文章编号：1003-0077（2011）00-0000-00

采用 Stack-Tree LSTM 的汉语一体化依存分析模型*

刘航，刘明童，张玉洁，徐金安，陈钰枫

（北京交通大学，计算机与信息技术学院 北京 100044）

摘要：在汉语一体化依存分析中，如何利用分词、词性标注和句法分析的中间结果作为分析特征成为核心问题，也是三个任务相互制约协调、共同提高性能的关键所在。目前无论基于特征工程的方法还是基于深度学习的方法尚无法充分利用分析过程中依存子树的完整信息，而依存子树作为中间结果的主要成分对三个任务的后续分析具有重要的指导意义。为解决该问题，本文在基于转移的依存分析框架下，提出 Stack-Tree LSTM 依存子树编码方法，通过对分析栈中所有依存子树的有效建模，获取任意时刻的依存子树的完整信息作为特征参与转移动作决策。我们利用该编码方式提出词性特征使用方法，融合 N-gram 特征构建汉语一体化依存分析神经网络模型。最后在宾州汉语树库上进行了验证实验，并与已有方法进行了比较。实验结果显示本文提出的模型在分词、词性标注和依存分析任务上的性能非常接近特征工程最好的结果，并且均超过已有的一体化依存分析神经网络模型。

关键词：中文分词、词性标注和依存分析；依存子树；神经网络

中图分类号：TP391　　**文献标识码：**A

Improved Character-Based Chinese Dependency Parsing by Using Stack-Tree LSTM

LIU Hang, LIU Mingtong, ZHANG Yujie, XU Jinan, CHEN Yufeng

(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract: In the character-based Chinese dependency parsing, it is crucial how to use the intermediate results of word segmentation, POS tagging and dependency parsing as the features, which are to be exploited by each of the three tasks and are expected to contribute to the performance improvement. However, the state-of-the-art methods for character-based Chinese dependency parsing did not fully utilize the dependency subtree information built in the stack. In order to solve this problem, this paper proposes a novel Stack-Tree LSTM to capture dependency subtree information for any time. On this basis, we construct a character-based neural network joint model by integrating subtree feature and POS feature in addition to N-gram feature. We conduct experiments on Penn Chinese Treebank 5 and compare the results with other models. The results show that our model approached to the best results of the feature engineering joint models and outperformed the state-of-the-art neural joint models in Chinese word segmentation, POS tagging and dependency parsing.

Key words: Chinese word segmentation, POS tagging and dependency parsing；dependency subtree；neural network

1 引言

分词、词性标注和依存句法分析三项任务是汉语自然语言处理的基础技术，其性能的提高对于机器翻译等其他任务的实际应用至关重要。将三项任务融合到一个模型中并行处理的一体化依存分析模型[1-4] 解决了串行处理方式中各任务间的错误传播问题；同时任务的中间

* **收稿日期：**　　**定稿日期：**

基金项目：国家自然科学基金（61370130,61473294），中央高校基本科研业务费专项资金(2015JBM033)，科学技术部国际科技合作计划（K11F100010）

结果可以相互利用，使得三项任务相互协调以提升性能。因此，如何充分利用中间结果作为特征成为一体化依存分析模型性能提升的关键问题。

基于特征工程的一体化依存分析模型[1-3]，总结了大量有效的特征模板获取中间结果信息作为特征，在三项任务上取得了很好的性能。基于神经网络的一体化依存分析模型[4]，利用神经网络自动学习有效特征，仅结合少量特征模板就达到了接近特征工程模型的性能。但是，目前无论哪种模型在利用中间结果时都未能充分考虑已经形成的依存子树信息，而分析栈中保存的依存子树作为中间结果的主要成分，对三项任务的后续分析有着重要的指导意义。

针对该问题，本文提出 Stack-Tree LSTM 依存子树编码方法获取分析栈中所有依存子树的完整信息。具体的，我们采用 Tree LSTM 对每棵依存子树信息进行编码，并利用 Stack LSTM 将分析栈中的所有依存子树的编码结果累积到栈顶，作为特征参与随后的动作转移决策。利用该编码机制，我们提出词性特征的使用方法，在依存子树编码中融入词性信息。以此为基础，搭建汉语一体化依存分析神经网络模型，并采用双向 LSTM 获取句子的 N-gram 特征和上下文信息。本文在 CTB5 数据上进行评测并与已有模型进行比较。评测结果显示，本文模型在分词、词性标注和依存句法分析任务上的 F1 值分别达到了 97.78%，93.51%和 79.66%，均优于已有基于神经网络的一体化依存分析模型。

