

文章编号: 1003-0077(2018)12-0041-07

基于多特征融合编码的神经网络依存句法分析模型

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: 在基于神经网络的依存句法分析中, 对分析栈和决策层信息的表示和利用依然有值得深入研究的空间。针对分析栈的表示, 已有工作并没有对单棵依存子树独立编码的表示, 导致无法利用各个依存子树的局部特征; 也没有对生成的依存弧序列进行编码, 导致无法利用依存弧的全局信息。针对决策层的表示, 已有工作利用 MLP 预测转移动作, 该结构无法利用历史决策动作的信息。对此, 该文提出基于多特征融合编码的神经网络依存句法分析模型, 基于依存子树和历史生成的依存弧表示分析栈, 利用 TreeLSTM 网络编码依存子树信息, 利用 LSTM 网络编码历史生成的依存弧序列, 以更好地表示分析栈的局部信息和全局信息。进一步提出基于 LSTM 网络的结构预测转移动作序列, 引入历史决策动作信息作为特征辅助当前决策。该文以汉语为具体研究对象, 在 CTB5 汉语依存分析数据上验证所提出的多特征融合编码的神经网络模型。实验结果显示, 汉语依存句法分析性能得到改进, 在目前公布的基于转移的分析系统中取得最好成绩, 在 UAS 和 LAS 评价指标上分别达到 87.8% 和 86.8% 的精度, 表明所提出的对依存子树局部特征及历史依存弧信息和历史决策动作信息的编码方法, 在改进依存分析模型性能方面的有效性。

关键词: 依存句法分析; 多特征融合编码; 依存子树; TreeLSTM 神经网络

中图分类号: TP391

文献标识码: A

A Neural Transition-based Dependency Parsing Model with Fused Multi-feature Encoding

LIU Mingtong, ZHANG Yujie, XU Jin'an, CHEN Yufeng

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

Abstract: For neural networks based dependency parsing, this paper presents a novel architecture for transition-based dependency parsing leveraging fused multi-feature encoding. We model the stack states based on subtrees representations and encode structural dependency subtrees with TreeLSTM. Particularly, we propose a LSTM-based technique to encode the historical parsed dependency arcs and states as global features. Finally, based on fused multi-feature encoding, we combine the extracted local features and global features for parsing decision. Experiments on Chinese Penn TreeBank (CTB5) show that our parser reaches 87.8% (unlabeled) and 86.8% (labeled) attachment accuracy with a greedy strategy, which effectively improves neural transition-based dependency parsing.

Keywords: dependency parsing; multi-feature encoding; dependency subtree; TreeLSTM neural network

0 引言

依存句法分析一直是自然语言处理中的关键技术之一, 广泛应用于机器翻译^[1]、信息抽取^[2]和语义角色标注^[3]等自然语言处理任务中。目前主流的依存句法分析框架主要包括基于转移 (transition-

based)^[4-5] 和基于图 (graph-based)^[6] 的两种模型。

近年来, 基于神经网络的依存句法分析方法日益成为关注热点, 是自然语言处理的重要方向之一。本文主要针对基于转移的依存句法分析模型的分析栈和决策层的表示方法开展研究。

早期的方法主要依赖人工定义特征模板提取依存分析过程中的特征^[4-7]。主要问题是特征模板的

收稿日期: 2018-09-29 定稿日期: 2018-10-29

基金项目: 中央高校基本科研业务费专项资金(2018YJS025, 2015JBM033); 国家自然科学基金(61370130, 61473294); 科学技术部国际科技合作计划(K11F100010); 国家自然科学基金(61876198)

