文章编号：1003-0077(2021)01-0001-08

基于 LSTM 的层次化篇章依存分析方法

贾延延[1,2]，程学旗[2]，冯 键[3]

(1. 中国再保险(集团)股份有限公司 博士后科研工作站，北京 100033；
2. 中国科学院 计算技术研究所，北京 100190；
3. 中国再保险(集团)股份有限公司 信息技术中心，北京 100033)

摘 要：在长距离依赖场景，篇章依存分析的效果欠佳，传统分析方法通常设计大量特征模板来缓解这一瓶颈问题。该文提出一种层次化篇章依存分析方法，减少了篇章分析器所需一次性处理的篇章分析单元的数量，从而缩短了分析器所处理的依存对之间的距离；并通过长短时记忆模型直接处理篇章分析单元中的序列信息，避免了特征提取。在 RST 语料库上进行实验，结果表明，即使在不提取任何特征的情况下，层次化篇章依存分析方法的分析效果依然优于同类深度学习模型在提取必要特征后的实验效果。

关键词：篇章；依存分析；LSTM

中图分类号：TP391 **文献标识码**：A

A Hierarchical Discourse Dependency Parsing Method with Long Short-Term Memory

JIA Yanyan[1,2], CHENG Xueqi[2], FENG Jian[3]

(1. Postdoctoral Research Workstation, China Reinsurance (Group) Corporation, Beijing 100033, China;
2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;
3. Information & Technology Center, China Reinsurance (Group) Corporation, Beijing 100033, China)

Abstract: The discourse parsing is challenged by the long distance dependency issue. In contrast to the traditional manual feature-engineering strategy, a hierarchical discourse dependency parsing method based on LSTM is proposed in this paper. It decreases the number of element discourse units that the parser had to address at one-time. The experimental results on RST Discourse Treebank show that the performance of the proposed method outperforms other deep learning methods combined with certain features.

Keywords: discourse; dependency parsing; LSTM

0 引言

篇章是由词、短语、句子和段落构成的自然语言单位，是一个有组织和层级的整体，可以表达完整的思想和意图。篇章具有连贯性(Coherence)、衔接性(Cohesion)、信息性(Informativity)、意图性(Intentionality)、情景性(Situationality)、可接受性(Acceptability)和跨篇章性(Intertextuality)等 7 种特性[1]。

基于修辞结构理论(Rhetorical Structure Theory, RST)[2]的篇章结构分析是篇章连贯性分析中的一个重要分支。在 RST 理论中，基本篇章分析单元(Element Discourse Unit, EDU)之间存在修辞关系。篇章成分分析通过这种修饰关系自底向上地合并分析单元，形成中间节点，直到建立包括整篇文章中所有 EDU 的篇章成分分析树。目前，绝大多数篇章分析工作都采用成分分析模式。几乎所有针对英语的篇章成分分析工作都基于经典的修辞结构理论篇章树库(RST DT)[3]。例如，Hernault 等[4]提出了基于支持向量机的篇章成分分析器 HILDA，他们采用二分类器进行结构分析，用多分类器预测修辞关系和核心附属属性，借助位置、长度、距离、句法分析结果、支配集等特征自底向上地建立成分分

收稿日期：2020-04-07 **定稿日期**：2020-07-06

基金项目：国家自然科学基金(91746301)

析树;Feng 等[5]为提升 HILDA 的分析效果,引入丰富的语言学特征。例如,规则、依存结构、语义相似度、支配节点、上下文信息等特征,构造了多达21 410个特征模板,通过互信息评价特征的贡献将其排序;Li 等[6]借助斯坦福自然语言处理工具获得句法树结构,利用递归神经网络获得 EDU 和中间分析单元的向量表示,再基于神经网络的分类器分别判断篇章分析树结构和修饰关系。但是,上述无论传统分析方法或是基于深度学习的篇章分析方法都无法避免人工特征提取。

