计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.190100108

融入结构信息的指代消解

付 健 孔 芳
苏州大学计算机科学与技术学院 江苏 苏州 251006
(20164227014@stu.suda.edu.cn)

摘 要 随着深度学习的兴起与发展,越来越多的学者开始将深度学习技术应用于指代消解任务中。但现有的神经指代消解模型普遍只关注文本的线性特征,忽略了传统方法中已证明非常有效的结构信息的融入。以目前表现最佳的 Lee 等提出的神经网络模型为基础,借助成分句法树对上述问题进行了改进:1)提出了一种枚举句法树中以结点为短语的抽取策略,避免了暴力枚举策略所受到的长度限制与不符合句法规则的短语集噪音的引入;2)利用树的遍历得到结点序列,结合结点的高度与路径等特征,直接对成分句法树进行上下文表示并将其融入模型中,避免了只使用字、词序列而产生的结构信息缺失问题。在 CoNLL 2012 Shared Task 的数据集上对所提模型进行了一系列实验,实验结果显示,其中文指代消解的 $F1$ 值达到了 62.35,英文指代消解的 $F1$ 值也达到了 67.24,从而验证了所提结构信息融入策略能大大提升指代消解的性能。

关键词:指代消解,成分句法树,结构信息,高度特征,嵌入
中图法分类号 TP391

Coreference Resolution Incorporating Structural Information

FU Jian and KONG Fang
School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 251006, China

Abstract With the rise and development of deep learning, more and more researchers begin to apply deep learning technology to coreference resolution. However, existing neural coreference resolution models only focus on the sequential information of text and ignore the integration of structural information which has been proved to be very useful in traditional methods. Based on the neural coreference model proposed by Lee et al., which has the best performance at present, two measures to solve the problem mentioned above with the help of the constituency parse tree were proposed. Firstly, node enumeration was used to replace the original span extraction strategy. It avoids the restriction of span length and reduces the number of spans that don't satisfy syntactic rules. Secondly, node sequences are obtained through tree traversal, and the features such as height and path are combined to generate the context representation of the constituency parse trees directly. It avoids the problem of missing structural information caused by the use of word and character sequences only. A lot of experiments were conducted on the dataset of CoNLL 2012 Shared Task, and the proposed model achieves 62.35 average F1 for Chinese and 67.24 average F1 for English, which show that the proposed structural information integration strategy can improve the performance of coreference resolution significantly.

Keywords Coreference resolution, Constituency parse tree, Structural information, Height features, Embedding

1 引言

指代消解是自然语言理解领域的一项关键任务。准确、无歧义的指代消解能促进对篇章语义的整体理解,对于信息抽取、自动摘要、问答系统以及机器翻译等自然语言应用有着极为重要的基础支撑作用。

早期的研究表明,结构化信息对于指代消解非常重要。Hobbs[1]使用句法树进行代词消解,该算法首先为文档中的每个句子建立完全解析树,然后采用广度优先搜索的方法遍历该树,最后根据语法结构中的支配和绑定关系选择合适的名词短语作为先行词。Lappin 等[2]提出针对第三人称代词与反身代词的 RAP 算法,该算法利用 McCord 等[3]提出的槽文法来获得文档的句法结构,并通过手工加权各种语言特征来计算候选先行词的凸显度,经过过滤规则最终确定其先行词。Kong 等[4]提出了基于树核函数的中英文代词消解方法,该方法提出结合中心理论知识、竞争者信息以及驱动谓词的

到稿日期:2019-01-15 返修日期:2019-05-23 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61876118);人工智能应急项目(61751206);国家重点研发计划子课题(2017YFB1002101)
This work was supported by the National Natural Science Foundation of China (61876118), Artificial Intelligence Emergency Project (61751206) and National Key Research and Development Plan Sub-project (2017YFB1002101).
通信作者:孔芳(kongfang@stu.suda.edu.cn)

