基于语法功能匹配的汉语句法分析算法

卢俊之,陈小荷,王东波,陈 锋
LU Jun-zhi,CHEN Xiao-he,WANG Dong-bo,CHEN Feng

南京师范大学 文学院,南京 210097
School of Chinese Language and Literature,Nanjing Normal University,Nanjing 210097,China
E-mail:lujunzhi@gmail.com

LU Jun-zhi,CHEN Xiao-he,WANG Dong-bo,et al.Chinese parsing algorithm based on grammar function match. Computer Engineering and Applications,2008,44(16):151-153.

Abstract: Based on the primary method of Grammar Function Match as the syntactic analysis,this paper realizes a kind of parsing algorithm,which views the TCT 973 as principal resource to survey the grammatical function.This algorithm not only efficiently reduces the fake ambiguities,but also has a favorably analyzed efficiency,which analyzes results including abundant and accurate grammatical information.The experiment indicates that the rate of phrase precision and recall reaches 75.17%, 73.69% and 65.06%,56.55% respectively in close test and open test.

Key words: syntactic parsing;Grammar Function Match;Chinese treebank;Chinese parser

摘 要:以语法功能匹配作为句法分析的基本方法,以 100 万词清华 973 树库作为语法功能调查的主要资源,实现了一种基于语法功能匹配的句法分析算法。该算法能有效减少伪歧义,具有良好的分析效率,其分析结果包含了丰富而准确的语法信息。实验表明,短语正确率与召回率在封闭测试和开放测试中分别达到 75.17%、73.69%和 65.06%、56.55%。

关键词:句法分析;语法功能匹配;汉语树库;汉语分析器
DOI:10.3778/j.issn.1002-8331.2008.16.046 **文章编号**:1002-8331(2008)16-0151-03 **文献标识码**:A **中图分类号**:TP391

1 引言

句法分析是中文信息处理领域一个重要的基础性课题,同时也是一个公认的难题。究其原因,核心问题是信息不足——由于汉语缺乏形态变化,现有的词类标记(如 n、v)和短语标记(如 NP、VP)并不能清晰地反映其语法功能(如做主语、做谓语、做定语、做状语),因此在计算机看来充满歧义,难以支持自动句法分析。

陈小荷[1]指出:(1)每个词类到底有多少种语法功能不明确;(2)属于同一词类的词,其语法功能可能差异很大;(3)不同词类的词,其语法功能也许反而相似;(4)一些词的语法功能没有得到充分描写;(5)缺乏词的各种语法功能的定量描写。他提出了彻底地按照词的语法功能来划分汉语词类的设想,从 8 种句法结构、13 种句法成分中推导出词类。

徐艳华[2]实现了该设想,完成了面向中文信息处理的词类体系重构。她手工考察了 3 514 个常用词的语法功能,构建了语法功能信息库,并抽取 11 206 个 $v+v$ 序列和 10 081 个 $v+n$ 序列,利用该库进行结构消歧实验,匹配后只有一种句法关系的分别占到 83.9%和 70.7%。

本文在上述理论与成果的支持下,使用词和短语的语法功能集代替现有的词类标记和短语标记,以语法功能匹配(Grammar Function Match,以下简称 GFM)作为句法分析的基本方法,以 100 万词清华 973 树库(Tsinghua Chinese Treebank,以下简称 TCT 973)作为语法功能调查的主要资源,实现了一种基于 GFM 的句法分析算法。该算法能有效减少伪歧义,具有良好的分析效率,其分析结果包含了丰富而准确的语法信息。实验表明,短语正确率与召回率在封闭测试和开放测试中分别达到 75.17%、73.69%和 65.06%、56.55%。

全文共分 3 部分,首先介绍了 GFM 句法分析的基本思想,其次介绍了一种基于 GFM 的句法分析算法,最后介绍了实验方法、结果和分析。

2 GFM 基本思想

GFM 句法分析的基本思想是:把词和短语的语法功能调查清楚并存入机读词典,这相当于给汉语的词和短语加上了形态标记。于是,句法分析的过程基本上就是一个语法功能匹配的过程。比如:<地名> 有关 部门 接到了 <人名> 的 投诉信。理想的分析过程如图 1 所示。

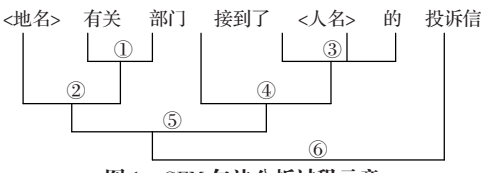

图 1 GFM 句法分析过程示意

作者简介:卢俊之(1980-),男,硕士生,主要研究领域为计算语言学;陈小荷(1952-),男,教授,博士生导师,主要研究领域为计算语言学;王冬波(1981-),男,硕士生,主要研究领域为计算语言学;陈锋(1982-),男,硕士生,主要研究领域为计算语言学。