虽然篇章成分分析较篇章依存分析[7]更受关注,但篇章依存分析的优势不容忽视。篇章成分分析通过引入中间节点的方式,缓解"长距离依赖"这一性能瓶颈问题。然而,篇章依存分析无需增加中间节点,就可以直接分析 EDU 之间的关系,水平建立分析树。因此篇章依存分析便于直接判断篇章中任意两个分析单元之间是否存在依存关系,其分析结果更为直观和便捷。典型的依存分析工作如 Li 等[7],选择基于图模型的 Eisner 算法和最大生成树算法进行篇章依存分析。首先,将 RST 篇章树库中的成分分析树转换为依存分析树。然后,结合词汇、词性、长度、位置、句法分析结果、语义信息等六类特征集进行实验,所生成的依存分析树不包含额外引入的中间节点。然而,虽然基于图模型的分析方法便于全局优化,且实验效果通常优于基于转移的篇章分析器。但是,用图模型进行分析的算法时间复杂度较高。更重要的是,基于图模型的分析方法依然无法克服篇章依存分析的两大难点与挑战问题:(1)在篇章依存分析中,长距离依赖场景的分析效果差;(2)为提高分析效果,引入大量人工特征来辅助判断。

在实际应用场景,只有减少和规避特征提取才能提高篇章分析器的易用性和鲁棒性,以避免人力浪费。若要缓解长距离依赖场景分析效果差这一瓶颈问题,单纯从特征设计和后处理技巧入手势必低效,应该考虑篇章分析基础框架和模式。

另一方面,分层次处理的篇章成分分析框架具有启发性。Joty 等[8]分别使用两个动态条件随机场建立句子内部的篇章成分分析树和句子之间的篇章成分分析树,选择 CKY 算法进行全局最优解码,并为句内分析和句间分析分别引入丰富且有差异性的特征集进行实验。Liu 等[9]同样分层次地进行句子内和句子间的篇章成分分析,分别用两个线性链条件随机场来建模篇章结构和关系。采用贪心策略自底向上的建立篇章成分分析树。他们利用长短时记忆模型(Long Short-Term Memory, LSTM)[10]来建模 EDU 和句子的特征,并在句间篇章分析场景引入更能体现结构化特征的递归神经网络来表达上下文信息。

上述两个层次化的篇章成分分析工作都取得了不错的实验效果。因此,本文给出了层次化的篇章依存分析方法。这种分析方法不再一次性分析篇章中的所有分析单元,而是分层次地进行篇章分析。首先,建立句子内以 EDU 为叶子节点的篇章分析子树;然后建立句子间以句子为叶子节点的篇章分析树。最后,整合两层分析结果,形成整篇文章的篇章依存分析树。分层次的方式可以避免一次性分析篇章中的所有 EDU,减少了篇章依存分析器所需面对长距离依赖对的数目,从而缓解了长距离依赖这一性能瓶颈问题。另一方面,该方式还带来了可以根据不同层次的特点、设计更有针对性的分析策略的好处。与此同时,本文选取改进的长短时记忆模型,结合注意力机制来获得分析单元的表示,避免特征提取。在 RST 篇章树库上进行实验,结果表明,本文基于 LSTM 的层次化篇章依存分析方法避免了耗时的特征设计,且实验效果超越了同类深度学习模型。

1 如何分层次建立篇章依存分析树

本文采用 Arc-eager[11]模式,给出了基于转移的篇章分析方法,具体转移方式在 2.2 节详述。本模型用 B、S 和 A 分别保存输入的篇章分析单元序列、转移过程中形成的子树表示和转移状态(转移动作和篇章关系)。在这种模型框架下,传统的篇章分析器将 EDU 一次性地输入到 B 中,再结合 S 和 A 的向量表示去预测当前的转移动作。由于篇章的篇幅长,这种传统的篇章分析方式无法回避依存分析中的长距离依赖造成的分析效果差的问题。

