，我们将每个节点编码为一个状态向量 $\boldsymbol{c}$ 和一个输出向量 $\boldsymbol{h}$。如栈顶的头两个元素和队列的头一个元素分别表示为：$(\boldsymbol{h}^{s1},\boldsymbol{c}^{s1})$，$(\boldsymbol{h}^{s2},\boldsymbol{c}^{s2})$ 和 $(\boldsymbol{h}^{b1},\boldsymbol{c}^{b1})$。$t$ 时刻 trackingLSTM 的输入为 $\boldsymbol{h}^{s1},\boldsymbol{h}^{s2},\boldsymbol{h}^{b1}$ 三个向量的拼接。trackingLSTM 在时刻 t 的输出记为 $\boldsymbol{h}_t^{\text{tracking}}$，该输出首先经过一个两层的 Relu 激活函数的多层感知机，然后经过一个 softmax 层归一化为下一个转移动作的概率分布，如式(2)、式(3)所示。

$$h_t^{\text{MLP}}=\max(0,W_{\text{MLP}}\,h_t^{\text{tracking}}+b_{\text{MLP}}) \tag{2}$$

$$p_{t+1}=\text{softmax}(h_t^{\text{MLP}}) \tag{3}$$

2.3.2　子句的特征表示

栈和队列中每个元素，对应部分解析的篇章结构树的某个叶子节点（子句节点）或者内部节点（关系节点）。每个元素分别表示为一个状态向量 $\boldsymbol{c}$ 和一个输出向量 $\boldsymbol{h}$。我们首先介绍如何使用不同的方法对子句进行特征表示，不同表示方法的性能差异在第 3 节中以最后性能评价的方式给出。

1. 基本特征

首先我们使用规则的方法从子句中提取启发式的特征作为子句表示的性能基准，包括第一个词和最后一个词的 Embedding，其中 Word Embedding 在 GigaWords 上预训练，并在训练中不断优化；另外还有第一个词的词性的向量表示，词性由 Berkeley Parser 解析获得。词向量的维度为 50，词性的 Embedding 维度为 15。

将上述特征向量拼接之后通过一个线性变换得到子句的 $\boldsymbol{c}$ 和 $\boldsymbol{h}$ 的表示，如式(4)所示。

$$\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{h}\end{bmatrix}=W_{\text{edu}}\begin{bmatrix}\boldsymbol{w}_1\\ \boldsymbol{w}_{-1}\\ \boldsymbol{p}_1\end{bmatrix}+b_{\text{edu}} \tag{4}$$

2. 基本特征 ＋ Average BOW Embedding

除基本特征外，Average BOW Embedding 的子句表示方法对子句中所有词对应的词向量 $\boldsymbol{w}_1$，

$\boldsymbol{w}_2,\cdots,\boldsymbol{w}_k$ 求和取平均，用上文同样的线性映射方法取得子句的 $\boldsymbol{h}$ 和 $\boldsymbol{c}$，如式(5)、式(6)所示。

$$f_{\text{bow}}=\frac{1}{k}\sum \boldsymbol{w}_i \tag{5}$$

$$\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{h}\end{bmatrix}=\boldsymbol{W}_{\text{edu}}\begin{bmatrix}\boldsymbol{w}_1\\ \boldsymbol{w}_{-1}\\ \boldsymbol{p}_1\\ \boldsymbol{f}_{\text{bow}}\end{bmatrix}+\boldsymbol{b}_{\text{edu}} \tag{6}$$

3. 基本特征 ＋ BiLSTM with Self-Attention

我们首先使用一个双向 LSTM 对子句的词序列进行正向和反向序列编码，将正反向 LSTM 的输出拼接后作为子句中词 $\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_k$ 编码为包含上下文序列信息的序列表示 $\boldsymbol{h}_{w1},\boldsymbol{h}_{w2},\cdots,\boldsymbol{h}_{wk}$。使用自注意力机制取得编码后的信息的权重，按权重将 h_{wi} 相加作为子句的特征表示。

$$\text{weight}_i=q\cdot\tanh(\boldsymbol{W}_{\text{attn}}\cdot\boldsymbol{h}_i+b_{\text{attn}}) \tag{7}$$

$$f_{\text{attn}}=\sum \text{weight}_i * h_{\text{wi}} \tag{8}$$

$$\begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{h}\end{bmatrix}=\boldsymbol{W}_{\text{edu}}\begin{bmatrix}\boldsymbol{w}_1\\ \boldsymbol{w}_{-1}\\ \boldsymbol{p}_1\\ \boldsymbol{f}_{\text{attn}}\end{bmatrix}+\boldsymbol{b}_{\text{edu}} \tag{9}$$

