编者按: 中国中文信息学会于 2008 年 11 月在北京成功地召开了“第四届全国信息检索与内容安全学术会议(NCIRCS-2008)”。会议的程序委员会向本刊推荐了 25 篇论文,并经作者仔细修改,编辑部得到授权,将在 2009 年第二、三期发表,以飨读者。

文章编号: 1003-0077(2009)02-0018-05

基于最大熵的依存句法分析

辛 霄,范士喜,王 轩,王晓龙

(哈尔滨工业大学 深圳研究生院 智能计算研究中心 广东 深圳 518055)

摘 要: 该文提出并比较了三种基于最大熵模型的依存句法分析算法,其中最大生成树(MST)算法取得了最好的效果。MST 算法的目标是在一个带有权重的有向图中寻找一棵最大的生成树。有向图的每条边都对应于一个句法依存关系,边的权重通过最大熵模型获得。训练和测试数据来源于 CoNLL2008 Share Task 的公用语料。预测的 F1 值在 WSJ 和 Brown 两个测试集上分别达到 87.42% 和 80.8%,在参加评测单位中排名第 6。

关键词: 计算机应用; 中文信息处理; 句法分析; 最大生成树; 最大熵
中图分类号: TP391 文献标识码: A

Dependency Parsing Based on Maximum Entropy Model

XIN Xiao, FAN Shi-xi, WANG Xuan, WANG Xiao-long
(Intelligent Computing Research Center, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guangdong 518055, China)

Abstract: This paper presents three algorithms for dependency parsing based on the Maximum Entropy Models. The Maximum Spanning Tree (MST) algorithm achieves the best result. The target of MST is to find a Maximum Spanning Tree in a directed graph. Each edge of the directed graph corresponds to a dependency relation of the dependency parser, and the weights of the edges are obtained by using a Maximum Entropy Model. The training and test data sets are the CoNLL2008 share task corpora. The system achieves F1 scores of 87.42 and 80.8 for WSJ and Brown test data respectively, ranking sixth among all the competition teams.

Key words: computer application; Chinese information processing; parsing; maximum spanning tree; maximum entropy

1 引言

句法分析一直是自然语言处理的核心问题之一。法国语言学家 Lucien Tesni re (特思尼耶尔)在1959 年提出依存句法分析的理论[1]。因依存句法分析比结构化句法分析更容易处理,近年受到了人们的广泛关注[2]。在依存句法中,每个单词会以一定的关系依存于并且只能依存于句子中的另外一个单词或者是虚根(ROOT)。依存句法分析可以分解为两个子任务——识别(Identification)与分类(Classification),其中: 识别子任务是一个二分类问题,判断两个单词之间是否存在依存关系;分类子任务是一个多分类问题,确定两个单词之间的依存关系,CoNLL2008 Share Task 提供的语料中依存关系共有 69 种。依存句法可以用一棵依存关系树来

收稿日期: 2008-08-16 定稿日期: 2008-10-30
基金项目: 自然科学基金资助项目(60435020, 90612005);国家 863 高科技计划资助项目(2006AA01Z197)
作者简介: 辛霄(1984—),男,硕士生,主要研究方向为中文智能问答;范士喜(1978—),男,博士生,主要研究方向为中文智能问答;王轩(1969—),男,博士,教授,博导,主要研究方向为人工智能,计算机网络安全,计算语言学。

表示, 如图 1 例子所示, 在“全国人民共同关心汶川”这句话中, 每个词都依存于一个其他的词, 其中“关心”是句子的根节点依存于虚根(ROOT)。依存关系也标注在图上, 例如: “全国”依存于“人民”, 依存关系为定中关系; “人民”依存于关心, 依存关系为主谓关系。

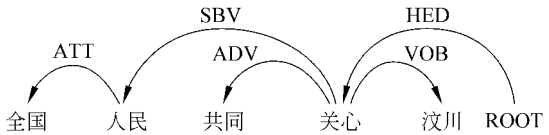

图 1　依存句法分析例子

国内外众多研究者对依存句法分析作了大量的研究工作。目前统计方法是依存句法分析的主流。Ryan McDonald 等采用大间隔的在线学习的方法来训练一个依存关系概率模型[3]。Yamada 和 Matsumotoy 以及 Nivre 和 Scholz 提出决策式依存分析方法[4-5]。决策式依存分析方法是将分析过程看成是一步步作用于输入句子之上的分析动作的序列, 逐步形成一个完整依存句法树。中国科学院段湘煜等提出基于动作的多阶段算法[8], 南京大学张亮等提出基于模式匹配的句法生成方法[9], 另外还有基于规则的方法[10] 等。