本文利用层次化的依存分析方法,为整个篇章建立一棵篇章分析树,其过程分为三个阶段。

(1) 句内层次篇章依存分析:针对每个句子,将句子中的 EDU 依次输入到 B 中,建立句子级别的、以 EDU 为叶子节点的篇章分析子树,如图 1 所示。

(2) 句间层次篇章依存分析:针对整个篇章,将句子作为一个篇章分析单元,将其向量表示依次输入 B 中,建立以句子为叶子节点的篇章分析树,如图 2 所示。

图 1　句内篇章依存分析示例

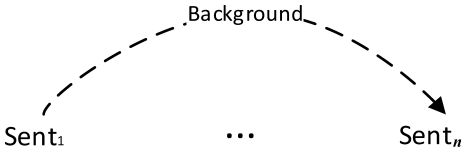

图 2　句间篇章依存分析示例

(3) 整合分析结果：用句内层所预测的句子级别的篇章分析子树的根节点标号代表句间层中的句子节点，整合两层的预测结果，得到整个篇章以 EDU 为叶子节点的篇章分析树，即最终篇章依存分析结果，如图 3 所示。

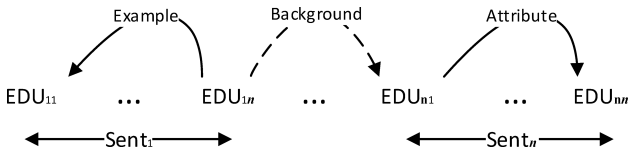

图 3　整合句内和句间篇章依存分析结果

2　基于 LSTM 的层次化篇章依存分析模型

2.1　长短时记忆模型

Sepp Hochreiter 于 1997 年设计了长短时记忆模型（LSTM）缓解了长期困扰循环神经网络（Recurrent Neural Network，RNN）[12-14] 的梯度消失或梯度爆炸问题。经典的 LSTM 包含输入门、输出门、遗忘门三种门控和一个记忆单元。长短时记忆模型具有多种改进形式。例如，双向 LSTM、树形 LSTM、多层 LSTM 等。本文选择双向 LSTM 来提供篇章分析单元的向量表示，具体表示方法在 2.4 节中详述。

2.2　基于转移的分析方法

本文采用 Arc-eager 模式的基于转移的篇章分析方法来生成篇章分析树。Arc-eager 转移模式改进了 Arc-standard[15] 转移方法的限制条件。树节点无需找到其所有子节点就可以连接其头节点。该分析方法包括 Shift、Left-Arc、Right-Arc、Reduce 等四种转移动作，动作转移过程如表 1 所示。以本文基于 RST 语料的篇章依存分析为例，表 1 中 B、S 和 A 分别用于保存输入的篇章分析单元序列、转移过程中形成的子树表示以及转移状态。x 和 y 表示 B 和 S 的头节点。Shift 操作将 B 的头节点转移到 S 的头元素位置；Reduce 操作将 S 中的头节点弹出。Left-Arc 根据所预测的依存关系在 S 和 B 的头节点之间建立依存弧，其中 B 的头节点为核心节点，S 的头节点为附属节点。动作执行后 S 中的头节点被弹出，将转移状态保存到 A 中。相应地，Right-Arc 根据所预测的依存关系在 S 和 B 的头节点之间建立依存弧，其中 S 的头节点为核心节点，B 的头节点为附属节点。动作执行后 B 中的头节点被推入 S 中，将转移状态保存到 A 中。

表 1　Arc-eager 模式分析方法转移状态

Arc-eager 分析动作	转移过程
Shift	$(S, x\|B, A)\rightarrow(S\|x, B, A)$
Left-Arc	$(S\|x, y\|B, A)\rightarrow(S, y\|B, A\cup\{(y\rightarrow x)\})$
Right-Arc	$(S\|x, y\|B, A)\rightarrow(S\|x\|y, B, A\cup\{(x\rightarrow y)\})$
Reduce	$(S\|x, B, A)\rightarrow(S, B, A)$

2.3　层次化篇章分析模型

2.3.1　模型结构

本文基于 LSTM 的 Arc-eager 模式篇章分析框架如图 4 所示。将输入的篇章分析单元依次存入 B 中。在初始状态下，使篇章中的第一个分析单元处于 B 的头元素位置，连接 B 中的前两个元素来获得输入序列 B 的向量表示；将分析过程中产生的中间子树结构存入 S 中，用 S 的头元素构造其向量表示；对于 S 和 B 而言，这里的“元素”在句内篇章分析层次为基本篇章分析单元，在句间篇章分析层次是指句子。A 用于存放篇章分析过程中产生的历史转移状态，包括转移动作和元素对之间的依存关系。连接 A 中的前三个转移状态的向量表示来构造模型的历史转移状态表示。本文句内层次的篇章依存分析和句间层次的篇章依存分析都依照此模型结构进行实验，句内和句间层次的篇章分析的输入信息有所不同，将在 2.4 节中详述。图 4 中，SH 代表转移动作为 Shift，RA（Li）表示转移动作为 Right-Arc，依存关系为 List。