其中，向量 $\boldsymbol{q}$ 是注意力机制中的 Query，可以被解释为哪些 LSTM 序列编码输出对于篇章结构解析比较重要。$\boldsymbol{q}$ 在训练开始时随机初始化并随着训练学习合适的参数。

4. 基本特征 ＋ CNN

卷积神经网络广泛用于提取句子的词汇特征。首先，对于子句中的每一个词分别用 50 维、15 维、15 维的向量分布式表示词、词性和位置信息。词向量同上使用在 GigaWords 上预训练的词向量，词性和位置的 Embedding 随机初始化。拼接后的每个词的向量表示维度为 d_w。如图 3 所示，我们使用 60 个 $1\times d_w$、30 个 $2\times d_w$ 和 10 个 $3\times d_w$ 窗口大小的卷积核从子句中提取一元、二元和三元的词汇特征。然后使用最大池化方法挑选每个卷积核的最显著的词卷积特征，最后将池化后的特征与基础特征拼接起来，经过线性变换得到子句的 $\boldsymbol{c}$ 和 $\boldsymbol{h}$ 的表示。

2.3.3 关系节点的特征表示

受逻辑语义学的组合原则启发，我们相信篇章关系节点的语义与其组成部分的语义及其部分的组合方式密切相关。Tree-LSTM 已经被证明在组合语义的表示方面十分有效[22]，我们引入 Tree-LSTM

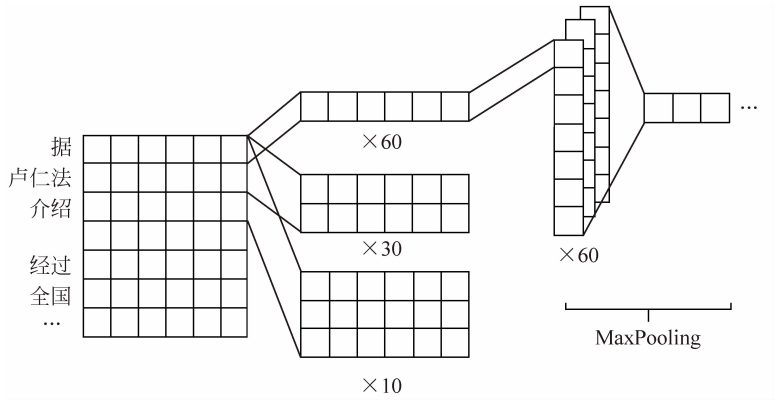


图 3 用于子句特征提取的 CNN 网络图示

作为转移状态栈中的部分篇章子树的组合语义表示方法。如表 3 中时刻 2，我们需要执行 Reduce 动作，并将 a 和 b 两个叶子节点组合为其父关系节点 $r1$，设此时根据上一节介绍的策略将 a 和 b 表示为 $(\boldsymbol{c}_a,\boldsymbol{h}_a)$，$(\boldsymbol{c}_b,\boldsymbol{h}_b)$。根据 Tree-LSTM 做如下推导，得到其 a、b 父节点 $r1$ 的特征表示，如式(10)～式(12)所示。

$$\begin{bmatrix}i\\ f_a\\ f_b\\ o\\ g\end{bmatrix}=\begin{bmatrix}\sigma\\ \sigma\\ \sigma\\ \sigma\\ \tanh\end{bmatrix}\left(W_{\text{comp}}\begin{bmatrix}h_a\\ h_b\\ h_{t-1}^{\text{tracking}}\end{bmatrix}+b_{\text{comp}}\right) \tag{10}$$

$$c_{r1}=f_a\cdot c_a+f_b\cdot c_b+i\cdot g \tag{11}$$

$$h_{r1}=o\cdot\tanh c_{r1} \tag{12}$$

3 实验配置

3.1 语料的划分与预处理

中文篇章语料(CDTB)[5]包含 500 篇文档，每个文档包含不等的若干以段落为单位的篇章树形结构。在分割之前为了方便处理，我们去掉了组织不良的编号为 150、193、208、287、300、644 的文档，其