相关信息来对句法树进行动态扩展，并使用 SVMLight 中提供的卷积树核函数直接进行基于结构化句法树的相似度计算，从而完成指代消解任务。

近几年，随着神经网络的兴起，单词可以表示为传递语义依赖关系的向量[5]，单词之间的依赖关系可以被循环神经网络等结构捕获，同时神经网络具有优异的数据拟合和分类能力，因此越来越多的学者开始将各种神经网络模型应用于指代消解任务。Clark 等[6]采用强化学习，利用其提出的奖励重新调整(Reward Rescaling)方法进行指代消解，并在 CoNLL 2012[7]中文测试集上获得了目前最好的性能；Wu 等[8]提出使用卷积神经网络来对词嵌入进行上下文表示，进而进行表述(Mention)表示，并利用分类器进行最终消解；Lee 等[9]提出了一种基于神经网络的端到端实体指代消解模型，该模型使用双向 LSTM[10]与 Head-finding 注意力机制来进行短语表征，并使用短语排序模型来完成指代消解工作；Lee 等[11]在之前工作的基础上，利用消解过程对短语的表示进行迭代更新，并利用双线性注意力修剪待消解项的候选先行词搜索空间，结合 ELMo[12]在 CoNLL 2012[7]英文测试集上获得了目前已知的最好性能。得益于神经网络优异的表征能力，目前基于神经网络的指代消解模型的性能已大幅领先于传统模型。但是，现有的神经网络模型普遍只关注文本的线性特征，忽略了传统方法中已证明非常有效的结构信息的融入。

融入结构信息最直接的方式便是使用成分句法树。本文以此为出发点，提出了两种融入策略来提升结构信息在模型中的表示。以 Lee 等[9]的模型为基准，对其进行如下改进。

1)提出了一种枚举句法树中结点作为短语的抽取策略。相比于原模型使用的贪婪枚举策略，结点枚举策略避免了前者所受到的长度限制与不符合句法规则的短语集噪音的引入，并隐式地将结构信息融入模型中。

2)利用后序遍历等方式得到树的结点序列，结合结点的高度与路径等特征，直接对成分句法树进行上下文编码，显式地将结构信息融入模型中，以提高后续短语修剪与先行词识别的性能。

2 指代消解的基本框架

本文 Lee 等[9]提出的基于神经网络的 Span-Ranking 模型作为基准模型。该模型可分为表述识别与先行词识别两部分，分别如图 1、图 2 所示。

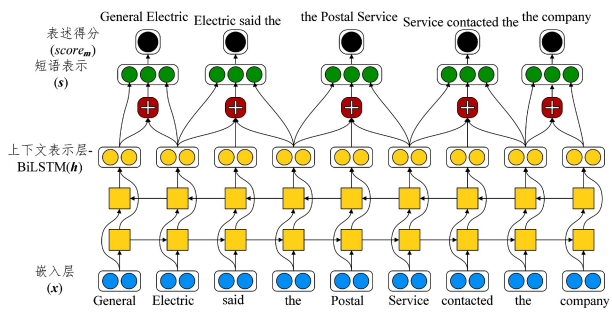

图 1 基准模型(表述识别部分)

Fig. 1 Benchmark model (expression recognition part)

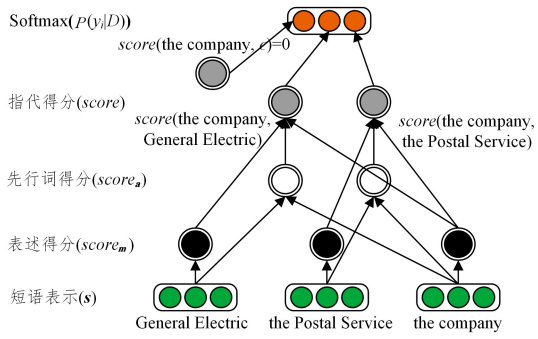

图 2 基准模型(先行词识别部分)

Fig. 2 Benchmark model (precedent recognition part)

对于输入文本 $D=\{w_1,w_2,\cdots,w_{N_D}\}$，$N_D$ 为文本所包含的单词数，我们从中抽取出短语集合 $S=\{s_1,s_2,\cdots,s_n\}$，其中 $s_i=\{w_{b_i},w_{b_i+1},\cdots,w_{e_i}\}$，$b_i$ 与 e_i 分别表示短语 s_i 的开始位置与结束位置的单词的下标，因此 $1\leqslant b_i\leqslant e_i\leqslant N_D$，$s_i$ 的宽度为 e_i-b_i+1。

Lee 等[9]所提模型的核心思想是利用嵌入层与上下文表示层得到的信息，结合 Head-Attention 机制对短语进行表征，然后通过前馈神经网络进行打分，并根据其得分进行修剪，最后利用其保留下来的短语集合，结合 Ranking 机制进行消解，得到最终的消解结果。

1)嵌入层。对于 $\forall w_i\in D$，通过字、词嵌入，得到 $\boldsymbol{x}_i=[\boldsymbol{w}_i,\boldsymbol{c}_i]\in\mathbb{R}^{d_x}$，其中 $\boldsymbol{w}_i\in\mathbb{R}^{d_w}$ 表示 w_i 的词嵌入向量，我们利用 Character CNN[13-14]或者 Character LSTM[15]得到 w_i 对应的字嵌入向量 $\boldsymbol{c}_i\in\mathbb{R}^{d_c}$，$d_x=d_w+d_c$，进而得到 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{N_D}]\in\mathbb{R}^{N_D\times d_x}$。