本文组织结构如下：第 2 节介绍一体化依存分析模型以及依存子树编码的相关工作；第 3 节详细描述本文提出的 Stack-Tree LSTM 依存子树编码方法；第 4 节详细描述基于 Stack-Tree LSTM 依存子树编码的一体化依存分析神经网络模型；第 5 节介绍评测实验与对比结果分析；第 6 节对本文工作进行总结。

2 相关工作

基于转移的汉语一体化依存分析模型最早由 Hatori[1]提出，通过以汉字为处理单位将分词任务设定为字的归约操作，并在字的归约操作时附加词性信息实现词性标注任务，由此，实现三项任务的并行处理。在一体化依存分析模型中，三项任务各自形成的中间结果保存在分析栈中，包括完成分词的词语、词性、依存子树和未成词的字串，这些信息构成了三项任务相互制约协调的特征。如何利用这些特征帮助转移动作决策成为之后研究的主要课题。Hatori[1]利用分析栈顶的三个元素抽取特征，对于依存子树只抽取根节点、最左最右孩子等部分节点，而不考虑其它节点。Zhang[2]在 Hatori[1]工作的基础上，标注词内汉字之间的依存关系用于字的归约操作，增加词内的依存关系信息作为特征，在依存分析任务上取得了目前最好的结果。

Kurita[4]首次将神经网络方法运用到一体化依存分析模型中，利用之前工作总结的特征模板抽取分析栈的信息，同样利用栈顶三个元素，只抽取依存子树根节点、最左最右孩子节点等部分节点信息，通过分布式方法表示输入到神经网络多层感知机进行转移动作决策，在分词和词性标注任务上取得了目前最好的结果。同时 Kurita[4]采用双向的长短时记忆神经网络(Long Short Term Memory, LSTM)对句子 N-gram 信息进行编码自动学习特征，该模型也仅利用了分析栈顶三个元素中汉字的 N-gram 信息，其性能接近特征工程的最好结果。

为了利用依存子树所有节点的信息，Socher[5]等人提出组合向量文法自上而下编码整棵依存子树。Tai[6]等人提出 Tree LSTM 神经网络模型，根据节点的分支个数调整神经单元中忘记门的数量，实现对依存子树信息编码。但是，这两个模型仅适用获得句子整体依存结构的情况。之后，Bowman[7]等人首次在句法分析过程中使用 Tree LSTM 编码子树，根据子树的构建特点将神经单元忘记门的数量设置为 2，使得 Tree LSTM 每次结合两个节点；编码结果保存在分析栈中，当与其他子树的根节点建立句法关系时，用于计算新的子树编码表示，以此得到子树的完整信息，并在句法分析和句子理解的联合任务上取得了较好的结果。但是 Bowman[7]的工作无法利用分析栈中同时存在的多棵子树信息。为了解决该问题，我们受

Dyer[8]工作的启发，利用 Stack LSTM 可以获取分析栈中所有元素信息的优势，提出了 Stack-Tree LSTM 依存子树编码方法，用于一体化依存分析。

已有的一体化依存分析神经网络模型忽略了词性标注的中间结果，而词性作为特征对于分词和依存分析有很大帮助。为了利用词性信息，我们提出了一种词性特征的利用方法，将词性信息融入到词向量中，参与转移动作决策。

3 Stack-Tree LSTM 依存子树编码方法

本文结合 Stack LSTM 和 Tree LSTM 的优势，提出 Stack-Tree LSTM 神经网络模型，获取分析过程中形成的依存子树信息。利用 Stack LSTM[8]的结构优势，将分析栈中所有元素信息汇总到栈顶；采用 Tree LSTM 获取依存子树的信息。主要思想是在一体化分析过程中，如果分析栈中某一元素是一棵依存子树，则使用 Tree LSTM 编码该子树；如果某一元素是词语或者未成词字串，则使用 LSTM 单元编码该词语或者字串。其中 Tree LSTM 神经单元按照依存关系对头节点和依存节点进行处理，获的依存子树的编码结果；而 Stack LSTM 结构可以将多棵依存子树的信息汇总到栈顶，从而增强模型对依存子树信息的获取能力。除此之外，本文使用的 Tree LSTM 神经单元与 LSTM 神经单元具有相似的门结构，通过门结构的控制作用避免了以往依存子树编码方案中的梯度消失问题[8]。