将 S、B、A 三部分的向量表示连接起来，经过

一个 ReLU 变换和两个用 ReLU 作为激活函数的全连接层处理后，得到 p_t，即 t 时刻的篇章分析状态。将 p_t 进行仿射变换后，输入到 softmax 多分类器中，预测各个转移状态的概率，取概率最大的转移状态为当前时刻的模型预测结果。本实验采用贪心策略进行解码，交叉熵作为损失函数。

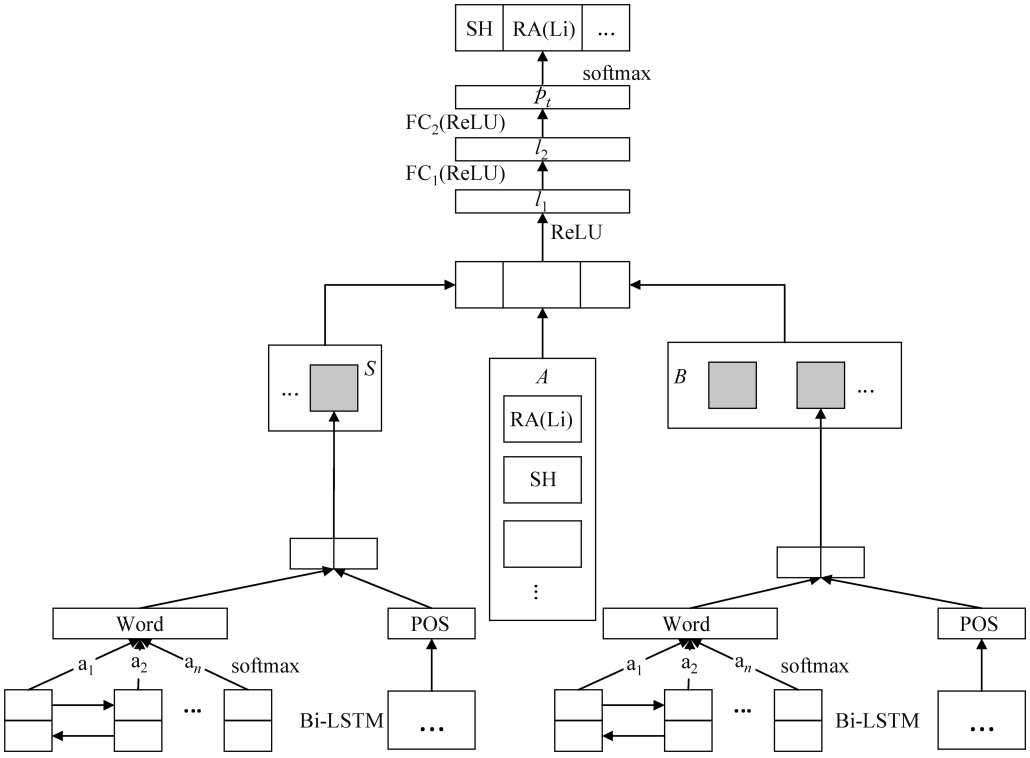

图 4 篇章依存分析模型结构

2.3.2 模型分析过程

本节以 RST 语料库中的篇章 wsj_0609 为例，来说明本模型的篇章依存分析过程（这里以不分层次的传统分析方法为例，即一次性处理篇章中的所有 EDU，直接得到整篇文章以 EDU 为叶子节点的分析树）。该篇章包含 185 个 EDU。这里给出其中前 4 个 EDU 所构成片段（[*President Bush insists*]$_{E_1}$ [*it would be a great tool*]$_{E_2}$ [*for curbing budget deficit*]$_{E_3}$ [*and slicing the lard out of government programs.*]$_{E_4}$）的依存分析过程。表 2 列出了执行完每一个转移状态后，A、S 和 B 中的内容和状态更新。状态 0 代表篇章分析的初始状态，此时 A 和 S 为空，B 中存放了所有输入 EDU，从第一个 EDU 开始顺序分析。根据当前 S、B 和 A 的状态表示，预测转移状态并存入 A 中，即更新 A 的状态。根据转移状态执行相应动作，并建立依存关系（Arc-eager