构建需要大量专家知识,而且系统在特征提取上的计算代价较大^[8]。后来,基于神经网络的依存句法分析模型显示出了优势^[9-12],研究者利用神经网络建模分析栈、缓冲区和决策序列,通过神经网络进行特征的自动提取,将依存分析过程中每一时刻各个部分对应的状态表示为低维稠密的特征向量,不需要人工定义大量的特征模板,避免了复杂的特征工程。基于神经网络的模型主要研究分析过程中特征的表示和利用方法,已有模型^[9-11]在局部信息和全局信息的表示上存在以下问题。一方面,没有对单棵依存子树独立编码表示,导致无法利用各个依存子树的局部特征,而依存关系的判断是针对当前两个节点(节点可以是依存子树根节点,也可以是单词)预测正确的转移动作,因此依存关系的建立需要直接考虑依存子树的信息作为特征;另一方面,这些模型没有对历史生成的依存弧信息和转移动作信息进行编码,以致整个分析过程丢失了全局信息,而依存树的建立需要从句子整体的视角考虑更多的全局特征^[7]。为了利用更多的局部特征和全局特征,针对分析栈的表示,本文提出基于依存子树和历史已生成的依存弧表示分析栈,利用 TreeLSTM 网络编码依存子树信息,利用 LSTM 网络编码历史生成的依存弧序列,以更好地表示分析栈的局部信息和全局信息。针对决策层的表示,本文提出基于 LSTM 网络的转移动作序列预测,引入历史决策动作信息作为特征辅助当前决策。

本文组织如下:第 1 节介绍相关研究;第 2 节针对多特征融合编码中的局部特征和全局特征,描述基于依存子树和历史生成依存弧的分析栈表示方法,以及基于 LSTM 网络的转移动作序列预测方法;第 3 节介绍依存句法分析模型的实现和训练细节;第 4 节介绍评测实验和结果分析;第 5 节对本文研究进行总结。

1 相关研究

在基于神经网络的依存句法分析方面,Chen 和 Manning^[9]利用神经网络将单词、词性标签和依存类型标签表示为连续、低维、稠密的向量,只利用少量人工定义的特征模板,就比传统完全基于特征模板的方法取得了更好的性能。该方法主要利用核心特征模板抽取局部特征作为决策输入,其主要问题是没有考虑依存子树的局部特征,而且该方法没有提供历史分析信息参与决策。为了利用更多的历史

信息,Dyer^[10]等人提出基于转移的 Stack-LSTM 依存分析模型,Wang^[12]等人进一步改进 Stack-LSTM 模型中依存子树的编码方法。Stack-LSTM 模型采用三个单向的 LSTM 网络编码依存分析过程中的信息,将传统的缓冲区、分析栈和转移动作序列的历史均记录在 LSTM 单元中,使用 LSTM 的最后一个隐藏层表示作为决策输入,最后利用多层感知机预测转移动作。该方法实现了完全自动的特征提取,模型精度得到提升。其主要问题是利用 LSTM 累计的信息表示当前分析栈的状态,无法表示依存子树的局部特征,而依存关系的建立需要考虑节点所在依存子树的信息。Kiperwasser 和 Goldberg^[11]采用 BiLSTM 网络获取单词的上下文特征表示,用分析栈栈顶的多个元素作为特征,考虑了依存子树根节点单词表示对转移动作的影响,但是没有考虑依存子树的整体信息,也没有考虑历史分析信息,导致决策丢失了全局信息。针对这些问题,本文研究基于依存子树和依存弧相结合的分析栈表示方法,同时研究历史转移动作信息的表示方法,通过多特征融合编码,以有效利用分析过程中的局部特征和全局特征,改进转移动作决策的精度。

2 基于多特征融合编码的神经网络依存句法分析

我们提出基于多特征融合编码的神经网络依存句法分析模型,采用基于转移的依存句法分析框架,主要框图如图 1 所示。下面从分析栈表示方法、缓冲区表示方法和转移动作序列表示方法三个模块详细介绍我们的模型。

2.1 基于转移的依存句法分析器

基于转移的依存句法分析框架^[4-5]由以下三部分组成:分析栈 S(stack)存放已完成的依存子树,每个元素存放依存子树的根节点;缓冲区 B(buffer)存放待分析句子的单词序列;依存弧 A(dependency arcs)存放已生成的依存关系(head-modifier pairs)。分析栈栈顶的第一个元素定义为右焦点词,分析栈栈顶的第二个元素定义为左焦点词,它们分别是两颗子树的根节点。三种动作的定义如下:

① SHIFT:将缓冲区的第一个元素移入分析栈中。

② LEFT-ARC(l):针对栈顶两个元素,建立左焦点词依存于右焦点词的依存关系。

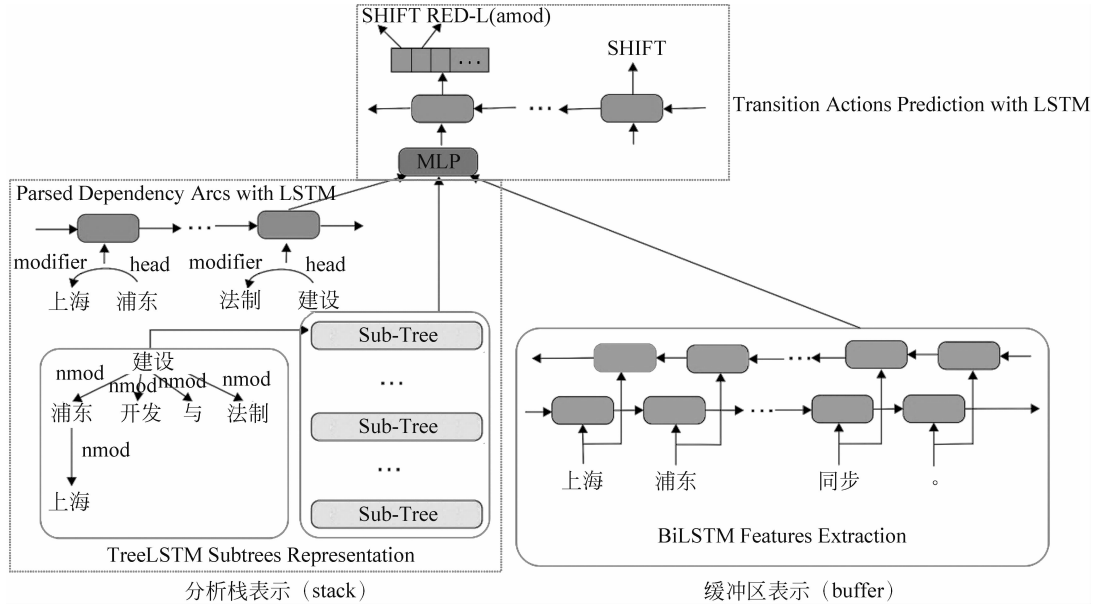


图1 本文提出的基于多特征融合编码的神经网络依存句法分析模型

③ RIGHT-ARC(1): 针对栈顶两个元素, 建立右焦点词依存于左焦点词的依存关系。

依存关系的建立发生在左焦点词和右焦点词之间, 即子树的根节点。一旦 LEFT-ARC(1) 或 RIGHT-ARC(1) 动作发生, 两个焦点词中的一个会成为另一个的子节点, 即生成一棵依存子树。由此, 依存句法分析转换成转移动作决策问题, 基于当前分析状态的信息预测下一步转移动作, 其核心是依存分析状态的表示和利用, 分析状态包括分析栈、缓冲区、依存弧以及转移动作序列。本文的工作集中在基于神经网络的分析状态表示方法, 通过设计神经网络结构有效地提取和利用分析状态作为特征, 用于转移动作决策。

2.2 分析栈表示方法

分析栈状态包括依存子树 (dependency subtrees) 和历史已生成的依存弧 (head-modifier pairs)。对此, 我们提出基于 TreeLSTM 网络的依存子树表示方法和基于 LSTM 网络的依存弧序列表示方法。

(1) 基于 TreeLSTM 的依存子树表示方法

我们设计基于 TreeLSTM^[13] 神经网络的结构对已完成的每棵依存子树单独编码, 通过门控制机制 (gate mechanism) 选择输入信息, 在合成子树表示时选择性地关注诸如动词、名词等重要节点, 从而更好地编码依存子树的表示。已有的分析栈表示方法^[10] 累计 LSTM 的最后一个隐藏层向量表示整个