2)上下文表示层。给定 $\boldsymbol{x}_i\in\mathbb{R}^{d_x}$，我们使用双向 *LSTM* 作为表示层来获得对应的上下文表示 $\boldsymbol{h}_i\in\mathbb{R}^{d_h}$。

3)短语表示层与修剪层。给定短语 $s_i=\{w_{b_i},w_{b_i+1},\cdots,w_{e_i}\}$，我们设定该短语的向量表示为 $\boldsymbol{s}_i=[\boldsymbol{h}_{b_i},\boldsymbol{h}_{e_i},\hat{\boldsymbol{x}}_i,\boldsymbol{f}_i]\in\mathbb{R}^{d_s}$，其中 $\boldsymbol{f}_i$ 表示额外的向量特征(此处为短语的宽度特征)，通过式(1)－式(3)计算 $\hat{\boldsymbol{x}}_i$：

$$\alpha_t=FFNN_a(\boldsymbol{h}_t) \tag{1}$$

$$a_{i,t}=\frac{\exp(\alpha_t)}{\sum_{k=b_i}^{e_i}\exp(\alpha_t)} \tag{2}$$

$$\hat{\boldsymbol{x}}_i=\sum_{t=b_i}^{e_i}a_{i,t}\cdot\boldsymbol{x}_t \tag{3}$$

得到所有短语的向量表示后，我们使用前馈神经网络对其进行打分，如式(4)所示：

$$score_m(i)=FFNN_m(\boldsymbol{s}_i) \tag{4}$$

然后取得分最高的前 k 个短语作为待消解项集合 $A=\{s_1',s_2',\cdots,s_k'\}$ 参与后续的消解操作。

4)消解层。给定待消解项 s_i 与其候选先行词 s_j，其中 $0\leqslant j<i\leqslant k$，$s_i\in A$，$s_j\in\{\varepsilon\}+A_{i-1}$。当 $j=0$ 时，$s_j=\varepsilon$，表示 s_i 没有潜在的先行词存在。同样，我们使用前馈神经网络来获得 s_i 与 s_j 的先行词得分，如式(5)所示：

$$score_a(i,j)=FFNN_a([\boldsymbol{s}_i,\boldsymbol{s}_j,\boldsymbol{s}_i\odot\boldsymbol{s}_j,\boldsymbol{f}_{i,j}]) \tag{5}$$

其中，$\boldsymbol{f}_{i,j}$ 编码 s_i 与 s_j 之间的讲述者、文档类型与距离特征。

进一步地,通过式(6)可以得到 s_i 与 s_j 之间的指代得分为:

$$score(i,j)=\begin{cases}0, & s_j=\varepsilon\\ score_m(i)+score_m(j)+score_a(i,j), & s_j\neq\varepsilon\end{cases} \tag{6}$$

最终,我们取 s_i 的先行词集合中与 s_i 指代得分最高的 s_{j^*} 作为 s_i 的最终消解结果,其中 $j^*=\arg\max_j score(i,j)$。

3　成分句法树的使用

本文对成分句法的使用主要表现在以下 3 个方面:1)将句法树的所有结点作为短语集合 S,替换掉原来的贪婪枚举的抽取策略;2)按照一定的遍历规则(如先序遍历、后序遍历、层次遍历等)对句法树进行遍历,得到结点序列,并用其替换掉原有的单词序列输入;3)从结点中提取典型的结点特征,以丰富结点序列的表示。

图 3(a)所示的成分句法树所表示的句子为"为什么/这样/讲/呢/?",其中与虚线连接的"单词结点"不作为树的一部分,只是作为注解标识,即对应该树的叶子结点集合为{AD,AD,VV,SP,PU}。我们以后序遍历为例,得到遍历后的结点序列为{AD,ADVP,AD,ADVP,VV,VP,VP,IP,SP,PU,CP,TOP},其顺序亦在图 3(b)中标出。针对树中的每一个结点,我们均可以将其视为一个短语,例如"VP-7"可被视为"为什么/这样/讲",即,"IP-8"亦可代表同样的短语,在经过去重后,我们可以得到结点序列对应的短语序列为{[1,1],[2,2],[3,3],[1,3],[4,4],[5,5],[1,5]}。同时,针对结点序列中的每个结点,我们可以抽取其高度、类别标签等特征来表达该结点所存储的信息。