收稿日期:2007-09-13 **修回日期**:2007-12-07

假定语法功能词典中,有关可以做定语、述语,部门可以做定语、定语中心语、宾语,因此有关部门可以构成定中结构(匹配①);<地名>可以做定语、主语、宾语,而有关部门这个定中结构可以做定语中心语、主语,因此<地名>有关部门就构成了一个新的定中结构(匹配②)。同样,③~⑥的匹配过程以此类推。最终得到分析结果:[ZJ [ZW [DZ <地名> [DZ 有关 部门]] [PO 接到了 [DZ<de> <人名> 的 投诉信]]] 。]。其中有3种匹配类型:(1)词与词,如匹配①、③;(2)词与短语,如匹配②、④;(3)短语与短语,如匹配⑤。匹配过程中也会有歧义,如有关部门同时能匹配成述宾结构,<人名>的能匹配成"的"字结构。原因一方面是因为词/短语(特别是短语)能充当的语法功能常常多样,另一方面匹配的优先顺序也是多样的。

GFM 方法将语法功能应用于句法分析的主要优势有:

(1)无需庞大的规则库,分析过程简单高效。将 TCT 973 中的句法规则归类精简后尚有 6 498 条,GLR 算法和 Chart 算法处理这些规则的时间、空间开销都很惊人,而本方法无需规则,也没有复杂的预处理,分析中只有相邻词/短语间的语法功能匹配。

(2)有效减少伪歧义,控制歧义树的数量。尽管本方法中歧义依然存在,但相比仅仅知道 n、b、n、v、n、u、n 这样的词性序列所能产生的分析结果,伪歧义的数量已大大减少。文献[2]的 $v+v$、$v+n$ 消歧实验和本文 4.3 节句平均歧义树数的统计结果证明了这点。

(3)分析结果提供了丰富而准确的语法信息。ZW(主谓)、DZ(定中)、PO(述宾)这样的短语标记提供给机器翻译等应用领域的信息大大超过 NP、VP 这样的标记。

3 GFM 句法分析算法

3.1 语法功能词典

词和短语的语法功能概率信息全部从 TCT 973 中自动统计。TCT 中共有短语标记 134 种[3]。如果完全照搬 TCT 标准,势必会存在较为严重的数据稀疏问题,同时也影响分析效率。因此,对 TCT 中的短语类型做了调整,将短语标记减少为 27 种,构成短语的语法功能标记为 60 种。表 1 是调整后的短语列表。

表 1 短语列表

短语标记	短语名称	元素个数	短语标记	短语名称	元素个数	短语标记	短语名称	元素个数
ZW	主谓	2	PO	述宾	2	AD	附加	2
DZ	定中	2	FW	方位	2	KS	框式	3
DZ<de>	带"的"定中	3	JB	介宾	2	SX	顺序	2
SL	数量	2	LW	连谓	2	SZ	"所"字	2
ZZ	状中	2	JY	兼语	3	LH	联合	多
ZZ<de>	带"地"状中	3	BK	比况	2	FH	复句	多
SB	述补	2	MJ	枚举	2	FJ	分句	多
SB<de>	带"得"述补	3	SJ	时间	2	YQ	语气	2
SB<bu>	带"不"述补	3	DE	"的"字	2	ZJ	整句	2

对于 2 元、3 元短语,功能标记是在短语标记前加上位置序号。如 1ZW 表示主谓结构的第 1 成分(主语),3SB<de>则表示带"得"述补结构的第 3 成分(补语)。而对于多元短语,序号 1 表示元素,2 表示连接成分。

从调整后的 TCT 中统计得到 4 部词典提供给句法分析算法使用:

(1)WDDict:43 057 个词直接充当 60 种语法功能的概率信息。

(2)WCDict:29 598 个词作为中心词充当 60 种语法功能的概率信息。

(3)PDDict:27 种短语直接充当 60 种语法功能的概率信息。

(4)ZPDict:27 种短语构成 ZJ 的概率信息。计算公式为(1),其中 2ZJ 由句末标点充当:

$$P(PHR_i|ZJ)=\frac{Count(ZJ\rightarrow PHRi+2ZJ)}{\sum_{j=1}^{27}Count(ZJ\rightarrow PHRj+2ZJ)} \quad (1)$$