分析栈的状态, 该结构无法利用依存子树的信息。同时, 已有方法^[10] 采用 Recursive Neural Network (RecNN) 编码依存子树, 该结构在处理较深的依存子树时会出现梯度消失的风险。在此, 我们关注每棵依存子树的信息, 对于新建立依存关系的头节点和修饰节点, 我们定义 h^{head} 和 c^{head} 为建立依存关系的头节点的隐藏层向量和记忆单元向量, 定义 h^{modi} 和 c^{modi} 为建立依存关系的修饰节点的隐藏层向量和记忆单元向量。对于新合成的依存子树, 定义 h^{st} 和 c^{st} 为依存子树根节点的隐藏层向量和记忆单元向量, h^{st} 和 c^{st} 的计算如式(1)~式(3)所示。

$$\begin{bmatrix} i \\ f^{\text{head}} \\ f^{\text{modi}} \\ o \\ \tilde{c} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(W^1 \begin{bmatrix} h^{\text{head}} \\ h^{\text{modi}} \end{bmatrix} + b^1 \right) \quad (1)$$

$$c^{\text{st}} = f^{\text{head}} \odot c^{\text{head}} + f^{\text{modi}} \odot c^{\text{modi}} + i \odot \tilde{c} \quad (2)$$

$$h^{\text{st}} = o \odot \tanh(c^{\text{st}}) \quad (3)$$

其中, σ 表示 sigmoid 函数, \odot 表示两个向量的逐点乘积操作。 i 、 f^{head} 、 f^{modi} 、 o 和 \tilde{c} 分别表示输入门、头节点对应的遗忘门、修饰节点对应的遗忘门、输出门和候选记忆单元。在我们的结构中, 子树表示是动态生成的, 每一步只考虑当前产生依存关系的两个节点, 从而递增地得到整棵依存子树的表示, 其包含了当前所有已找到的修饰节点信息。

(2) 基于 LSTM 的依存弧序列表示方法

我们提出基于 LSTM 网络的依存弧序列表示

方法,对已生成的历史依存弧序列编码作为全局特征。具体地,当一个依存弧生成时,我们将头节点和修饰节点作为 LSTM^[14] 网络的输入,利用 LSTM 网络的门控机制选择性的记忆历史信息,累计已生成的依存弧信息,得到全局特征表示辅助转移动作决策。依存弧序列编码计算如式(4)所示。

$$h_t^h = \text{LSTM}([h^{\text{head}}; h^{\text{modi}}], h_{t-1}^h) \quad (4)$$

其中, h^{head} 和 h^{modi} 分别表示头节点和修饰节点对应的隐藏层向量。

由此,我们用 TreeLSTM 对依存子树编码获取局部特征,用 LSTM 对依存弧序列编码获取全局特征,通过增加对分析栈中局部特征和全局特征的编码表示获取更多特征参与转移动作预测。

2.3 基于 LSTM 的转移动作序列表示方法

我们设计基于 LSTM 网络的转移动作序列表示方法,利用 LSTM 对时序信息的传递和记忆能力,对已产生的转移动作序列进行编码,在决策时引入了历史转移动作的信息。转移动作决策时, LSTM 网络的输入包括四个部分:分析栈栈顶 3 个元素 $h_t^{s1}, h_t^{s2}, h_t^{s3}$ 、缓冲区队列第一个元素 h_t^{b1} 、历史生成的依存弧信息 h_t^h 和该网络上一时刻的隐藏层表示 h_{t-1}^p 。首先利用式(5)整合分析栈、缓冲区和历史依存弧序列编码的特征,然后利用式(6)降低 LSTM 输入的维度。式(7)利用了上一时刻的信息和当前时刻的输入计算隐藏层表示,式(8)利用 softmax 分类器输出转移动作选择的概率。

$$p_t^1 = [h_t^{s1}; h_t^{s2}; h_t^{s3}; h_t^{b1}; h_t^h] \quad (5)$$

$$p_t^2 = \text{ReLU}(W^2 p_t^1 + b^2) \quad (6)$$

$$h_t^p = \text{LSTM}(p_t^2, h_{t-1}^p) \quad (7)$$

$$p_a = \text{softmax}(W^3 h_t^p + b^3) \quad (8)$$

由于基于转移的依存分析框架是一种局部的决策式处理方法,因此需要更多的全局信息帮助句法结构消歧,我们分别累计依存弧和转移动作的历史信息获取全局特征,参与转移动作决策。