他的文档按表 4 划分为训练集、开发集和测试集。由于 Shift-Reduce 框架只能生成二叉树，而中文篇章结构语料中的多核心节点可能包含多个孩子节点，所以我们在训练和评测之前会将所有的篇章树结构中多元节点通过不断归并，最后到两个孩子转换为二叉树。

表 4　语料划分

划分	训练集	开发集	测试集
编号范围	11～94 100,111～194 200,211～294 301～325 400～454 500,510～514 520～596 610～649 655～657	95～99 195～199 295～299 515～519 650～654	1～10 101～110 201～210 501～509 601～609
篇章数	1991	105	215

3.2　性能评价方法

相对于 RST 在整个篇章上标注篇章结构，中文篇章结构语料以段落为单位，树的高度和节点数都少得多。但 RST 的评价方法是将核心位置和篇章关系放在子节点上，如果子节点的边界位置和核心标注正确，则视为一个正确的节点。为了更好地反映篇章结构解析的性能，本次实验采用更为严格的性能评价方法。评价包括两方面：

① 节点结构正确的评价：对于一棵自动解析篇章结构树的内部节点，与其标准篇章结构树比对时，只有其子节点的边界位置全部正确，才能算作是一个结构正确的节点。

② 关系核心正确的评价：只有在结构正确的前提下，节点的核心位置判定正确，才能算作是一个核心位置正确的内部节点。

依据上述评价准则，我们得到了如下计算每棵树的准确率 P 和召回率 R，如式(13)、式(14)所示。

$$P=\frac{\text{解析正确的内部节点数}}{\text{解析的篇章树内部节点数}} \tag{13}$$

$$R=\frac{\text{解析正确的内部节点数}}{\text{正确的篇章树内部节点数}} \tag{14}$$

在求出测试集每棵树的准确率和召回率后，取平均求得平均准确率 P_{avg} 和平均召回率 R_{avg}，使用式(15)求得整个系统 $F1$ 性能指标：

$$F1=\frac{2\times P_{\text{avg}}\times R_{\text{avg}}}{P_{\text{avg}}+R_{\text{avg}}} \tag{15}$$

3.3　模型训练

模型的训练以单棵篇章树为单位。首先将篇章树的子句节点按照 2.2 节的方法编码为 (c^b,h^b)，然后在 Oracle 函数的引导下不断执行状态转移，并保存每一步的预测状态转移动作概率分布 p_i 和正确的转移动作 $o(c_t)$。对于每个转移状态，栈中的编码和元素 (c^s,h^s) 和队列中的编码后元素 (c^b,h^b) 以及 tracking LSTM 的隐藏状态 $(c^{\text{tracking}},h^{\text{tracking}})$ 的维度均为 128。假设一个篇章包含 n 个子句，则除终止状态外，构建对应篇章解析树有状态序列 $c_0,c_1\cdots c_{2n-2}$，每个状态由 SPINN 模型输出下一个动作的预测概率分布 $p_1,p_2\cdots p_{2n-1}$，损失函数定义为式(6)所示。

$$L=\frac{1}{2n-1}\sum_{t=0\cdots 2n-2}-\log p_{t+1}^{o(c_t)}+\lambda\|\theta\|_2 \tag{16}$$

L2 正则化项系数 λ 设置为 0.000 01，使用 Adam 优化器训练优化模型参数，学习速率设置为 0.001 并随着训练次数线性下降。

模型在训练集上迭代训练优化参数，保存每次训练过程中开发集上表现最好的模型用于最后性能评价的模型。

4　实验结果与分析

4.1　标准子句分割下的篇章结构解析的性能

标准子句分割是指，对于测试集中的标准篇章结构树，我们保留其子句的划分，去掉篇章树的内部关系节点，用上述模型重新构建篇章结构，最后使用 3.2 节介绍的性能评价方法给出最终的模型在结构构建和核心判定方面的性能。