基于转移的分词、词性标注和依存句法分析的一体化依存分析模型可视为一系列转移动作的决策问题。转移模型由分析栈 S={ •••，S_1，S_0}和待处理队列 Q={Q_0，Q_1，••• }构成，分别记录包含依存子树的中间结果和待分析的字序列。初始状态下 S 为空，Q 为句子的所有字符；每次执行一个转移动作，并从当前状态更新至下一个状态；终止状态下 S 为一棵完整的依存树，其中包含了分词和词性标注结果，Q 为空。本文采用以下 4 种转移动作[1]：

(1)SH（k）（移进）：将 Q 中的首元素作为词的开始字符移进 S，并赋予词性 k 作为该字符所在词的词性；

(2)AP（拼接）：将 Q 中的首元素拼接到 S 的顶部元素中，该动作执行的前提条件为前一步动作是 SH(k)或者 AP；

(3)RL（左归约）：S 顶部的两个元素 S_0 和 S_1 出栈，构建依存关系 $S_1 \leftarrow S_0$，将根节点 S_0 入栈，S_1、S_0 是依存子树或者词；

(4)RR（右归约）：S 顶部的两个元素 S_0 和 S_1 出栈，构建依存关系 $S_1 \rightarrow S_0$，将根节点 S_1 入栈，S_1、S_0 是依存子树或者词。

其中 SH（k）动作和 AP 动作完成分词和词性标注任务，通过 SH（k）确定每个词的开始边界及词性；RR 动作和 RL 动作完成依存分析任务，构建一棵完整的依存树。

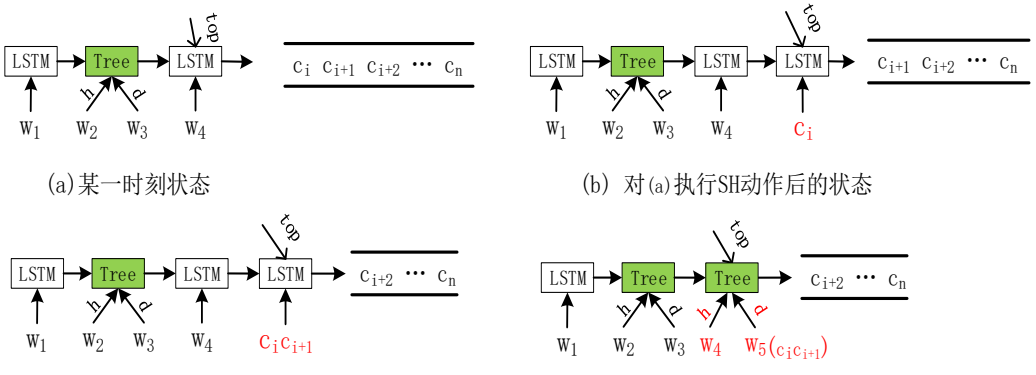


图 1 Stack-Tree LSTM 依存树编码方法示意图

采用上述转移动作，我们提出的 Stack-Tree LSTM 依存子树编码方法的结构如图 1 所示。图 1（a）显示了某一时刻分析栈 S 和待处理队列 Q 的状态，分析栈 S 中包含三个元素，其

中词 w_1 和 w_4 以 LSTM 神经单元编码、子树 $w_2 \rightarrow w_3$ 以 Tree LSTM 神经单元编码；从栈底到栈顶，神经单元之间进行信息的传输，栈顶元素神经单元的输出向量融合了栈中所有元素的信息，包括词、未成词的字串、以及依存子树的信息，本文称栈顶输出向量为 STL 向量。随后依次执行 SH、AP 和 RR 之后的模型状态如图 1（b）、1（c）和 1（d）所示。

图 1(b)显示执行 SH（k）时的状态，待分析队列首元素 c_i 被移入分析栈顶。因为 c_i 是词的开始字符，所以采用 LSTM 神经单元对字符 c_i 的字向量和当前的 STL 向量进行编码得到新的 STL 向量。图 1 （c）显示执行 AP 时的状态，移入栈中的 c_{i+1} 与栈顶 c_i 构成字串 c_ic_{i+1}。因为还未成词，所以需要将栈顶元素的所有字符作为字串处理。LSTM 对未成词 c_ic_{i+1} 的字串向量和当前的 STL 向量进行编码得到新的 STL 向量。