表 2 模型状态转移过程

篇章	状态	A	S	B
E_1: *President Bush insists* E_2: *it would be a great tool* E_3: *for curbing budget deficit* E_4: *and slicing the lard out of government programs.* E_5: *He wants it now.* ...	0	[]	[]	[E_1, E_2, E_3, E_4, E_5, …]
	1	[**SH**]	[E_1]	[E_2, E_3, E_4, …]
	2	[**LA(Attribution)**, SH]	[]	[E_2, E_3, E_4, …]
	3	[**SH**, LA(Attribution), SH]	[E_2]	[E_3, E_4, …]
	4	[**RA(Elaboration)**, SH, LA(Attribution), SH]	[E_2, E_3]	[E_4, …]
	5	[**RA (joint)**, RA (Elaboration), LA(Attribution), SH]	[E_2, E_3, E_4]	[…]
	…	…	…	…

模式)，从而更新 S 和 B 中的内容；更新后的 S、B 和 A 构成了下一次预测的状态表示基础。直到 B 为空，A 包含了分析整篇文章的所有转移状态，S 中即为整篇文章的篇章依存分析树。A 中粗体转移状态即为根据前一状态的向量表示所预测的转移。

根据表 2 中的状态转移过程，篇章中的前 4 个 EDU 可以构成图 5 中的篇章依存分析子树。在此基础上，通过继续进行转移预测和状态更新得到整个篇章的依存分析树。

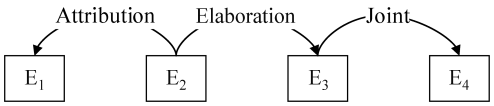

图 5　篇章片段的依存分析子树结构

2.4　不同层次的篇章分析单元表示方法

如图 4 所示，采用双向长短时记忆模型结合注意力机制来表示 B 和 S 中的篇章分析单元。篇章分析单元在句内层为 EDU，在句间层为句子。

具体来说，本文将篇章分析单元中的单词序列输入到双向 LSTM 中，使用注意力机制去捕捉词序列中的重点单词。将双向 LSTM 的顺序和逆序输出连接起来，构成篇章分析单元的词汇信息表示。采用 GloVe 词向量[16]初始化篇章分析单元中的单词的向量表示。本文通过斯坦福自然语言处理工具(Stanford CoreNLP Toolkit)[17]来获取篇章分析单元中单词的词性信息。与词汇信息的建模方式类似，本文同样采用双向长短时记忆模型结合注意力机制来获得篇章分析单元的词性信息表示。由于建模词汇信息和词性信息的网络结构相同，图 4 中省略了建模词性信息的网络结构。最后，将篇章分析单元的词汇信息和词性信息的向量表示连接起来构成了 S 和 B 中的篇章分析单元的向量表示。

3　实验与分析

3.1　实验语料

本文采用 RST 篇章树库进行实验，RST 语料库包含 385 篇来自《华尔街日报》的新闻报道，包括超过 176 000 个单词。最长的篇章包括 2 124 个单词，平均每篇文章包含 458.14 个单词，56.59 个 EDU。平均每个 EDU 包含 8.1 个单词[3]。虽然 RST 篇章树库所包含的篇章数目不多，但是语料库中的篇章篇幅较长；并且包括财务报告、故事、商业新闻、文化评论和社论等多种题材，篇章结构关系丰富且复杂。因此，几乎所有针对英文的篇章成分分析和篇章依存分析工作都选用 RST 篇章树库进行实验。这也带来了实验结果公平、易于对比的优点。