3.2 GFM 句法分析算法

算法涉及到两种基本数据结构:

(1)共享森林(SF)。一个由若干结点构成的结点池,其中的结点可能是句子中的一个词,也可能是分析过程中得到的一个短语。

(2)出边/入边链(OLIST/ILIST)。将 *SF* 中每个结点看作一条以其所覆盖的左词为起点,右词为终点的有向边,OLIST/ILIST 记录第 i 个词位置上出边/入边的结点编号集。

算法还定义了两种运算:

(1)匹配概率运算。计算结点 $Node_m Node_n$ 匹配成短语 PHR_i 的概率,计算公式为(2)。为方便算法设计,将该运算统一为 2 目运算。对于 3 元短语,首先对前 2 元执行概率运算,得到中间结果再与第 3 元二次匹配;

$$P(PHR_i)=P(1PHR_i|Node_m)*P(2PHR_i|Node_n) \quad (2)$$

其中结点的功能概率有 4 种情况:①结点为词典词,直接在 WDDict 中查询;②结点为非词典词,假设它等概率地具有全部的语法功能;③结点为向心结构短语,概率为 PDDict 中查询的短语功能概率与 WCDict 中查询的中心词功能概率的平均值。多层结构中,中心词是继承的;④结点为离心结构短语,直接在 PDDict 中查询。

(2)子树/树生成概率运算。计算公式为(3)。其中 k 为子树/树的全部短语数。对于树,第 k 个短语将与句末标点匹配成 *ZJ*,因此要将子树概率乘以成句概率,成句概率从 ZPDict 中查询。

$$P(ST|S)=\prod_{i=1}^{k}P(PHR_i)\quad P(T|S)=P(PHR_k|ZJ)\prod_{i=1}^{k}P(PHR_i) \quad (3)$$

算法的原理是在 *SF* 中相邻(指词/短语在分析树上的位置相邻)的结点间动态地进行匹配,穷尽地发现所有可能的新匹配对(即短语结点)并加入 *SF*,不断循环直至不再有新结点产生,最终以 *SF* 中所有能覆盖整个句子的结点为根得到若干棵歧义树。消歧主要依靠语法功能词典中的概率信息,从歧义树中选择生成概率最大的一棵输出。匹配过程中,算法在 3 个方面进行了优化以减少匹配次数,提高分析效率:(1)组块捆绑:将高匹配概率的词对捆绑成组块,使其内部元素不再与外界匹配;(2)局部剪枝:提前发现没有继续生长价值的子树并剪去;(3)控制膨胀:限制跨度相同的匹配对的数量以防止局部歧义过度膨胀。

句法分析算法和功能匹配算法的描述分别如算法 1、算法 2 所示。

算法 1 句法分析算法。

输入:句子 $w_1w_2\cdots w_n$;

输出:最大概率分析树。

初始化:加载词典。

第 0 轮：依次读入 w_i 并插入 *SF*，填写 OLIST/ILIST。

重复以下步骤，直至不再增加新结点。每 1 次循环为 1 轮。

对上一轮 SF 新产生的每一个结点 $Node_i$，执行如下操作：

在 OLIST 中寻找 $Node_i$ 的每个右邻结点 $Node_j$，将 $Node_i$ 与 $Node_j$ 执行匹配算法，转算法 2；

在 ILIST 中寻找 $Node_i$ 的每个左邻结点 $Node_j$，若其也为上一轮新产生的结点，则跳过，否则将 $Node_j$ 与 $Node_i$ 执行匹配算法，转算法 2。

若存在 1 棵以上的分析树，则输出最大概率树；否则，分析失败。

算法 2 功能匹配算法。

输入：结点 $Node_i$，$Node_j$。

(1)判断下列两种情况：

①$Node_i$ 为 3 元结构已匹配 2 元。$Node_i$、$Node_j$ 进行二次匹配，若匹配概率 *MatchProb*>*MatchValue* 则转(2)，否则退出；

②$Node_i$ 为已完全匹配。处理 $Node_i$、$Node_j$ 的每一个匹配对(包括 2 元和 3 元的前 2 元)，若 *MatchProb*>*MatchValue* 则分别转(2)，否则退出。