图 1（d）显示执行归约动作时的状态，下面详细介绍用 Tree LSTM 神经单元对形成的依存子树进行编码以及用 Stack-Tree LSTM 对栈中所有元素信息编码的原理。执行 RR 时，首先栈顶的字串 c_ic_{i+1} 出栈，成词为 w_5；随后 w_4 出栈，构建依存关系 $w_4 \rightarrow w_5$，得到新的依存子树。Tree LSTM 神经单元对依存子树的头结点 w_4 和依存结点 w_5 的向量以及当前的 STL 向量进行编码，得到新的 STL 向量，其中包含了新形成的依存子树的信息。最后，以 w_4 为根节点的依存子树作为新的元素被移入分析栈顶。本文采用的 Tree LSTM 神经单元在时刻 t 的计算方法如公式（1）、（2）和（3）所示。

$$\begin{bmatrix} i_t \\ f_{head} \\ f_{dep} \\ o_t \\ \tilde{c}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(W \begin{bmatrix} h_{head} \\ h_{dep} \\ STL \end{bmatrix} + b \right) \tag{1}$$

$$c_t = f_{head} * c_{head} + f_{dep} * c_{dep} + i_t * \tilde{c}_t \tag{2}$$

$$h_t = o_t * \tanh(c_t) \tag{3}$$

其中，i_t 是神经单元的输入门，f_{head}，f_{dep} 分别是头结点和依存结点对应的忘记门，o_t 是输出门，c_t 是状态值，h_t 是隐藏层输出；W 是神经单元中的参数矩阵，b 是偏置项；σ 是 sigmoid 激活函数，* 是点积运算。

在一体化依存分析过程中，句子的依存关系随着转移动作的预测、操作，逐步形成；分析栈中保存了多棵依存子树。本文提出的 Stack-Tree LSTM 依存子树编码方法在执行归约动作时，对形成的新依存子树进行编码，并将分析栈中其余的依存子树信息输入到当前神经单元中，得到了当前时刻下包含所有依存子树信息的 STL 向量。已有的 Tree LSTM 依存树编码方法是在单棵依存子树上进行编码，无法将多棵依存树信息融合在一起，因此无法充分利用分析栈中的中间结果信息参与后续的转移动作决策，而本文的方法针对这一个问题给出了有效的解决方案。

4 一体化依存分析神经网络模型

利用本文提出的 Stack-Tree LSTM 依存子树编码方法构建汉语一体化分析神经网络模型，如图 2 所示，主要由 Stack-Tree LSTM 依存子树编码层、N-gram 特征层和多层感知机决策层三部分组成。本节主要介绍后面两个组成部分。

4．1 字向量、词向量和字串向量

本文通过预训练的方式获取最基础的字和词的向量表示。我们首先对生语料进行字符分割和分词处理，然后通过外部语言模型学习字向量和词向量，并将字、词向量的维度设置一致，最后利用预训练的字向量和词向量计算 N-gram 字串的向量表示。对于一个包含 n 个字

符 C_i^{i+n} 的字串 g，首先在预训练的向量中查询 g，如果 g 存在则使用对应的预训练向量；否则以字符为最小计算单元，按照公式（4）获得字串 g 的向量。

$$e_g = \frac{1}{n}\sum_i^{i+n} e_{c_i} \tag{4}$$

本文采用字向量加和求平均的方式获得字串向量，保留了字串中所有字符的信息，更具体准确地表示字串的语义。这种计算方法改进了已有研究中集外词的处理方式，在一定程度上避免了仅仅采用一个特殊标记，如 *oov* 或 *unk*，代替集外词所造成的信息损失。

在依存子树编码层，我们首先按照上述字串向量的计算方法得到依存子树中节点的词向量 w_g，然后将该词的词性标注信息 w_{pos} 加入其向量中，具体计算方式如公式（5）所示。