RST 篇章树库建立在修辞结构理论框架下，首先将篇章切分为基本篇章分析单元，然后通过修辞结构来标注 EDU 之间的结构和修饰关系，并按照 EDU 的作用和重要性将其分为核心(Nucleus)和附属(Satellite)两种成分。其中表达中心思想和主要信息的 EDU 作为核心，起到补充说明和修饰作用的 EDU 作为附属。本文选择 Li 等[7]的方式，将 RST 语料库中的成分分析树转换为依存分析树，同样选取其中 380 篇文章进行实验，包括训练集 312 篇，验证集 30 篇，测试集 38 篇。同时本文选取 RST 篇章树库中的 111 个细粒度关系进行实验。

3.2　评价指标

无标记正确率(Unlabeled Attachment Score，UAS)[18-19]和有标记正确率(Labeled Attachment Score，LAS)[20]是句法依存分析和篇章依存分析工作普遍采用的评测指标，便于比较各种同类工作的实验效果。本文即采用 UAS 和 LAS 作为篇章依存分析的评测标准。以 RST 篇章树库为例，无标记正确率是指测试集中找到正确的支配节点的 EDU 数目占该篇章中总 EDU 数的比例；有标记正确率是指测试集中找到正确的支配节点，并且 EDU 对之间的修辞关系也预测正确的 EDU 数占该篇章中总 EDU 数的比例。其中，支配节点指在修辞关系中占据核心和主导地位的节点即核心节点；相应地，附属节点指在修辞关系中充当附属成分的节点。

3.3　基线方法

本文将层次化的篇章分析模型和表 3 中的几种基线方法进行对比。①Basic[21]：该方法同样为基于转移的篇章依存分析器，使用深度学习模型(LSTM)获得篇章分析单元的向量表示；但是，为达到较好的实验效果，该工作引入多种位置信息来获得篇章分析单元的向量表示，并且采用一次性处理文章中所有基本分析单元的方式进行篇章依存分析。②Hierarchical parser(no feature)：本文层次化的篇章分析法，在句内和句间的篇章分析过程中都不引入任何特征和位置信息，采用 2.4 节介绍的篇章分析单元表示法来建模 EDU 或句子；③Re-

fined[21]：在Basic方法的基础上，为缓解长距离依赖的篇章分析单元对间的结构和修饰关系难以捕捉的问题，该方法设计了一种记忆网络，自动地捕获篇章分析单元间的衔接性和话题线索，从而提高篇章依存分析效果。④Hierarchical parser：本文层次化的篇章分析法。为发挥层次化的依存分析方法根据不同层次建模的优势，在2.4节的篇章分析单元表示方法基础上，在句间分析层次，引入待分析的句子对是否在同一段内的信息来反应篇章结构特点。⑤MST-full[7]：该方法是目前效果最好的基于图模型的篇章依存分析器。

表3 篇章依存分析效果对比

ID	方法	UAS	LAS
1	Basic(word+POS+position)[21]	0.593 3	0.383 2
2	**Hierarchical parser(no feature)**	**0.607 0**	**0.385 3**
3	Refined[21]	0.619 7	0.394 7
4	**Hierarchical parser**	**0.625 7**	**0.399 0**
5	MST-full[7]	0.733 1	0.430 9

3.4 实验结果分析

本文在表3中列出了篇章依存分析结果。通过比较可以发现，使用LSTM获取篇章分析单元的向量表示的Basic方法依然无法避免各种特征提取。采用本文层次化的篇章分析方法(Hierarchical parser(no feature))，即使在不引入任何手工或外部工具提取的特征的前提下，实验效果在UAS和LAS上都高于Basic方法。这说明通过层次化的方式减少篇章分析器所需处理长距离依赖的数目，确实能够提升篇章分析效果。但是，和Refined方法相比，Hierarchical parser(no feature)效果稍逊。主要原因是Refined方法不仅需要抽取多种特征，而且该方法设计了一个记忆网络，将篇章中在向量空间上相似的篇章分析单元聚类到相同的记忆槽中，再将记忆槽的向量表示加入到篇章分析单元的向量表示中。这样，为每一个篇章分析单元标记了其话题线索，这种话题线索反应了篇章的结构信息和分析单元对间的依存关系。为此，在Hierarchical parser中，在句间层次，本文引入待分析的篇章分析单元对(句子对)是否在同一段内的简单位置信息来反应篇章中浅层的结构信息。虽然加入段落信息的方式比使用记忆槽捕捉话题线索的方式简单粗略，但是，Hierarchical parser的篇章依存分析效果依然在UAS和LAS上都超过了Refined分析方法。并且，Hierarchical parser只在句间层次引入句子对是否在同一段这一种位置信息来标记篇章浅层结构，并没有引入任何其他特征；而Refined方法中运用了多种不同特征，例如，用EDU在句子内、段落内和文章中的位置来表示篇章分析单元；还引入了EDU之间是否在一句内、是否在一段内、以及距离信息来表示EDU对之间的位置关系。Hierarchical parser所引入的结构信息远少于Refined方法。可见，层次化的篇章依存分析模式本身较传统的整篇文章一次性处理完成的篇章依存分析模式更有优势。