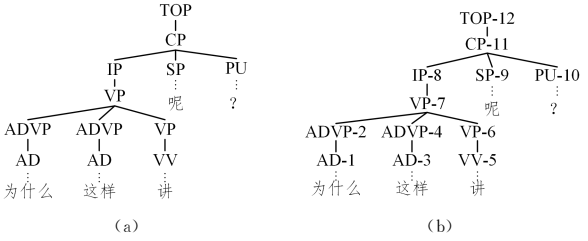

图 3　成分句法树及其遍历

Fig. 3　Constituency parse tree and its traversal

我们提出用图 4 所示嵌入方法将成分句法树所存储的信息尽可能地加入神经网络中,以丰富原有的上下文表征,进而提升指代消解的性能。

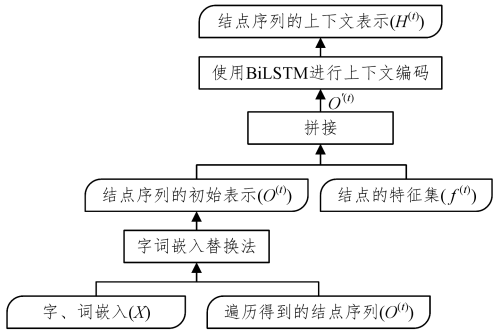

图 4　结构嵌入的主要流程

Fig. 4　Main process of structural embedding

3.1　结构信息的加入

给定成分句法树 t,假设我们使用某种遍历方式(先序遍历、后序遍历或者层次遍历)得到树 t 的结点序列 $O^{(t)}=\{o_1^{(t)},o_2^{(t)},\cdots,o_{n_t}^{(t)}\}$,其中 n_t 表示 t 的所有结点的数目(不包含"单词结点")。

为了得到 $O^{(t)}$ 的向量表示,我们首先使用零向量对其进行初始化:$\boldsymbol{O}^{(t)}=[\mathbf{0}_1^{(t)},\mathbf{0}_2^{(t)},\cdots,\mathbf{0}_{n_t}^{(t)}]\in\mathbb{R}^{n_t\times d_x}$;然后将序列中与"单词结点"连接结点对应位置的向量表示替换为对应单词的字词嵌入表示。我们称该方法为"字词嵌入替换法"。

以图 3 所示的成分句法树为例,我们对其后序遍历序列进行零向量初始化,得到 $\boldsymbol{O}=[\mathbf{0},\mathbf{0},\cdots,\mathbf{0}]\in\mathbb{R}^{12\times d_x}$,然后替换上对应位置的字词嵌入向量,得到 $\boldsymbol{O}=[\boldsymbol{x}_1,\mathbf{0},\boldsymbol{x}_2,\mathbf{0},\boldsymbol{x}_3,\mathbf{0},\mathbf{0},\mathbf{0},\boldsymbol{x}_4,\boldsymbol{x}_5,\mathbf{0},\mathbf{0}]\in\mathbb{R}^{12\times d_x}$。最后将其作为上下文表示层的输入,使得上下文表示层能够学习到其潜在的树形结构表示,得到对应 $\boldsymbol{O}^{(t)}$ 的上下文表示:

$$\boldsymbol{H}^{(t)}=[\boldsymbol{h}_1^{(t)},\boldsymbol{h}_2^{(t)},\cdots,\boldsymbol{h}_{n_t}^{(t)}]\in\mathbb{R}^{n_t\times d_h}$$

对文档 D 的所有句子对应的成分句法树应用相同的操作,最终得到文档 D 的新的上下文表示 $\boldsymbol{H}\in\mathbb{R}^{N_T\times d_h}$,$N_T$ 表示 D 中所有成分句法树的结点的总和,即 $N_T=n_1+n_2+\cdots+n_T$,T 表示 D 中成分句法树的总数,也即句子的数量。

3.2　特征的加入

相较于 $\boldsymbol{X}\in\mathbb{R}^{N_D\times d_x}$ 而言,使用 3.1 节中提到的替换方法得到的 $\boldsymbol{O}\in\mathbb{R}^{N_T\times d_1}$ 只增加了隐式的结构信息。本节使用结点的高度显式地表征树的结构,同时利用类别标签特征丰富 $\boldsymbol{O}$ 的表示。对于 $\forall o_i\in O$,我们可以得到其高度 $height$、类别标签 $label$,对其进行嵌入并与 o_i 的原向量表示 $\boldsymbol{o}_i$ 进行拼接,从而得到新的向量表示:

$$\boldsymbol{o}_i'=[\boldsymbol{o}_i,\boldsymbol{f}_{height,i},\boldsymbol{f}_{label,i}]\in\mathbb{R}^{d_x+2d_f}$$