$$w = \max\{0, W_{word}[w_g; w_{pos}] + b_{word}\} \tag{5}$$

其中 W_{word} 是参数矩阵，b_{word} 是偏置项。

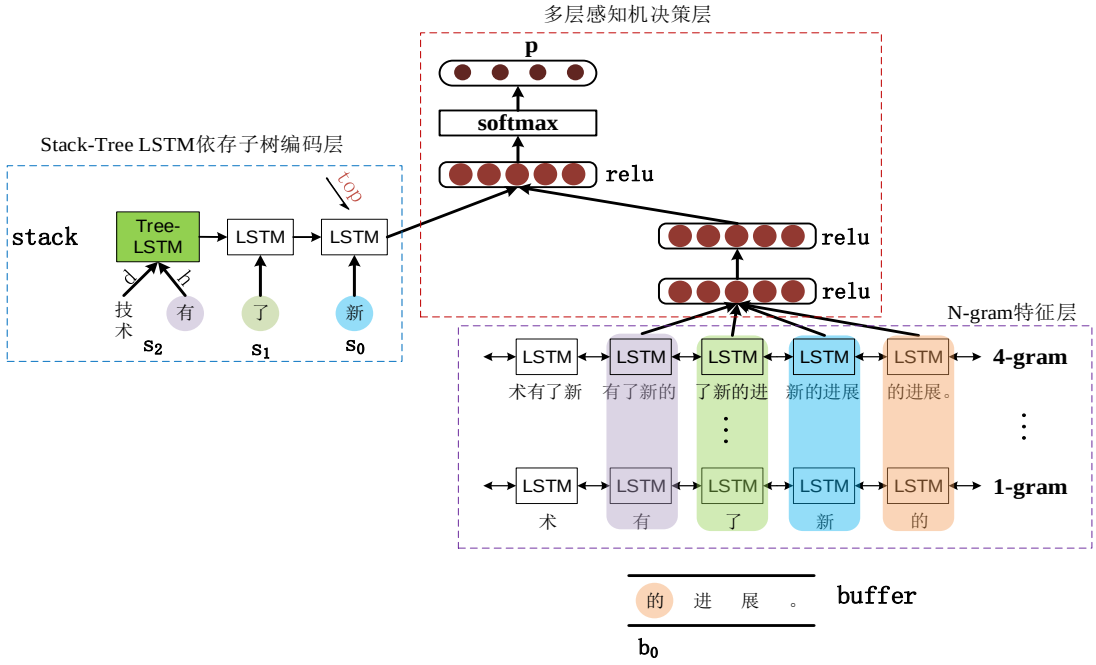


图 2 一体化依存分析神经网络模型

4．2 N-gram 特征层

字符的 N-gram 特征提供上下文信息，可以有效帮助一体化分析[4]。本文采用 Uni-gram、Bi-gram、Tri-gram、Four-gram 四种特征。在一体化依存分析神经网络模型中，我们按照以下方法对 N-gram 特征进行编码，构建 N-gram（N=4）特征层。对于给定句子 C_1^n，计算每种特征中所有字串的向量，分别得到四个向量矩阵矩阵 Uni-gram$_{1:n}$、Bi-gram$_{1:n}$、Tri-gram$_{1:n}$、Four-gram$_{1:n}$，每个向量矩阵代表一种 N-gram 特征的分布式表示。本文使用双向 LSTMs 对每种 N-gram 特征的分布式表示进行编码，以获取句子的全局信息。

对双向 LSTMs 编码后的四种 N-gram 特征，我们按照公式（6）融合四种特征得到每个字符的 N-gram 特征向量 v_i，然后用公式(7)获取一体化分析模型在当前分析状态下的 N-gram 特征层状态表示，由分析栈顶三个元素和待处理队列首元素所对应的 N-gram 特征向量组成。

$$v_i = biLSTM_{uni}(i) \circ biLSTM_{bi}(i) \circ biLSTM_{tri}(i) \circ biLSTM_{four}(i) \tag{6}$$

$$\phi(c) = v_{s_2} \circ v_{s_1} \circ v_{s_0} \circ v_{b_0} \quad (7)$$

其中 i 是字符在句中的位置下标($1 \leqslant i \leqslant n$)；$s_2$、$s_1$、$s_0$ 分别表示分析栈顶三个元素所对应的下标(词、字串开始字符的下标或者子树根节点开始字符的下标)，b_0 表示待处理队列中首字符的下标；o是拼接运算。在获取 N-gram 特征层状态信息时，为了简化模型的运算量，本文在公式（7）的计算中只采用了 4 个特征[4,9]。

4．3 多层感知机决策层

本节利用前面介绍的依存子树编码层和 N-gram 特征层的状态表示，搭建基于多层感知机的转移动作决策模型，如图 2 所示。前两个 *relu* 隐藏层对 N-gram 特征层的状态表示进行信息的加工再抽取，获得更深层次的 N-gram 特征表示和更有效的全局信息。最后一个 *relu* 隐藏层融合前两个 *relu* 隐藏层的输出向量 h 与依存子树编码层的状态表示 STL 向量，得到当前状态下模型整体信息表示 h_1，其计算方法如公式（8）所示。