由于现存的篇章依存分析工作较少，依存分析树又不能一一对应的转换为成分分析树，因此本模型难以和其他篇章成分分析工作公平的对比实验结果。本实验采取同样的实验设置和目前效果最好的篇章依存分析实验MST-full进行对比，虽然效果还有差距，但是MST-full运用了6个复杂特征集，包括词汇、词性、长度信息、位置信息、语义相似度特征、句法分析结果等。其中语义相似度和句法分析结果等特征需要引入外部资源和工具才能获得；另外，MST-full是基于图模型的篇章分析方法，不需要按照某个顺序去判断篇章分析单元之间的结构关系，可以搜索全局最优解。但图模型的篇章分析方法($O(n^3)$)具有比本文基于转移的分析法($O(n)$)更高的时间复杂度。

为更好地说明分层次的篇章依存分析模型在不同细粒度关系上的分析效果，本文对表3中的Hierarchical parser(ID为4)的实验结果进行细化，在表4中给出语料中数量最多的前8种细粒度关系和两种数量较少的典型关系(example和background)的UAS和LAS分析结果，并标记了这些关系在语料库和测试集中出现的次数。可以发现，除了elaboration-additional和List两种关系之外，语料库中数量较多的关系，由于训练数据丰富，实验效果通常更好。关系elaborate-additional在语料库中的总数量较多，但分析效果不理想的主要原因是：elaborate-additional(此关系表示附属成分是核心成分的细化或附加详尽说明)在关系含义上和elaboration-additional-e(当附属成分是嵌套结构elaborate-additional变为elaboration-additional-e)以及elaboration-object-attribute-e(不同于elaboration-additional-e之处在于附属成分是其所修饰的核心成分的本质属性)十分相似，容易混淆。并且elaborate-additional在句内和句间层次的分布不均匀，句

内层次分布较少。与其相似的 elaboration-object-attribute-e 在句内篇章分析层次出现了超过 2 000 次，导致篇章分析器因为“从众”倾向，做出误判。List 关系通常标识并列语义或者结构，不同于其他关系，List 关系的跨度通常较长，因此判断难度更大。

表 4　不同细粒度关系的分析效果

关系名称	总关系数量	测试集关系数量	UAS	LAS
elaboration-additional	3 224	312	0.419 9	0.294 9
attribution	2 803	329	0.814 6	0.805 5
elaboration-object-attribute-e	2 524	250	0.900 0	0.704 0
List	1 896	206	0.388 3	0.276 7
Same-Unit	1 357	127	0.811 0	0.732 3
elaboration-additional-e	816	69	0.942 0	0.652 2
circumstance	625	80	0.537 5	0.287 5
explanation-argumentative	594	70	0.542 9	0.042 9
example	275	34	0.323 5	0.088 2
background	220	26	0.230 8	0

4　结束语

本文提出了一种层次化的篇章依存分析方法，该方法通过长短时记忆模型处理篇章分析单元中的序列信息，获得篇章分析单元的向量表示，避免了特征提取。在 RST 篇章树库上进行实验，结果表明，层次化的篇章依存分析方法的实验效果超过了不分层次、但提取了必要特征的同类深度学习模型。这说明分层次建立依存分析树的方式，通过减少篇章分析器所需处理长距离依赖对的数量，缓解了长距离依赖分析效果差这一依存分析的性能瓶颈问题。实验效果证明，这种层次化的篇章依存分析框架是一种提高篇章依存分析性能的有效途径。

参考文献

[1] Beaugrande R D, Dressler W. Introduction to text linguistics[M]. London: Longman, 1981.