哈尔滨工业大学深圳研究生院智能计算研究中心参加了 2008 年 CoNLL share task 的评测比赛。评测的目标是在统一的依存关系框架下对英文语句中的句法关系和语义关系进行识别和标注。我们的参赛系统可以分为三个部分: 1. 句法分析, 生成句法依存树。2. 识别句子中心词。3. 通过包括句法依存关系在内的一些特征, 对识别出的中心词进行语义角色标注。其中句法分析部分是中心词识别和语义角色标注的基础, 本文介绍句法分析的算法和模型。

本文后续内容组织如下: 第二部分介绍了三种依存句法分析算法。第三部分介绍了基于最大熵模型的有向边权重计算模型。第四部分介绍实验。

2　依存句法分析算法

在接下来的论文中用 $X = x_1, x_2 \cdots x_n$ 表示一个句子, 其中 x_i 表示句子的第 i 个单词。句子的语法树 $Y = \{(x_i -> x_j)\}$ 表示一个有向边的集合, 其中边 $(x_i -> x_j)$ 表示单词 x_i 依存于单词 x_j 或者虚拟根 $(ROOT)$。为了描述方便也称 x_j 为 x_i 的父节点, x_i 为 x_j 的儿子节点。每条边都具有一定的权重, 这个权重可以表示为:

$$w(x_i -> x_j) = w \cdot f(i,j) \qquad (1)$$

公式(1)中, $f(i,j)$ 是根据边 $(x_i -> x_j)$ 提取的特征向量, w 是特征向量的权重。这个点乘公式给出了一条有向边的概率(权重)的计算方法。一棵依存句法树的权值可以通过计算这个句法树中所有边的权重和来获得:

$$P(Y) = \sum_{(x_i -> x_j) \in Y} w(x_i -> x_j) \qquad (2)$$

如果把每个单词作为有向图的一个顶点, 每个单词之间都存在一个带有权重的有向边。得到一个有向图: $G = (V, E)$, $V = \{x_0 = ROOT, x_1, \cdots x_n\}$, $E = \{(i -> j): i \neq j, i \in [1: n], j \in [0: n]\}$。其中 x_i 对应于第 i 个单词, E 是有向边的集合。设 T 为句法树的集合, 最大生成树算法就是在 T 中找到一棵最大的生成树:

$$Y_* = \arg\max_{Y \subset T} P(Y)$$
$$\text{s.t.} \ |Y| = n; \quad \forall x_i, \exists x_j, x_j \neq x_i, (x_i -> x_j) \in Y \qquad (3)$$

本文提出三种依存句法分析算法, 自顶向下算法, 自底向上算法, 最大生成树算法, 下面分别介绍。

2.1　自顶向下算法(Up2Down)

自顶向下算法的思想是从根节点向下扩展寻找生成树。

第一步: 初始化子孙集合 $child = \{x_1, \cdots x_n\}$
初始化祖先集合 $parent = \{\}$
初始化生成树 $Y = \{\}$

第二步: 判断根节点 p。$p = \arg\max_{x_i \in child} w(x_i -> ROOT)$
将 p 从子孙集合中移出, $child = child - \{p\}$
将 p 加入祖先集合 $parent = parent + \{p\}$
将边 $(p -> ROOT)$ 加入生成树 $Y = Y + \{(p -> ROOT)\}$

第三步: 循环直到子孙节点集合为空。
查找从子孙节点指向祖先节点的最大的边。
$(x_i^*, x_j^*) = \arg\max_{x_i \in child, x_j \in parent} w(x_i -> x_j)$
将 x_i^* 从子孙集合中移出, $child = child - \{x_i^*\}$
将 x_j^* 加入祖先集合 $parent = parent + \{x_j^*\}$
将边 $(x_i^* -> x_j^*)$ 加入生成树 $Y = Y + \{(x_i^* -> x_j^*)\}$

2.2　自底向上算法(Down2Up)