$$h_1 = \max\{0, W_1(h \circ STL) + b_1\} \quad (8)$$

其中 W_i 是最后一个 *relu* 隐藏层的参数矩阵，b_i 是偏置项。

最后设置 softmax 层将多层感知机输出向量 h_1 映射到转移动作概率空间，得到转移动作的预测概率分布，如公式（9）所示，其中 W_p 是权重矩阵。

$$p_i = softmax(W_p h_1) \quad (9)$$

本文采用贪心算法确定转移动作，从当前状态下有效的转移动作中选出概率最大的动作，然后对待处理队列和分析栈执行相应的操作，更新模型状态。

4．4 模型训练

从训练数据的标注树中抽取其对应的转移动作序列，用于一体化依存分析神经网络模型的训练。本文采用交叉熵损失函数作为训练目标，并使用 l_2 正则化缓解过拟合现象，目标函数如公式（10）所示。

$$\lambda(\theta) = -\sum_{i \in A} \log p_i + \frac{\lambda}{2}\|\theta\|^2 \quad (10)$$

其中 A 是训练数据的转移动作序列；θ 是模型所有参数的集合；$\|\theta\|^2$ 是 l_2 范数的正则化项，用来减少参数空间；λ 用来控制正则化的强度。本文还使用 dropout[10]随机删除网络中的某些神经单元来缓解过拟合现象，并采用 Adam[11]算法和误差反向传播方式学习模型参数。Stack-Tree LSTM、N-gram 特征层的双向 LSTM 和多层感知机在目标函数约束下同步更新。

5 实验

我们采用多种依存树编码方式设计对比实验，验证本文所提方法的有效性，并与已有的一体化依存分析模型进行比较。

5．1 实验数据及评价方法

本文使用实验数据为宾州汉语树库 CTB5，并按照已有工作[1-4]的数据划分方案，其中训练集为 1-270 篇、400-931 篇和 1001-1151 篇；开发集为 301-325 篇；测试集为 271-300 篇。实验数据集详细信息如表 1 所以。

我们使用斯坦福分词工具[12]对 gigaword 生语料进行分词，然后利用 word2vec[13]预训练字向量和词向量。分词、词性标注和依存句法分析的评测指标均采用准确率、召回率、综合性能指标 F1 值。其中词性标注和依存分析的评测均是在分词正确的基础上进行，若分词错误，则相应的词性标注和依存分析结果也被视为错误。在依存句法分析的评测时，具有依存关系的两个词语均被正确分词且依存弧的方向正确才被视为正确的依存关系；遵循惯例，与

标点符号有关的依存关系不予考虑。

表 1　数据集统计信息

	句子	词语	集外词
训练集	18k	494k	*
开发集	350	6.8k	553
测试集	348	8.0k	278

5．2 参数设置

我们使用训练集、开发集以及预训练得到的字向量和词向量进行初步实验，获取超参数。对于预训练中未获得的字向量和词向量进行随机初始化。模型训练终止条件为三项任务在开发集上的 F 值趋于稳定。最终获得的各项超参数如下：字向量和词向量维度均设置为 200，双向 LSTM 和 Stack-Tree LSTM 隐藏单元个数均为 200，词性向量维度为 32。多层感知机隐藏层节点数为 400，模型初始学习率为 0.001，dropout 值为 0.2。

5．3 实验结果与分析

对于本文提出的依存子树编码方法和词性使用方法，我们在下面分别分析它们在模型性能改进上的贡献。

依存子树编码方式　我们使用上面参数训练本文提出的一体化依存分析神经网络模型，由 Stack-Tree LSTM 依存子树编码层、N-gram 特征层和多层感知机决策层组成，命名为 Ours。同时，为了与其他编码方式比较，我们对模型中的依存子树编码层进行替换，分别用 Stack LSTM[8]、递归卷积神经网络 RCNN[14]和双向 LSTM[15]替换并训练得到三个模型，分别命名为 Stack LSTM、RCNN 和 BiLSTM。四个模型在测试集上的评测结果如表 2 所示。

实验结果显示，在词性标注任务和依存分析任务上，本文所提模型（Ours）的性能优于 Stack LSTM、RCNN 和 BiLSTM 三个模型，表明 Stack-Tree LSTM 依存子树编码方法在依存子树信息获取方面具有较强的能力。同时，在分词任务上，本文所提模型的性能略低于 RCNN 模型，主要原因是 RCNN 模型在对依存子树编码时还使用了依存距离特征，而目前我们的模型还未利用该特征，今后我们将研究融合该特征的模型。

表 2　不同依存子树编码方法对模型性能的影响

模型	分词	词性标注	依存分析
Stack LSTM	97.71	93.36	79.21
BiLSTM	96.69	92.10	79.02
RCNN	**98.03**	93.35	79.58
Ours	97.78	**93.51**	**79.66**