2.4 缓冲区表示方法

我们设计基于 BiLSTM 的缓冲区表示方法,对构成句子的所有单词进行累计的表示,即使单词移除缓冲区,信息也不会丢失,从而避免了单向 LSTM 编码方式丢失移出缓冲区单词信息的问题^[10]。为了获取更加丰富的上下文信息表示单词特征,我们结合 LSTM 每一时刻的隐藏层向量 h_t 和记忆单元向量 c_t 表示缓冲区中的每一个单词如

式(9),式(10)所示。

$$\vec{h}_t, \vec{c}_t = \overrightarrow{\text{LSTM}}(x_t, (\vec{h}_{t-1}, \vec{c}_{t-1})) \quad (9)$$

$$\vec{h}_t, \vec{c}_t = \overleftarrow{\text{LSTM}}(x_t, (\vec{h}_{t+1}, \vec{c}_{t+1})) \quad (10)$$

我们整合 \vec{h}_t 和 \vec{h}_t , 获得当前时刻的隐藏层表示 h_t , 整合 \vec{c}_t 和 \vec{c}_t 获得当前时刻的记忆单元表示 c_t 。由于序列化的 LSTM 网络在计算当前时刻的信息时传递了历史的信息,因此,同一个单词在不同的上下文中,会得到不同的词向量,从而可以学习到有效的基于上下文的单词特征表示参与转移动作决策。

3 模型训练和实施细节

我们利用上面提出的多特征融合编码方法,搭建神经网络依存句法分析模型,模型的训练目标和实施细节描述如下。

首先,我们利用端到端的方法训练整个模型。目标函数采用交叉熵损失函数,添加 l_2 正则化项防止过拟合,通过超参数 λ 调整目标函数权重,如式(11)所示。

$$l(\theta) = - \sum p_i + \frac{1}{2} \lambda \|\theta\|_2^2 \quad (11)$$

其中, θ 表示模型中所有的参数。在选取转移动作时,我们只计算当前状态下有效的动作,例如,当分析栈中元素为空、缓冲区存在单词时,只能执行 SHIFT 转移动作。最终,在训练阶段,每一步决策按照标准的动作执行分析操作;在测试阶段,我们采用贪心决策算法选取当前概率最大的动作执行分析操作。

其次,关于输入词向量表示,本文中,我们采用三种类型的向量表示当前输入单词 x_i : 预训练的词向量 v_i^{pre} 、可学习的词向量 v_i^{learn} 和词性标签向量 v_i^{pos} 。其中,预训练的词向量由 Skip-gram 模型得到^[15],和已有工作一致^[10],我们采用 Chinese Gigaword 语料作为训练数据。最后,我们采用一个非线性变换获得三个部分整合的词向量表示 x_i ,如式(12)所示。

$$x_i = \text{ReLU}(W^4 [v_i^{\text{pre}}; v_i^{\text{learn}}; v_i^{\text{pos}}] + b^4) \quad (12)$$

在此,我们添加“ROOT”词表示根节点,“NULL”词表示可能用到的无效特征组合(例如,分析栈中元素个数小于 3 时),未登录词通过高斯采样得到。

最后,关于模型实施具体细节描述如下。我们设置预训练的词向量为 300 维,可学习的词向量为 30 维,词性标签向量为 50 维,所有的 ReLU 层为 300 维。设置 BiLSTM、LSTM 和 TreeLSTM 结构