自底向上算法的思想是从叶子节点向上扩展寻找生成树。已经找到父节点的节点不会再成为其他

节点的父节点。

第一步: 初始化子孙集合 $child=\{x_1, \cdots x_n\}$
初始化生成树 $Y=\{\}$
第二步: 循环直到子孙集合只剩下一个节点。
查找子孙集合中最大的边。$(x_i^*, x_j^*) = \arg\max_{x_i \in child, x_j \in child} w(x_i -> x_j)$
将 x_i^* 从子孙集合中移出, $child = child - \{x_i^*\}$
将边$(x_i^* -> x_j^*)$加入生成树 $Y= Y+ \{(x_i^* -> x_j^*)\}$
第三步: 将最后一个节点作为生成树的根节点
$Y= Y+ \{(p -> ROOT)\}$

2.3 最大生成树算法

最大生成树算法是一个 $O(n^3)$ 算法, 在保证句法树没有环路的前提下, 每次在整个有向图中选择一个权重最大的边加入到句法树中。

第一步: 初始化子孙集合 $child=\{x_1, \cdots x_n\}$
初始化所有节点集合 $All=\{x_1, \cdots x_n\}$
初始化生成树 $Y=\{\}$
第二步: 循环直到子孙集合只有一个节点。
查找从 $child$ 集合出发到达 All 集合中最大的边, $(x_i^*, x_j^*) = \arg\max_{x_i \in child, x_j \in All} w(x_i -> x_j)$
(如果增加边$(x_i^* -> x_j^*)$到 Y 中会产生环路, 则放弃这条边)
将 x_i^* 从子孙集合中移出, $child = child - \{x_i^*\}$
将边$(x_i^* -> x_j^*)$加入生成树 $Y= Y+ \{(x_i^* -> x_j^*)\}$
第三步: 将最后一个节点作为树的根节点。$Y= Y+ \{(p -> ROOT)\}$

3 句法依存关系权重模型

3.1 最大熵模型简介

最大熵属于辨识模型, 是 Adam Berger, Stephen Della Pietra 和 Vincent Della Pietra 在1996年提出的[6]。最大熵模型能够满足所有已知的约束, 对未知的信息不做任何假设。最大熵中的约束是通过特征函数来实现的, 特征函数的使用也解决了N元文法和HMM等模型无法处理的长距离依存问题。特征函数是一个二值函数, 如公式(4), 其中 $C(X)$ 为约束函数。

$$f_i(c, y_i) = \begin{cases} 1, & y = y_i \,\&\, c = C_i(X) \\ 0 & \end{cases} \quad (4)$$

特征函数的真实期望可以通过统计的方法来获得:

$$P(f) = \sum_{x,y} P(x, y) f(x, y) \quad (5)$$

特征函数的期望值在模型中的期望值有如下表示:

$$P(f) = \sum_{x,y} P(x) P(y \mid x) f(x, y) \quad (6)$$

针对每一个特征函数为了使模型满足所有已知的假设, 特征函数的真实期望和模型期望必须相等。这样每一个特征就对应着一个等式约束, 模型必须满足所有特征等式的约束。

$$P(f) = P(f) \quad (7)$$

对于任意一个满足以上对一个模型 $P(Y|X)$ 能够计算模型在数据集上的条件熵。这里用 $H(P)$ 来表示。

$$H(Y \mid X) = -\sum_{x,y} P(x) P(y \mid x) \log P(y \mid x) \quad (8)$$

最大熵模型对所有未知信息不做任何假设, 在所有满足已知约束的模型中, 不对未知作任何假设的那个模型具有最大的条件熵, 这个模型就是要求的最大熵模型。

$$P^* = \arg\max_{P \in C} H(P)$$
$$C = \{P \in \boldsymbol{P} \mid P(f_i) = P(f_i), i = 1, 2, \cdots, n\} \quad (9)$$

通过计算最后最大熵模型具有如下的表示形式:

$$P(y_i \mid X) = \frac{1}{Z} \exp\left(\sum_i \lambda_i f_i(c, y_i) \right) \quad (10)$$

其中 λ 是最大熵模型的参数, 每个 λ 对应于一个特征函数。Z 是归一化因子, 确保整个模型是一个合法的概率分布。在依存句法分析中依存关系权重计算公式(1)中权重向量 w 可以用 $\lambda_1, \lambda_2 \cdots \lambda_n$ 代替; 而特征向量: $f(i, j)$ 则可以用 $f_1(c, y_i), f_2(c, y_i) \cdots f_n(c, y_i)$ 这些特征来表示, 如果特征出现为1, 否则为0。