句法分析难免会引入一些错误,这些错误可能会影响系统的性能。我们通过实验验证了错误的类别标签数据会对系统的性能造成严重损害,因此,为了提升系统的容错率,我们提出使用结点所在树的根结点到该结点的路径 $path$ 来替换原有的类别标签特征。具体地,对于 $\forall o_i\in O$:

$$path_i=\{label_{root},\cdots,label_i\}$$

$path_i$ 的长度即为 o_i 的深度 $depth_i$。注意,这里的"深度"与"高度"的定义不同:结点 o_i 的高度是指从 o_i 到叶结点最长的一条简单路径上边的数目,而深度是指树的根结点 $root$ 到结点 o_i 的一条简单路径上边的数目。因此,两者具有一定的互补性。

然后,通过类似 Character LSTM 的方式,先对 $path_i$ 进行嵌入表示:

$$\boldsymbol{path}_i=[\boldsymbol{f}_{label,root},\cdots,\boldsymbol{f}_{label,i}]\in\mathbb{R}^{depth_i\times d_f}$$

将其作为双向 LSTM 的输入,得到:

$$\boldsymbol{h}_{label,i}=BiLSTM(\boldsymbol{f}_{label,i})=[\overrightarrow{\boldsymbol{h}}_{label,i},\overleftarrow{\boldsymbol{h}}_{label,i}]$$

最终,我们取 $\boldsymbol{f}_{path,i}=[\overrightarrow{\boldsymbol{h}}_{label,i},\overleftarrow{\boldsymbol{h}}_{label,root}]$作为路径特征的最终表示,从而得到结点 o_i 的最终表示:

$$\boldsymbol{o}_i''=[\boldsymbol{o}_i,\boldsymbol{f}_{height,i},\boldsymbol{f}_{path,i}]\in\mathbb{R}^{d_x+2d_f}$$

一方面，LSTM 通过门控机制对输入的信息进行了有效过滤和保留；另一方面，高度与路径特征的加入进一步加深了结构信息的影响。

综合遍历后的结点序列与高度、路径特征，我们从“横向”与“纵向”两个方向对成分句法树所存储的信息进行了编码。

4 实验与结果分析

前文介绍了端到端的实体指代消解模型，讨论了成分句法树如何编码加入到神经网络中。本节将使用 CoNLL 2012[7] 数据集进行实验，并对相关结果进行分析与讨论。

4.1 实验语料

本文的实验均构建在 CoNLL 2012 Shared Task[7] 的数据集上，数据集规模如表 1 所列。CoNLL 2012 Shared Task 是由计算自然语言学习会议（Conference on Computational Natural Language Learning，CoNLL）于 2012 年举办的针对英文、中文和阿拉伯文的多语言指代消解评测，用于研究不同语言指代消解研究的差别与共性，其数据集基于 OntoNotes 5.0 语料构建，并融合了多种语言学标注信息。该评测的出现使得研究人员在真正意义上有了可以相互比较的平台，极大地促进了指代消解的发展；其评测语料也成为了指代消解研究中使用得最为广泛的语料之一。

表 1 CoNLL 2012 数据集的规模

Table 1 Size of CoNLL 2012 dataset

语言	类型	训练集	验证集	测试集	总计
英文	词数	1.3×10^6	160×10^3	170×10^3	1.6×10^6
	文档数	2.8×10^3	0.3×10^3	0.3×10^3	3.4×10^3
	实体链数	35.1×10^3	4.5×10^3	4.5×10^3	44.2×10^3
	指代关系数	120.4×10^3	14.6×10^3	15.2×10^3	150.2×10^3
	表述数	155.5×10^3	19.1×10^3	19.7×10^3	194.4×10^3
中文	词数	750×10^3	110×10^3	90×10^3	950×10^3
	文档数	1.8×10^3	0.2×10^3	0.2×10^3	2.2×10^3
	实体链数	28.2×10^3	3.8×10^3	3.5×10^3	35.6×10^3
	指代关系数	74.5×10^3	10.3×10^3	9.2×10^3	94.1×10^3
	表述数	102.8×10^3	14.1×10^3	12.8×10^3	129.8×10^3

该数据集包含中文、英文与阿拉伯文 3 种语言，中文与英文大约有 100 万字，语料来源于新闻、杂志、网页信息和脱口秀节目等，提供了词性标注、句法标注、命名实体标注与指代链标注等多种标注信息，这些标注信息均可以很好地应用于指代消解。

4.2 实验设置

实验数据采用 4.1 节中提到的 CoNLL 2012 数据集，若无特别说明，训练、验证和测试中使用的成分句法树均源于自动标注数据。实验结果的评价标准由 MUC[16]，B^3[17]，$CEAF_{\phi4}$[18] 及 3 者 F1 值的平均值（即 CoNLL 2012 Shared Task[7] 的官方评价标准）构成。