的隐藏层为 300 维。在实验中,预训练的词向量保持固定不变。我们采用 Adam^[16] 优化算法用于模型参数学习,设置初始学习率 α 为 $5E-4$, β_1 为 0.9, β_2 为 0.999。在每一轮迭代中,学习率 α 以 $\rho=0.95$ 的频率衰减。设置 l_2 正则化强度为 $1E-6$,训练 batch 的大小为 8。为了防止过拟合,我们使用了 dropout^[17] 正则化技术,设置词向量输入层的 drop 率为 0.2, BiLSTM 输出层的 drop 率为 0.5, ReLU 层输出层的 drop 率为 0.07。我们对预训练的词向量和最终整合的词向量使用了 batch normalization^[18] 正则技术。其中,dropout 层作用在 batch normalization 层之后。

4 评测实验与结果分析

为了验证本文所提方法的有效性,我们在汉语数据上进行了评测实验。

4.1 实验数据

本文使用宾州汉语树库 CTB5 作为实验数据,我们采用了标准的数据划分^[9],利用标注数据中的分词和词性标签进行依存句法分析,表 1 给出了 CTB5 汉语数据的统计信息。

表 1 CTB5 汉语数据统计信息

数据	训练集	开发集	测试集	词数	词性 标签数	依存类型 标签数
CTB	16 091	803	1 901	34 577	35	12

4.2 评测实验结果

实验过程中,我们使用 UAS 和 LAS 两种评测指标评价模型的性能,和先前工作一致,评测结果不考虑标点符号^[9],利用开发集选择性能最好的模型参数用于测试阶段。我们和已有公开发表的基于转移的汉语依存句法分析系统进行了性能比较,比较结果如表 2 所示。UAS 和 LAS 两种评测指标定义如下。

表 2 和已有基于转移的汉语依存分析系统的性能比较结果

系统	UAS	LAS
Zhang and Nivre (2011) ^[7]	86.0	84.4
Chen and Manning (2014) ^[9]	83.9	82.4
Dyer et al. (2015) ^[10]	87.2	85.7
Ballesteros et al. (2016) ^[19]	87.6	86.2

续表

系统	UAS	LAS
Kiperwasser and Goldberg (2016) ^[11]	87.6	86.1
本文提出的模型	87.8	86.8

① 不考虑依存弧类型的依存正确率(unlabeled attachment score, UAS): 修饰关系正确的单词(包括根节点对应的单词)数量占总词数的百分比。

② 考虑依存弧类型的依存正确率(labeled attachment score, LAS): 修饰关系和依存弧类型都正确的单词(包括根节点对应的单词)数量占总词数的百分比。

从表 2 的结果可以看出,相比于 Dyer 提出的 Stack-LSTM 依存句法分析系统^[10],我们的模型在 UAS 和 LAS 评测指标上分别取得了 0.6 和 1.1 个百分点的提高。相比于 Kiperwasser 和 Goldberg 的依存句法分析系统^[11],我们的模型在 UAS 和 LAS 评测指标上分别取得了 0.2 和 0.7 点的提高。和已有模型的比较结果表明,本文提出的基于多特征融合编码的依存分析模型,对依存子树局部特征、历史依存弧信息和历史决策动作信息的编码对于改进依存句法分析性能的有效性。最终,通过引入更多的局部信息和全局信息参与转移动作决策,我们的模型达到 87.8% 的 UAS 得分和 86.8% 的 LAS 得分,为目前公开的基于转移的系统中最好的性能。

4.3 模型各个部分有效性分析

我们分析了模型中各个部分对最终依存句法分析精度的影响,实验结果展示在表 3 中。

表 3 模型各个部分有效性分析结果

模型	UAS	LAS
本文提出的模型	87.8	86.8
(1) 一基于 TreeLSTM 的依存子树	87.3	86.3
(2) 一基于 LSTM 的历史依存弧	87.4	86.3
(3) 一基于 LSTM 的转移动作预测	87.4	86.5
(4) 一基于 BiLSTM 的缓冲区表示	83.5	81.7
(5) 一(2)和(3)	87.2	86.2

从表 3 的实验结果来看,移除我们模型中的任何一部分(1)~(5)都会降低依存分析的精度,由此证明了我们所提模型中各个部分的有效性。针对分析栈的表示,当移除基于 TreeLSTM 编码的依存子树表示(1)时,模型没有利用单棵依存子树的信息作