词性特征的影响　为了验证词性特征对每个任务的影响，我们在 Ours 模型中删除词性特征，重新训练模型,命名为-POS。具体的，在依存子树编码层，字串成词时，其词性向量不加入词向量。在测试集上的评测结果如表 3 所示。表 3 结果显示，-POS 模型在分词和依存分析任务上的 F1 值均低于使用词性特征的模型，说明词性特征的使用对模型性能的提升有帮助，特别在依存分析任务上 F1 值提升了 0.2 个百分点。

与已有工作对比　为了与已有的汉语一体化依存分析模型进行对比，我们在表 3 中还列出了已有代表性工作的评测结果。其中 Hatori12、Zhang14 和 Guo14 是基于特征工程的一体化依存分析模型；Kurita17（4feat）和 Kurita17（8feat）是基于神经网络的一体化依存分析模型；Kurita17 是基于特征工程与神经网络融合的一体化依存分析模型。

首先进行神经网络模型之间的比较。与 Kurita17（4feat）相比，Kurita17（8feat）使用了依存子树部分孩子节点的特征信息，在词性标注和依存分析任务上取得了更好的 F1 值，但在分词任务上的 F1 值低于前者。本文模型 Ours 在三项任务上的性能均优于 Kurita17

（4feat）和 Kurita17（8feat），与两个模型的最好结果相比，在分词、词性标注和依存分析三项任务上分别提升了 0.06%、0.14%和 0.28%。考虑到这两个模型未使用词性特征，我们将本文模型 Ours 去除词性特征的模型-POS 与之比较，-POS 模型在三项任务上的性能仍优于这两个模型。这些对比结果显示，本文模型在基于神经网络的模型中取得了最好的结果，表明本文提出的Stack-Tree LSTM依存子树编码方法可以有效获取分析栈中各依存子树的整体信息，而且这些信息对于一体化依存分析的动作决策发挥了有效的指导作用，帮助模型提高性能。

表 3　汉语一体化依存分析模型的对比结果。*代表使用了大量特征模板

模型	解码方法	分词	词性标注	依存分析
Hatori12*	beam	97.75	94.33	81.56
Guo14*	beam	97.52	93.93	79.55
Zhang14*	beam	97.67	94.28	**81.63**
Kurita17*	greedy	**98.24**	**94.49**	80.15
Kurita17(4feat)	greedy	97.72	93.12	79.03
Kurita17(8feat)	greedy	97.70	93.37	79.38
Ours	greedy	**97.78**	93.51	**79.66**
-POS	greedy	97.74	**93.53**	79.45

然后与基于特征工程的模型比较。在基于特征工程的模型中，Hatori12 在分词和词性标注任务上的 F1 值最高，Zhang14 在依存分析任务上 F1 值最高而且是目前所有模型中最高水平。与这些结果对比，本文模型在分词任务上的 F 值高于 Hatori12，但是在词性标注上的 F1 值略低于 Hatori12，在依存分析上的 F1 值略低于 Zhang14，说明我们的神经网络模型在仅利用很少特征的情况下已经非常接近基于特征工程的模型，表明本文所提出的依存子树编码方法的有效性。同时，表明基于特征工程的模型在特征选取与使用上达到了相当成熟的水平。

最后与特征工程和神经网络融合的模型比较。Kurita17 使用了特征工程中得到的特征模板共 50 个，并采用分布式方法表示。该模型在分词和词性标注上的 F1 值达到了目前所有模型中最高水平，表明了两种方法融合的有效性。与之相比，Kurita17（4feat）和 Kurita17（8feat）分别只使用了 4 个和 8 个特征模板，但是由于使用双向 LSTM 对句子进行编码，有效获取句子的全局信息，所以即使使用少量特征模板也达到了接近特征工程的效果，表明神经网络模型具有自动学习所需特征的能力。本文模型同样只使用了 4 个特征模板，在三项任务上的 F1 值超过了 Kurita17（4feat）和 Kurita17（8feat），更接近 Kurita17 模型，表明本文的 Stack-Tree LSTM 依存子树编码的有效性。