实验所用平台通过 PyTorch 1.0 构建，中英文平台的主要参数差异如表 2 所列，其他相关参数为：Character LSTM 使用单向 LSTM，隐藏层维度为 100；前馈神经网络为 3 层，使用 ReLU 激活函数[19]，最后一层的输出维度为 1；字词嵌入的 *dropout*[20] 为 0.5，LSTM 隐层输出及特征向量的 *dropout* 为 0.2；使用 Adam 优化器[21]；其他参数与文献[9]中的相同。

表 2 模型的参数设置

Table 2 Hyperparameters of model

名称	中文	英文
预训练的词向量	Polyglot[22]，64 维	Glove[23]，300 维
字嵌入方式	Character LSTM	Character CNN
d_w	100 维	300 维
d_c	100 维	100 维
d_n	200 维	400 维
d_f	20 维	20 维

4.3 实验分析

4.3.1 短语集合的影响

使用从句法树抽取的短语集合来替换在基准系统上使用贪婪枚举得到的短语集合，实验结果如表 3 所列。其中，PSD（Pruned Span Detection）表示修剪后的短语集合的评价，MD（Mention Detection）表示最终消解产生的表述集合的评价。

表 3 短语集合的影响

Table 3 Influence of span set

（单位：%）

数据集	系统	PSD			MD			CoNLL
		P	*R*	*F*1	*P*	*R*	*F*1	Avg. *F*1
CoNLL 2012 中文测试集	基准系统	29.24	84.54	43.45	85.00	49.37	62.46	58.41
	十句法树短语集合	29.59	85.56	43.98	85.71	49.71	62.93	60.33
CoNLL 2012 英文测试集	基准系统	26.93	92.61	41.73	88.56	62.14	73.03	65.39
	十句法树短语集合	27.29	93.85	42.28	88.29	63.49	73.86	66.07

从表 3 可以看出，从句法树上抽取出的短语集合，相对于贪婪枚举得到的短语集合，在准确率和召回率上均有提升，最终的消解性能在中文上相对提升了 1.92，在英文上相对提升了 0.68。我们认为此提升一方面来源于成分句法树的合理性，另一方面则是由于贪婪枚举所受到的最大长度限制。相对于贪婪枚举，从成分句法树上抽取短语，一方面不仅大大减少了冗余与无用短语的数量，进而减少了噪音的干扰，同时也减少了模型的计算量与资源需求；另一方面，避开了最大长度的限制，使得较大长度的短语能够参与到训练中，提高了短语与表述的召回率，进而提升了系统的性能。

4.3.2 句法树结构信息的影响

不同的遍历方式会产生不同的结点序列，从而生成不同的结构信息。本节对不同的遍历方式进行了实验，结果如表 4 所列。

表 4 结构信息的影响

Table 4 Influence of structural information

（单位：%）

数据集	遍历方式	Avg. *F*1	Δ
CoNLL 2012 中文测试集	基准系统	58.41	
	+先序遍历结点序列	60.55	+2.14
	+后序遍历结点序列	60.73	+2.32
	+层次遍历结点序列	51.26	−7.16
CoNLL 2012 英文测试集	基准系统	65.39	
	+先序遍历结点序列	65.95	+0.57
	+后序遍历结点序列	65.84	+0.45
	+层次遍历结点序列	62.90	−2.48

从表 4 可以看出，先序遍历及后序遍历产生的结点序列所编码的结构信息加入到系统后，均对系统的性能产生了积极影响，其中后序遍历在中文上表现得最好，先序遍历在英文

上表现得最好。但从层次遍历中产生的信息在中英文平台上均产生了极大的负面影响,我们认为其原因有以下两个方面:1)相对于先序与后序遍历,层次遍历产生的结点序列中的"单词"无法保证其在文本中的原有顺序,表示层显然无法"理解""乱序"的文本;2)叶子结点均处于树中较低的层次,这会导致层次遍历产生的结点序列经由"字词嵌入替换法"初始化后,前面的大部分均为零向量,从而使得表示层无法有效学习。

4.3.3　句法树特征的影响

不同的特征对系统的贡献不尽相同,本节对不同的特征选择进行了实验,实验结果如表 5 所列。

表 5　特征的影响

Table 5　Influence of features

(单位:%)

数据集	特征选择	Avg. $F1$	Δ
CoNLL 2012 中文测试集	基准系统(+后序遍历)	60.73	
	+结点的高度特征	60.92	+0.19
	+结点的类别特征	60.11	−0.62
	+结点的路径特征	61.45	+0.72
CoNLL 2012 英文测试集	基准系统(+先序遍历)	65.95	
	+结点的高度特征	66.06	+0.11
	+结点的类别特征	65.88	−0.07
	+结点的路径特征	66.26	+0.31