为特征预测转移动作,只利用了子树根节点单词基于上下文的 BiLSTM 隐藏层表示参与决策,模型性能在 UAS 和 LAS 评测指标上都下降了 0.5 个百分点,这一结论表明了依存子树的信息在转移动作预测中的重要性,同时也表明了预测两个节点之间的依存关系时局部特征需要考虑单棵依存子树节点的完整信息。当移除基于 LSTM 编码的历史依存弧序列表示(2)时,模型性能在 UAS 和 LAS 评测指标上分别下降了 0.4 和 0.5 个百分点,这表明引入历史依存弧的信息作为全局特征用于决策,提高了转移动作预测精度。

针对决策层的表示,当移除基于 LSTM 的转移动作预测(3),仅利用 MLP 作为预测函数时,模型性能在 UAS 和 LAS 上分别下降了 0.4 和 0.3 个百分点。由此可以发现 LSTM 比 MLP 在预测转移序列上更为有效,因为 LSTM 可以传递历史转移动作和分析状态的信息,从而引入更多的全局特征用于

当前决策。

针对缓冲区表示,当我们移除基于 BiLSTM 的单词特征表示层(4),仅利用词向量表示缓冲区每一个元素时,模型性能在 UAS 和 LAS 评价指标上分别下降了 4.3 和 5.1 个百分点,由此表明依存句法分析需要从句子整体考虑更多的上下文信息。最终,实验结果表明,本文提出的基于 TreeLSTM 网络编码依存子树和基于 LSTM 网络编码依存弧序列的表示方法,以及基于 LSTM 网络编码的转移动作序列预测模型,通过多特征融合编码,将更多依存分析中的局部特征和全局特征用于转移动作决策,有效提高了依存句法分析的精度。

4.4 不同依存类型的 LAS 分析

本节针对模型结构对依存分析性能的影响,分别调查了不同类型依存弧的 LAS 精度,在 CTB 测试集上的实验结果如图 2 所示。

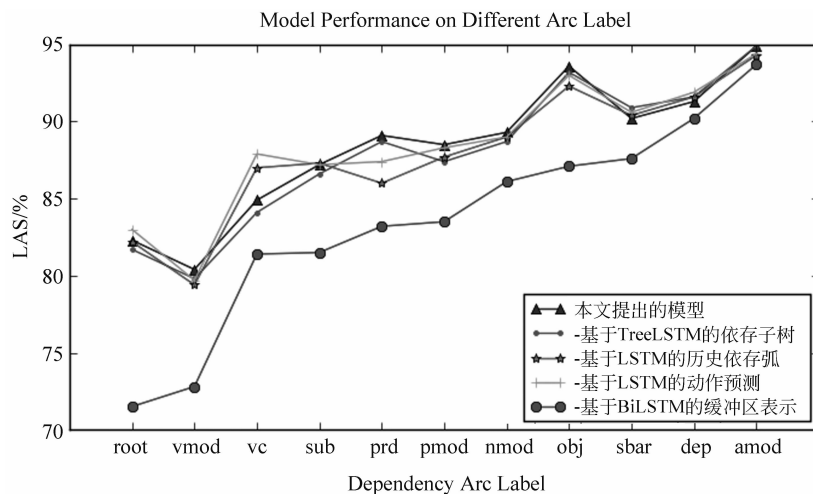


图2 模型结构对不同依存类型的 LAS 影响结果

从图 2 的结果来看, root 和 vmod 依存类型的精度较低,表明了这类依存关系较难正确预测。在移除基于 TreeLSTM 网络的依存子树表示时,图 2 的结果展示了利用依存子树作为特征预测转移动作提高了几乎所有依存类型的精度,进一步表明了编码依存子树信息对依存关系预测的重要性。同时,图 2 的结果也表明利用更多的历史分析信息作为全局特征,包括历史生成的依存弧信息、历史转移动作信息,有效提高了模型在 prd、obj 和 pmod 依存类型上的精度。在移除 BiLSTM 的模型中,图 2 展示了各类依存关系的精度都呈现了明显的性能下降,这表明基于上下文特征的单词表示,在依存关系预测中有十分重要的作用。