表 5 给出了在标准子句分割下使用不同子句特征表示，最终取得的解析性能。值得指出的是，由于在标准分割的子句上构建关系，二叉树的内部节点数比关系节点数少 1，所以自动生成篇章结构树的内部数与标准篇章结构树的内部节点相等，所以准确率、召回率和 $F1$ 值在同一个指标下相等。由实验结果可知，在标准子句分割下，average BOW Embedding、BiLSTM with Attention 和 CNN 的子句表示的方法相对于只使用基本特征，都会带来性能提升。使用卷积神经网络对子句进行表示在结构和核心位置的判别上都取得了最好的性能，说明词汇

短语特征对于启发式的篇章结构分析仍然起主要作用，应该放到结构生成的首选位置。

表5 标准子句分割下的结构解析性能

子句表示	结构		结构+核心	
基本特征	*P*	0.789	*P*	0.492
	R	0.789	*R*	0.492
	*F*1	0.789	*F*1	0.492
基本特征+BOW	*P*	0.821	*P*	0.533
	R	0.821	*R*	0.533
	*F*1	0.821	*F*1	0.533
基本特征+BiLSTM with Attention	*P*	0.839	*P*	0.531
	R	0.839	*R*	0.531
	*F*1	0.839	*F*1	0.531
基本特征+CNN	*P*	0.840	*P*	0.539
	R	0.840	*R*	0.539
	*F*1	0.840	*F*1	0.539

4.2 自动子句分割下的篇章结构解析性能

中文篇章结构语料中，子句都有标点作为边界标识，但不是所有的标点都可以作为边界标识。比如图1中的f子句：

f. 到十一月底，全国企业欠缴的工商税款累计达到了四百一十九亿元，

“到十一月底”单独并不能表达一个主题，所以其后面的逗号不能作为边界的标识。我们参照文献[23]中的方法，对是否为边界标识包含歧义的标点分类。先使用文本中出现的“。!?”标点作为分割标识，将篇章文本切割为句子。然后依照文献[23]从逗号和分号的上下文中抽取词法、句法等特征，使用SVM分类器将句子中的逗号和分号分类为“是边界标识”和“不是边界标识”两个类别。逗号和分号的分类性能在表6中给出。

表6 逗号、分号是否为子句边界分类性能

	P	*R*	*F*1
正例	0.91	0.87	0.89
负例	0.94	0.95	0.94
平均	0.93	0.93	0.93

将分割后的子句作为输入，使用上述篇章结构解析框架构建篇章结构，最后测试集上的性能在表7中给出。

表7 自动子句分割下的结构解析性能

子句表示	结构		结构+核心	
基本特征	*P*	0.729	*P*	0.456
	R	0.740	*R*	0.470
	*F*1	0.735	*F*1	0.463
基本特征+BOW	*P*	0.754	*P*	0.521
	R	0.767	*R*	0.524
	*F*1	0.762	*F*1	0.523
基本特征+BiLSTM with Attention	*P*	0.771	*P*	0.489
	R	0.781	*R*	0.493
	*F*1	0.776	*F*1	0.491
基本特征+CNN	*P*	0.777	*P*	0.530
	R	0.786	*R*	0.534
	*F*1	0.782	*F*1	0.532

从性能中可以看出，虽然逗号、分号的分类性能达到0.93的*F*1值，但是由于我们采用严格的结构评价方法，边界分割的错误直接或间接影响篇章树上层结构，子句分割的错误会对最后结果造成较大影响。另外，在标准子句分割下，采用基本特征+BOW和基本特征+BiLSTM with Attention在核心位置检测上取得了相当的性能，而在自动子句分割下，采用基本特征+BiLSTM with Attention的核心位置检测性能相比基本特征+BOW配置有较大差距。我们的解释为：由于LSTM擅长捕获序列上长短距离的语义依赖关系，因而自动子句分割的错误更容易影响到LSTM的序列模型。