3.2 训练实例选择

在最大熵模型中, 实例是一个(x, y)对。其中 x 是上下文特征, y 是相应的标记。在依存句法分析问题中, 要考虑词与其他词的依存关系, 所以针对一个词会有多个实例。最简单的处理方法是考虑当前词与所有其他词之间的依存关系, 对于一个长度为 n 的句子, 会有 $n \times n-1$ 个实例。这种处理方法会导致实

例过多, 训练文件过大等问题。实际上随着两个词之间距离的增加, 它们之间存在依存关系的可能性会逐渐减少。我们统计了句法依存关系在词距离上的分布关系(如图 2)。图 2 中横坐标代表词之间的距离, 纵坐标代表依存关系的比例, 可以看出随着词之间距离的增大, 词之间存在依存关系的可能性逐渐减少。

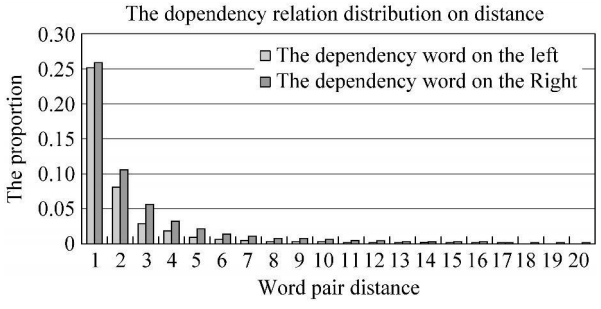

图 2　依存关系在单词距离上的分布

统计发现距离范围[- 13, 15] 内的依存关系数量占据了依存总数的 96.5%, 因此将[- 13, 15] 作为实例的选择范围。这样对于一个单词来说最多会生成 28 个实例。在训练过程中如果实例的两个词之间存在句法依存则 y 为他们的句法依存关系, 如果两个词之间不存在句法依存则 y 统一设为“null”。

3.3　特征模版

根据统计和语义关系选择特征模版来提取特征。特征提取是针对一个词对(i, j)进行的, 表示词 i 依存于词 j, 其中 i 代表当前正在处理的词, j 代表其他词。特征模版分为原子特征模版和组合特征模版, 如表 1。

表 1　特征模版表

W(i)	W(j)
P(i)	P(j)
P(i+1)	P(j+1)
P(i+2)	P(j+2)
P(i-1)	P(j-1)
P(i-2)	P(j-2)
Dis = (i-j)	
W(i) + W(j)	P(i) + P(j)
W(i) + W(j) + Dis	P(i) + P(j) + Dis
P(i) + P(j) + P(i-1)	P(i) + P(j) + P(i+1)
P(i) + P(j) + P(j-1)	P(i) + P(j) + P(j+1)

在特征模版中 W 代表词, P 代表词性, 括号内的数值代表位置信息。Dis 特征是距离特征, 表示单词 i, j 的距离, 距离特征不取绝对值有正负之分。+ 号代表多个特征的组合。

4　依存句法分析实验

训练的数据为 Conll2008 Share Task 的训练语料, 包含 39 279 个句子, 句子平均长度为 24.3 个单词。Conll2008 Share Task 的测试语料有两个, 分别是 WSJ 语料和 Brown 语料, 两个语料分别含有 2 399 和 425 个句子。三种算法的实验结果如表 2。从表 2 可以看出最大生成树算法明显好于自顶向下算法和自底向上算法。Wsj 测试语料的结果好于 Brown 语料, 一个原因是训练语料来源于 Wsj 语料库不能很好的描述 Brown 语料, 另一个可能的原因是 Brown 测试语料太少具有一定的偏差性。

表 2　三种算法实验结果

算　法	WSJ/%	Brown/%
Down2Up	81.05	75.98
Up2Down	83.17	76.25
MST	87.42	80.80

为了对实验的结果作进一步的分析, 统计了不同词性的标记结果。针对 Wsj 测试语料, 表 3 给出了详细的统计结果。

表 3　Wsj 测试语料详细标注结果

词　性	单词数	识　别	分　类	识别与分类同时正确
NN	8 321	7 774 (93%)	7 772 (93%)	7 633 (92%)
IN	5 992	5 019 (84%)	4 706 (79%)	4 351 (73%)
NNP	5 573	5 083 (91%)	5 122 (92%)	4 967 (89%)
DT	4 907	4 775 (97%)	4 822 (98%)	4 751 (97%)
NNS	3 680	3 403 (92%)	3 407 (93%)	3 368 (92%)