5 总结

本文提出基于多特征融合编码的神经网络依存句法分析模型,采用基于转移的依存句法分析框架。本文基于依存子树和历史依存弧信息表示分析栈,设计基于 TreeLSTM 网络编码各个依存子树和基于 LSTM 网络编码依存弧序列,以更好地表示分析栈的状态。同时,设计基于 LSTM 网络编码转移动作序列,引入了历史转移动作的信息辅助当前决策。最终,在决策时,我们融合了依存子树信息、缓冲区单词信息、历史依存弧信息和历史转移动作信息用于当前转移动作预测,有效结合了局部特征和全局

特征。实验结果表明本文所提模型明显提高了依存句法分析的精度。针对未来的研究工作,我们需要利用无标注数据获取更多的依存句法训练数据,以提高模型泛化能力。同时,可以结合基于搜索的技术,如 beam-search,进一步改进模型性能。

参考文献

- [1] Su N K, Baldwin T. Interpreting semantic relations in noun compounds via verb semantics [C]//Proceedings of International Conference on Computational Linguistics and Meeting of the Association for Computational Linguistics, Sydney, Australia, 2006; 17-21.
- [2] 高源, 席耀一, 李弼程. 基于依存句法分析与分类器融合的触发词抽取方法[J]. 计算机应用研究, 2016, 33(5):1407-1410.
- [3] 胡禹轩. 基于依存句法分析的语义角色标注[D]. 哈尔滨工业大学硕士学位论文, 2009.
- [4] Nivre J. An efficient algorithm for projective dependency parsing [C]//Proceedings of the 8th International Workshop on Parsing Technologies (IWPT), 2003; 149-160.
- [5] Yamada H, Matsumoto Y. Statistical dependency analysis with support vector machines [C]//Proceeding of International Workshop on Parsing Technologies (IWPT). 2003;195-206.
- [6] McDonald R. Discriminative learning and spanning tree algorithms for dependency parsing [D]. University of Pennsylvania PHD Thesis, 2006.
- [7] Zhang Y, Nivre J. Transition-based dependency parsing with rich non-local features [C]//Proceedings of Meeting of the Association for Computational Linguistics; Human Language Technologies; Short Papers. Association for Computational Linguistics, 2011;188-193.
- [8] He H, Daum III H, Eisner J. Dynamic feature selection for dependency parsing [C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013; 1455-1464.
- [9] Chen D, Manning C. A fast and accurate dependency parser using neural networks [C]//Proceedings of Conference on Empirical Methods in Natural Language Processing, 2014;740-750.
- [10] Dyer C, et al. Transition-based dependency parsing with stack long short-term memory [J]. Computer Science, 2015, 37(2):321-332.
- [11] Kiperwasser E, Goldberg Y. Simple and accurate dependency parsing using bidirectional LSTM feature representations [J]. Transactions of the Association of Computational Linguistics, 2016, 4(1): 313-327.
- [12] Wang Y, et al. A neural transition-based approach for semantic dependency graph parsing [C]//Proceedings of AAAI. 2018.
- [13] 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.
- [14] Schmidhuber J, Hochreiter S. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735-1780.
- [15] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.
- [16] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
- [17] Srivastava N, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15 (1): 1929-1958.
- [18] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//Proceedings of International Conference on International Conference on Machine Learning. JMLR.org, 2015.
- [19] Ballesteros M, et al. Training with exploration improves a greedy stack LSTM parser[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016; 2005-2010.



刘明童(1993—), 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。
E-mail: 16112075@bjtu.edu.cn



徐金安(1970—), 副教授, 主要研究领域为自然语言处理和机器翻译。
E-mail: jaxu@bjtu.edu.cn



张玉洁(1961—), 通信作者, 教授, 主要研究领域为自然语言处理和机器翻译。
E-mail: yjzhang@bjtu.edu.cn