表8给出了自动子句分割、子句表示使用基本特征+CNN的配置下，核心位置判别的性能。再次强调，文本给出的评价机制中，只有在节点结构正确的前提下才能保证评价核心位置判别的正确性。自动子句分割和节点结构的性能都对核心位置性能的判断产生影响，因此表8中给出的核心位置判别的准确率和召回率都偏低。此外，最终的性能印证了1.2节对于核心位置统计结果的推断。篇章关系核心位置的分布不平衡和隐式篇章关系的高占比，使得核心位置的自动判定需要在数量不多的篇章语言资源的上学习篇章单元之间的深层语义信息，因而很难取得理想的判定效果。

表 8　整个结构解析框架的核心位置判定性能

核心位置	*P*	*R*	*F*1
核心在左	0.438	0.302	0.358
核心在右	0.290	0.242	0.264
多个核心	0.528	0.621	0.571

5　总结和展望

本文在中文篇章结构语料上，基于转移系统和神经网络的方法，构建了面向中文的篇章结构解析框架。其能够从纯篇章纯文本构建篇章树形结构，并具有线性的时间复杂度，有望为下游如篇章摘要生成、篇章级的情感分析等任务提供帮助。

相较于句子，篇章中远距离的语义依赖关系更多，中文在行文时句子的组织更为松散。自底向上的结构解析方式只从局部特征考虑篇章结构的组织，缺乏全局信息如篇章主题的指引。我们未来的工作将考虑篇章整体信息，尝试用自顶向下的方法依次分解篇章组成成分，完成篇章结构的构建。

参考文献

[1] 徐凡，朱巧明，周国栋. 篇章分析技术综述[J]. 中文信息学报，2013，27(3)：20-33.

[2] 周强. 汉语句法树库标注体系[J]. 中文信息学报，2004，18(4)：2-9.

[3] 胡金柱，等. 面向中文信息处理的复句关系词提取算法研究[J]. 计算机工程与科学，2009，31(10)：90-93.

[4] Zhou Y，Xue N. The Chinese Discourse TreeBank：A Chinese corpus annotated with discourse relations[J]. Language Resources and Evaluation，2015，49(2)：397-431.

[5] Li Y，Kong F，Zhou G. Building Chinese discourse corpus with connective-driven dependency tree structure[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)，2014：2105-2114.

[6] Mann W C，Thompson S A. Rhetorical structure theory：Description and construction of text structures [M]. Natural Language Generation. Springer，Dordrecht，1987：85-95.

[7] Miltsakaki E，et al. The Penn Discourse Treebank [C]//Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04). 2004.

[8] 邢福义. 汉语复句研究[M]. 北京：商务印书馆，2001.

[9] 徐凡，朱巧明，周国栋. 基于树核的隐式篇章关系识别[J]. 软件学报，2013，24(5)：1022-1035.

[10] 孙静，等. 汉语隐式篇章关系识别[J]. 北京大学学报（自然科学版），2014，50(1)：111-117.

[11] Huang H H，Chen H H. Chinese discourse relation recognition [C]//Proceedings of 5th International Joint Conference on Natural Language Processing，2011：1442-1446.

[12] Bhatia P，Ji Y，Eisenstein J. Better document-level sentiment analysis fromrst discourse parsing[J]. arXiv preprint arXiv:1509.01599，2015.

[13] Cheng S，Fang K，Guodong Z. Towards better Chinese zero pronoun resolution from discourse perspective[C]//Proceedings of National CCF Conference on Natural Language Processing and Chinese Computing. Springer，Cham，2017：406-418.

[14] 李艳翠. 汉语篇章结构表示体系及资源构建研究[D]. 苏州：苏州大学博士学位论文，2015.

[15] Hernault H，Prendinger H，Ishizuka M. HILDA：A discourse parser using support vector machine classification[J]. Dialogue and Discourse，2010，1(3)：1-33.

[16] Feng V W，Hirst G. A linear-time bottom-up discourse parser with constraints and post-editing[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics，2014(1)：511-521.

[17] Ji Y，Eisenstein J. Representation learning for text-level discourse parsing[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. 2014(1)：13-24.

[18] Li J，Li R，Hovy E. Recursive deep models for discourse parsing[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014：2061-2069.

[19] Kong F，Zhou G. A CDT-styled end-to-end Chinese discourse parser[J]. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)，2017，16(4)：26.