续表

词 性	单词数	识 别	分 类	识别与分类同时正确
JJ	3 401	2 847 (94%)	2 881 (95%)	2 802 (92%)
,	3 037	2 328 (77%)	3 015 (99%)	2 322 (76%)
.	2 345	2 220 (95%)	2 339 (100%)	2 220 (95%)
CD	1 989	1 820 (92%)	1 791 (90%)	1 744 (88%)
RB	1 901	1 598 (84%)	1 476 (78%)	1 400 (74%)
VBD	1 797	1 463 (91%)	1 649 (92%)	1 617 (90%)
VB	1 551	1 521 (98%)	1 516 (98%)	1 510 (97%)
VBN	1 371	1 264 (92%)	1 256 (92%)	1 220 (89%)
CC	1 367	1 106 (81%)	1 333 (98%)	1 104 (81%)
TO	1 233	1 077 (87%)	1 013 (82%)	993 (81%)
VBZ	1 233	1 111 (90%)	1 122 (91%)	1 096 (89%)
…	…	…	…	…
All	57 676	52 062 (90%)	53 020 (92%)	50 422 (87%)

表 3 列出了不同词性的词的依存关系预测实验结果。因为词性过多, 只列出数量较多的前 16 种词性。第一列是词性; 第二列是测试数据中对应具体词性的单词个数; 第三列是依存识别的正确数量和正确率; 第四列是依存关系分类的正确数量和正确率; 第五列是依存识别和分类同时正确的数量和正确率。标注结果显示不同词性的标记正确率不尽相同, 其中“ DT”和“ .”标注效果较好, 这是因为“DT”代表冠词, 一般来说是依存于紧跟着的名词, 而“ .”是句子的结束标志, 固定依存于句法树的根节点。有些词性比如“ IN”和“ RB”标注效果不是很好。“ IN”是介词一般依存于名词, 但有的时候又会依存于动词等; 而“ RB”有多种可能的依存关系而且依存距离分布范围较大, 所以标注难度较高。针对那些标注难度较大、效果不理想的词性应该进一步统计分析提取更加有效的特征以进一步提高模型的预测能力。

5 结论

本文提出了三种依存句法树算法, 其中最大生成树算法取得了最好的实验结果。在 CoNLL2008 提供的标准测试数据 Wsj 语料库上, 取得了 87.42% 的正确率。该实验结果在 2008 年 5 月份刚刚结束的 share task 评测中取得了第六名的成绩。实验证明基于最大熵模型的最大生成树算法对依存句法分析是有效的。分析了不同词性的标注结果, 发现有些词性的依存关系比较复杂, 标注正确率也很低, 下一步的工作是针对这些标注难度较大的词性增加一些特征以提高准确率。

参考文献:

[1] Lucien Tesni re. l ments de syntaxe structurale[M]. Klincksieck, Paris 1959.

[2] Ryan McDonald, Fernando Pereira, Kiril Ribarov, Non-projective Dependency Parsing using Spanning Tree Algorithms[M]. HLT-EMNLP 2005.

[3] Ryan McDonald, Fernando Pereira. Online Learning of Approximate Dependency Parsing Algorithms. [C]//EACL 2006.

[4] Hiroyasu Yamada and Yuji Matsumoto. Statistical dependency analysis with support vector machines [C]//Proceedings of the 8th InternationalWorkshop on Parsing Technologies (IWPT), 2003.

[5] Joakim Nivre and Mario Scholz. Deterministic dependency parsing of English text[C]//Proceedings of the 20th International Conference on Computational Linguistics (COLIN G), 2004.

[6] Adam Berger, Stephen Della Pietra, Vincent Della Pietra. A Maximum Entropy Approach to Natural Language Processing[J]. Computational Linguistics, 1996.

[7] M. Collins, A new statistical parser based on bigram lexical de-pendencies[C]//Proc. 34th Annu. Meeting Association for Computational Linguistics, May 1996: 184-191.

[8] 段湘煜, 赵军, 徐波. 基于动作建模的中文依存句法分析[J]. 中文信息学报, 2007, 21(5): 25-30.

[9] 张亮, 陈家骏. 基于大规模语料库的句法模式匹配研究[J]. 中文信息学报, 2007, 21(5): 31-35.

[10] 周明. 汉语句法分析器的鲁棒性研究[R]. 清华大学博士后出站报告, 1993.