目前分词、词性标注和依存分析任务上的最好结果分别为 98.24%、94.49%和 81.63%，其中分词和词性标注由 Kurita17 特征工程和神经网络融合模型获得，依存分析由 Zhang14 基于特征工程的模型获得，这些模型均使用了大量特征模板。从以上评测结果的比较与分析中，我们发现基于神经网络的模型仅使用少量特征模板已经取得了接近使用大量特征模板模型的性能，表明本文神经网络模型具有自动学习特征的能力，同时更全面的利用中间结果信息，学习到传统特征工程未能覆盖的有效特征；通过引入依存子树编码机制为一体化依存分析模型利用分析栈中从未被使用的中间结果提供了一条新的途径。

6 总结

本文提出了 Stack-Tree LSTM 依存子树编码方法，通过对分析栈中所有依存子树的整体信息进行编码，为三个并行任务提供所有中间结果信息作为后续分析的特征。我们以此为基础搭建一体化依存分析神经网络模型，并提出词性特征的使用方法。CTB5 上的对比评测实验验证了本文提出的依存子树编码方法和词性特征使用方法的有效性。作为今后的工作，我

们将考虑在一体化依存分析神经网络模型中融入经典的特征模板知识，增加对比实验。

参考文献

[1] Hatori J, Matsuzaki T, Miyao Y, et al. Incremental joint approach to word segmentation, POS tagging, and dependency parsing in Chinese[C]// Meeting of the Association for Computational Linguistics: Long Papers. Association for Computational Linguistics, 2012:1045-1053.

[2] Zhang M, Zhang Y, Che W, et al. Character-Level Chinese Dependency Parsing[C]// ACL. 2009.

[3] 郭振, 张玉洁, 苏晨,等. 基于字符的中文分词、词性标注和依存句法分析联合模型[J]. 中文信息学报, 2014, 28(6):1-8.

[4] Kurita S, Kawahara D, Kurohashi S. Neural Joint Model for Transition-based Chinese Syntactic Analysis[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017, 1: 1204-1214.

[5] 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.

[6] Tai K S, Socher R, Manning C D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks[J]. Computer Science, 2015, 5(1):: 36.

[7] Bowman S R, Gauthier J, Rastogi A, et al. A Fast Unified Model for Parsing and Sentence Understanding[J]. 2016.

[8] Dyer C, Ballesteros M, Ling W, et al. Transition-Based Dependency Parsing with Stack Long Short-Term Memory[J]. Computer Science, 2015, 37(2):321–332.

[9] Kiperwasser E, Goldberg Y. Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations[J]. 2016.

[10] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1):1929-1958.

[11] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.

[12] Tseng H, Chang P, Andrew G, et al. A conditional random field word segmenter for sighan bakeoff 2005[C]// 2005:168--171.

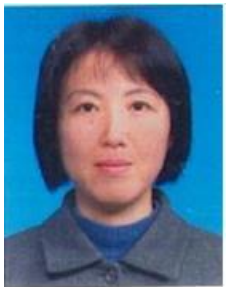
[13] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.

[14] Zhu C, Qiu X, Chen X, et al. A Re-ranking Model for Dependency Parser with Recursive Convolutional Neural Network[J]. Computer Science, 2015.

[15] Dyer C, Kuncoro A, Ballesteros M, et al. Recurrent Neural Network Grammars[J]. 2016:199-209.

刘航（1996——），男，硕士研究生，主要研究领域为自然语言处理、依存句法分析。
Email: 16120389@bjtu.edu.cn

刘明童（1993——），男，博士研究生，主要研究领域为自然语言处理、神经机器翻译、复述。
Email: 16112075@bjtu.edu.cn



张玉洁（1961——），女，教授，主要研究领域为自然语言处理和机器翻译，通讯作者。

Email: yjzhang@bjtu.edu.cn

徐金安（1970——），男，副教授，主要研究领域为自然语言处理和机器翻译。

Email: jaxu@bjtu.edu.cn

陈钰枫（1981——），女，副教授，主要研究领域为自然语言处理和机器翻译。

Email: chenyf@bjtu.edu.cn

作者联系方式：刘航，北京市海淀区上园村 3 号，北京交通大学计算机与信息技术学院，100044，18800102825，16120389@bjtu.edu.cn；刘明童，北京市海淀区上园村 3 号，北京交通大学计算机与信息技术学院，100044，16112075@bjtu.edu.cn；张玉洁，北京市海淀区上园村 3 号，北京交通大学计算机与信息技术学院，100044，yjzhang@bjtu.edu.cn；徐金安，北京市海淀区上园村 3 号，北京交通大学计算机与信息技术学院，100044，jaxu@bjtu.edu.cn；陈钰枫，北京市海淀区上园村 3 号，北京交通大学计算机与信息技术学院，100044，chenyf@bjtu.edu.cn。