[2] Mann W C, Thompson S A. Rhetorical structure theory: Toward a functional theory of text organization[J]. Text-Interdisciplinary Journal for the Study of Discourse, 1988, 8(3): 243-281.

[3] Carlson L, Marcu D, Okurovsky M E. Building a discourse-tagged corpus in the framework of rhetorical structure theory [C]//Proceedings of the 2nd Annual Meeting of the Special Interest Group on Discourse and Dialogue. Aalborg, Denmark: Association for Computational Linguistics, 2001.

[4] Hernault H, Prendinger H, Duverle D A, et al. HILDA: A discourse parser using support vector machine classification[J]. Dialogue and Discourse, 2010, 1(3): 1-33.

[5] Feng V W, Hirst G. Text-level discourse parsing with rich linguistic features [C]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Jeju, Republic of Korea: Association for Computational Linguistics, 2012: 60-68.

[6] Li J W, Li R M, Hovy E H. Recursive deep models for discourse parsing [C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 2061-2069.

[7] Li S J, Wang L, Cao Z Q, et al. Text-level discourse dependency parsing[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, Maryland, USA: Association for Computational Linguistics, 2014: 25-35.

[8] Joty S, Carenini G, Ng R, et al. Combining intra- and multi-sentential rhetorical parsing for document-level discourse analysis[C]//Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia, Bulgaria: Association for Computational Linguistics, 2013: 486-496.

[9] Liu Y, Lapata M. Learning contextually informed representations for linear-time discourse parsing [C]// Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, 2017: 1300-1309.

[10] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[11] Abney S P, Johnson M. Memory requirements and local ambiguities of parsing strategies[J]. Journal of Psycholinguistic Research, 1991, 20(3): 233-250.

[12] Hopfield J J. Neural networks and physical systems with emergent collective computational abilities[J]. Proceedings of the National Academy of Sciences, 1982, 79(8): 2554-2558.

[13] Jordan M I. Serial order: A parallel distributed processing approach[R]. San Diego: University of California, Institute for Cognitive Science, 1986.

[14] Elman J L. Finding structure in time[J]. Cognitive Science, 1990, 14(2): 179-211.

[15] Nivre J. An efficient algorithm for projective dependency parsing[C]//Proceedings of the 8th International Workshop on Parsing Technologies. Nancy, France: IWPT03, 2003: 149-160.

[16] Pennington J, Socher R, Manning C D. GloVe: Global vectors for word representation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar: Association for Computational Linguistics, 2014: 1532-1543.

[17] Manning C D, Surdeanu M, Bauer J, et al. The Stanford CoreNLP natural language processing toolkit [C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, Maryland, USA: Association for Computational Linguistics, 2014: 55-60.

[18] Collins M, Singer Y. Unsupervised models for named entity classification [C]//Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. Maryland, USA, 1999: 100-111.

[19] Eisner J. An empirical comparison of probability models for dependency grammar[R]. USA: Institute for Research in Cognitive Science, University of Pennsylvania, 1996.

[20] Nivre J, Hall J, Nilsson J. Memory-based dependency parsing[C]//Proceedings of the 8th Conference on Computational Natural Language Learning. Boston, MA, USA: CoNLL, 2004: 49-56.

[21] Jia Y Y, Ye Y, Feng Y S, et al. Modeling discourse cohesion for discourse parsing via memory network [C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Australia, 2018: 438-443.

贾延延(1983—),通信作者,博士,主要研究领域为自然语言处理、篇章分析、智慧金融。
E-mail: 516752364@qq.com

冯键(1972—),博士,总经理,主要研究领域为多方安全计算、区块链技术。
E-mail: fengj@chinare.com.cn

程学旗(1971—),博士,研究员,主要研究领域为网络科学与社会计算、机器学习与智能搜索、智能金融、网络信息安全。
E-mail: cxq@ict.ac.cn