高度特征显式地指明了结点之间的层次结构。由表 5 可以看出,高度特征均在中英文上产生了正向作用,但相对于基准实验来说提升较小。我们认为提升较小的原因是表示层已有足够的能力对具有隐式结构的序列进行编码及上下文表示,加入高度后只是锦上添花。

与高度特征相反,类别标签特征反而产生了较大的负面作用,我们将其归结为标注错误以及神经网络对标签数据的容错性小于对结构信息的容错性。为了验证这个猜想,我们使用 CoNLL 2012 提供的人工标注数据集(记为"GOLD")来替换掉自动标注数据集,并使用同样的参数设置重新进行了表 5 中的所有实验,结果如表 6 所列。

表 6　标注错误的影响

Table 6　Influence of tagging errors

(单位:%)

数据集	特征选择	Avg. $F1$	Δ
CoNLL 2012 中文测试集 (GOLD)	基准系统(+后序遍历)	67.10	
	+结点的高度特征	67.89	+0.79
	+结点的类别特征	67.65	+0.55
	+结点的路径特征	71.97	+4.87
CoNLL 2012 英文测试集 (GOLD)	基准系统(+先序遍历)	67.51	
	+结点的高度特征	67.79	+0.28
	+结点的类别特征	67.71	+0.20
	+结点的路径特征	69.04	+1.53

从表 6 可以看出,在排除了标注错误的因素,加入了结点的高度或者类别标签特征后,系统的性能均有明显提升。此外,高度特征的贡献依然高于类别标签特征,这也从侧面反映了结构信息对性能的提升尤为重要。

为了减弱标注错误对性能的影响,同时提高类别标签特征的表征能力,我们使用路径特征替换类别标签特征,结合 LSTM 强大的表示能力,在加入该特征后,无论在自动标注数据还是人工标注数据上,系统的性能均获得了显著提升。另外,由于路径特征隐式地包含了结点的深度信息,而高度与深度又具有一定的互补性,因此高度与路径特征均将加入最终的模型中。

最后,结合从句法树中提取的短语集合、结构信息和特征进行实验,实验结果如表 7 所列。

表 7　CoNLL 2012 中英文测试集上的实验结果

Table 7　Results on test sets from CoNLL-2012 shared task

(单位:%)

数据集	模型	MUC			B^3			$CEAF_{\phi4}$			CoNLL
		P	R	$F1$	P	R	$F1$	P	R	$F1$	Avg. $F1$
CoNLL 2012 中文测试集	Clark & Manning[24]	73.85	65.42	69.38	67.53	56.41	61.47	62.84	57.62	60.23	63.66
	Clark & Manning[6]	73.64	65.62	69.40	67.48	56.94	61.76	62.46	58.60	60.47	**63.88**
	Baseline (replicate)	73.67	59.26	65.68	66.87	49.01	56.56	60.59	47.08	52.99	58.41
	Our Model	74.82	64.06	69.02	67.90	54.20	60.28	63.28	53.11	57.75	**62.35**
CoNLL 2012 英文测试集	Clark & Manning[6]	79.19	70.44	74.56	69.93	57.99	63.40	63.46	55.52	59.23	65.73
	Lee[9]	78.40	73.40	75.80	68.60	61.80	65.00	62.70	59.00	60.80	67.20[1)]
	Lee[11]	81.40	79.50	80.40	72.20	69.50	70.80	68.20	67.10	67.60	**73.00**
	Baseline (replicate)	78.04	71.05	74.38	68.35	58.27	62.91	61.96	56.09	58.88	65.39
	Our Model	78.82	72.75	75.66	68.86	61.26	64.84	63.31	59.23	61.20	**67.24**

从表 7 可以看出,对于中文,最终模型相对于基准模型提升了 3.94 个点,相较于 Clark 等[6]的模型仍有 1 个点左右的差距,其背后的强化学习技术值得我们学习与借鉴。对于英文,最终模型相较于基准模型提升了 1.85 个点,相较于 Lee 等的最新模型[11]相差近 6 个点,这个最新的模型与 Lee 等的工作[9]同样相差 6 个点左右,但这 6 个点中 3 个点的提升来源于 ELMo[12],另 3 个点的提升来源于参数的调整,其对基准模型的改进在本质上只提升了不到 1 个点,但其工作同样值得我们学习。未来,我们将探索使用强化学习、预训练语言模型(即 ELMo[12]或者 BERT[25]等)等最新技术,以进一步提升模型的性能。