[20] 吴永芃，等. 中英文篇章依存树库构建与分析[J]. 中文信息学报，2018，32(1)：75-82.

[21] Bowman S R，et al. A fast unified model for parsing and sentence understanding[J]. arXiv preprint arXiv：1603.06021，2016.

[22] Tai K S，Socher R，Manning C D. Improved semantic representations from tree-structured long short-term memory networks[J]. arXiv preprint arXiv：1503.00075，2015.

[23] 李艳翠，等. 基于逗号的汉语子句识别研究[J]. 北

京大学学报（自然科学版），2013，49(1)：7-14.

孙成(1993—)，硕士研究生，主要研究领域为自然语言处理、篇章分析。
E-mail：20165227002@stu.suda.edu.cn

孔芳(1977—)，通信作者，博士，教授，主要研究领域为机器学习、自然语言处理、篇章分析。
E-mail：kongfang@suda.edu.cn

“中国法研杯”司法人工智能挑战赛(CAIL 2018)颁奖会举行

2018 年 10 月 12 日，首届“中国法研杯”司法人工智能挑战赛(CAIL 2018)颁奖典礼暨学术交流研讨会在国家审判资源信息管理中心举行。中国电子科技集团副书记、董事胡爱民，最高人民法院信息中心副主任孙福辉，中国中文信息学会副理事长、秘书长孙乐，共青团中央青年发展部副部长赵宝东，中国电科团委书记金铁增，中国司法大数据研究院总经理王珩、副总经理艾中良，清华大学副教授刘知远，北京大学副教授冯岩松等与参赛选手共同出席了活动。

本挑战赛立足于推动司法人工智能技术发展，为学术界和科技界提供统一评测平台，由最高人民法院信息中心、中国中文信息学会、共青团中央青年发展部担任指导单位，由中国司法大数据研究院、中国中文信息学会评测工委会、中国电科团委共同主办，由清华大学自然语言处理与社会人文计算实验室、北京大学计算机科学技术研究所、中国科学院软件研究所中文信息处理研究室共同承办。本挑战赛从 2018 年 5 月正式开赛，主要包括刑事案件的罪名预测、法条推荐和刑期预测等三个判决预测任务，共吸引来自全球的 269 家单位的 601 支队伍、1144 名选手报名，最终有 205 支队伍提交了比赛模型。本次挑战赛结束后，比赛使用的 200 多万份法律文书全部开放下载(下载方式和说明论文见文末链接)。

中文信息学会副理事长孙乐在致辞中对挑战赛高度肯定，阐述了人工智能技术的发展趋势，以及中文信息处理技术对司法人工智能的重要意义，并祝愿该挑战赛未来能越办越好。

最高人民法院信息中心副主任孙福辉在致辞中从司法业务角度论述了司法行业对人工智能技术的渴望，回顾了法院信息化建设历程以及在人工智能技术方面目前的积累，介绍了法院在人工智能技术方面未来的探索方向。

中国电子科技集团副书记胡爱民在致辞中从电科集团的技术布局和新一代人工智能发展战略的高度，肯定了本挑战赛的积极探索，并对比赛未来的发展报以美好祝愿。

随后，隆重举行颁奖仪式。来自哈尔滨工业大学、西安电子科技大学、山西大学等高校，国双科技、中国电科、阿里巴巴、民生银行、华宇信息、汉王数字、达观数据、富驰科技等高科技企业以及安徽省高院、浙江省高院等法院的队伍分别获得各个奖项(详细获奖名单见文末挑战赛官方网站)。

颁奖会上，中国司法大数据研究院胡振博士、清华大学涂存超博士还对本挑战赛的赛事设置、数据准备、参赛情况和主要技术进行了综述(相关综述论文见文末下载链接)。下午，各获奖队伍分别报告，开展技术交流。参赛选手们还就明年挑战赛命题进行了讨论。

挑战赛相关资源：

挑战赛官方网站：http://cail.cipsc.org.cn

挑战赛数据下载网站：https://github.com/thunlp/CAIL

挑战赛数据说明论文：https://arxiv.org/abs/1807.02478

挑战赛参赛情况综述：https://arxiv.org/abs/1810.05851