结束语　本文借鉴 Lee 等提出的指代消解框架,提出了将成分句法树提供的短语集合、结构信息与特征经过嵌入加入到系统后,在 CoNLL 2012 中英文数据集上均获得了不错的效果,说明了语言学的先验知识对模型性能提升的重要性。未来我们将不断尝试将句法、语法及语义信息等加入神经指代系统中,以在保证一定通用性的同时,进一步提升其性能和可用性,并探索将强化学习、预训练语言模型等新技术加入模型中。

1) Lee 等[9,11]报告的结果只精确到小数点后 1 位,小数点后第 2 位的 0 均为我们补充的默认值。

参 考 文 献

[1] HOBBS J R. Resolving pronoun references[J]. Lingua, 1978, 44(4): 311-338.

[2] LAPPIN S, LEASS H J. An algorithm for pronominal anaphora resolution[J]. Computational linguistics, 1994, 20(4): 535-561.

[3] MCCORD M C. Slot grammar [M]// Natural Language and Logic. Berlin: Springer, 1990: 118-145.

[4] KONG F, ZHOU G D. Pronoun Resolution in English and Chinese Languages Based on Tree Kernel[J]. Journal of Software, 2012, 23(5): 1085-1099.

[5] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality [C]// Advances in Neural Information Processing Systems. Lake Tahoe: NIPS, 2013: 3111-3119.

[6] CLARK K, MANNING C D. Deep Reinforcement Learning for Mention-Ranking Coreference Models[C]// Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin: EMNLP, 2016: 2256-2262.

[7] PRADHAN S, MOSCHITTI A, XUE N, et al. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes[C]// Joint Conference on EMNLP and CoNLL-Shared Task. Jeju Island: ACL, 2012: 1-40.

[8] WU J L, MA W Y. A deep learning framework for coreference resolution based on convolutional neural network[C]// 2017 IEEE 11th International Conference on Semantic Computing (ICSC). San Diego: IEEE, 2017: 61-64.

[9] LEE K, HE L, LEWIS M, et al. End-to-end Neural Coreference Resolution[C]// Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen: ACL, 2017: 188-197.

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

[11] LEE K, HE L, ZETTLEMOYER L. Higher-Order Coreference Resolution with Coarse-to-Fine Inference[C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans: ACL, 2018, 2: 687-692.

[12] PETERS M, NEUMANN M, IYYER M, et al. Deep Contextualized Word Representations[C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans: ACL, 2018: 2227-2237.

[13] LIANG D, XU W, ZHAO Y. Combining word-level and character-level representations for relation classification of informal text[C]// Proceedings of the 2nd Workshop on Representation Learning for NLP. Vancouver: ACL, 2017: 43-47.

[14] ZHANG X, ZHAO J, LECUN Y. Character-level convolutional networks for text classification[C]// Advances in Neural Information Processing Systems. Montreal: NIPS, 2015: 649-657.

[15] LING W, DYER C, BLACK A W, et al. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation[C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: ACL, 2015: 1520-1530.

[16] VILAIN M, BURGER J, ABERDEEN J, et al. A model-theoretic coreference scoring scheme[C]// Proceedings of the 6th Conference on Message Understanding. Columbia: ACL, 1995: 45-52.

[17] BAGGA A, BALDWIN B. Algorithms for scoring coreference chains[C]// The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference. Granada: LREC, 1998, 1: 563-566.

[18] LUO X. On coreference resolution performance metrics [C]// Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. Vancouver: ACL, 2005: 25-32.

[19] NAIR V, HINTON G E. Rectified linear units improve restricted boltzmann machines[C]// Proceedings of the 27th International Conference on International Conference on Machine Learning. Haifa: Omni press, 2010: 807-814.

[20] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting [J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[21] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. arXiv: 1412.6980, 2014.

[22] AL-RFOU R, PEROZZI B, SKIENA S. Polyglot: Distributed Word Representations for Multilingual NLP [C]// Proceedings of the Seventeenth Conference on Computational Natural Language Learning. Sofia: ACL, 2013: 183-192.

[23] PENNINGTON J, SOCHER R, MANNING C. Glove: Global vectors for word representation[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha: ACL, 2014: 1532-1543.

[24] CLARK K, MANNING C D. Improving Coreference Resolution by Learning Entity-Level Distributed Representations[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin: ACL, 2016: 643-653.

[25] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv: 1810.04805, 2018.

FU Jian, born in 1994, postgraduate student. His main research interest include coreference resolution and natural language processing.

KONG Fang, born in 1977, doctor. Her main research interest include machine learning, natural language processing, and text analysis.