(2)执行如下优化：

①组块捆绑。将 *MatchProb* 大于 *ChunkValue* 的词对捆绑为组块，转(3)；

②局部剪枝。若该子树/树的生成概率小于已成功的分析树的最大生成概率则退出，否则转(3)；

③控制膨胀。跨度相同的结点只保留 *K* 个生成概率最大的，跨度相同且语法功能相同的结点只保留 1 个生成概率最大的。若不在保留之列则退出，否则转(3)。

(3)插入新结点至 *SF*。情况 II 填写 OLIST/ILIST；情况 I，填写 ILIST。

其中涉及的 3 组阈值将在 4.2 节讨论：(1)*MatchValue*：分为词与词、词与短语、短语与短语 3 类，每类还分有 2 元、3 元两种情况；(2)*ChunkValue*：分为 2 元、3 元两种情况；(3)*K*。

3.3 尚未解决的问题

目前，算法对如下问题尚无法妥善解决，需依靠预识别处理：

(1)多重 LH 结构与无连词 LH 结构的识别。对于多重 LH 结构，观察窗口的大小无法控制；而无连词 LH 结构，匹配十分随意。

(2)复句的识别。算法只能处理单句的情况，对于复句需预识别出复句关系并将其分割为若干单句分别分析。

4 实验分析

4.1 实验设计与评测标准

实验分封闭测试与开放测试。封闭测试集是从 TCT 973 中选取出的能回避 3.3 节问题的单句 2 856 句。开放测试集为按相同标准从哈工大依存树库中选取的单句 100 句，并按本文规范人工重新标注目标集。

评测标准采用 PARSEVAL 句法分析评价体系[4]，它是一种粒度适中的评价方法。设树库中标注的所有短语的集合为目标集，句法分析实际分析出的短语集合为分析集，分析集和目标集的交集为共有集。短语正确率(*PP*)和召回率(*PR*)的计算公式为(4)。

$$PP=\frac{Count(\text{共有集})}{Count(\text{分析集})} \quad PR=\frac{Count(\text{共有集})}{Count(\text{目标集})} \tag{4}$$

这里还统计了句子分析率[5]、整句正确率和句平均歧义树数。句子分析率是指可以得到分析结果的句子在所有待处理句子中的比率；整句正确率是指分析完全正确的句子的比率；而句平均歧义树数是指有分析结果的句子平均歧义树的数量。

4.2 阈值的选取

MatchValue 的选取。从 TCT 973 中分类统计了所有匹配的概率，每类阈值保证有 98%的识别率，确定见表 2。

表 2　匹配概率的阈值

匹配类型	2 元	3 元
词与词	0.001 663	0.000 037
词与短语	0.001 051	0.000 013
短语与短语	0.000 871	0.000 154

ChunkValue 和 *K* 的选取。在封闭测试中将阈值各分 10 级并比较分析结果(图 2)，最终确定为：*ChunkValue*(2 元)=0.6，*ChunkValue*(3 元)=0.3，*K*=3。

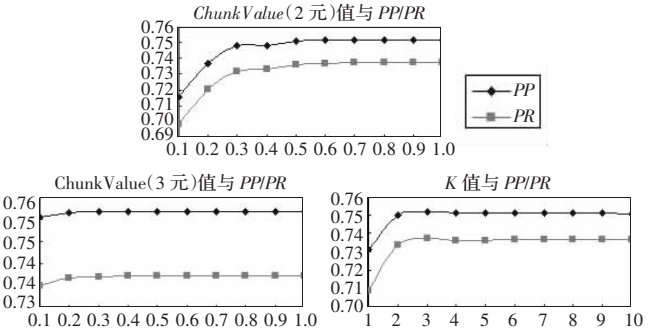

图 2　*ChunkValue*/*K* 与 *PP*/*PR* 关系

4.3 实验结果

各项参数的实验结果如表 3 所示。

表 3　实验结果

参数	封闭测试	开放测试
句子数	2856	100
平均词长	8.53	9.11
分析时间*	129.953 s	3.672 s
PP	75.17%	65.06%
PR	73.69%	56.55%
句子分析率	98.04%	86%
整句正确率	66.88%	46%
句平均歧义树数	3.51	3.94

这里还分别统计了封闭测试中各主要短语的 *PP*/*PR*(见表 4)和词与词、词与短语、短语与短语匹配的 *PP*/*PR*(见表 5)。

表 4　各主要短语的 *PP*/*PR*

短语	*PP*	*PR*	短语	*PP*	*PR*	短语	*PP*	*PR*
ZW	66.05%	64.76%	*ZZ<de>*	90.35%	88.03%	*FW*	88.42%	83.79%
DZ	77.93%	80.75%	*SB*	95.11%	74.56%	*JB*	85.74%	83.58%
DZ<de>	77.88%	78.42%	*SB<de>*	98.31%	87.88%	*JY*	81.48%	47.48%
SL	95.28%	87.37%	*SB<bu>*	100.00%	35.48%	*DE*	73.49%	40.13%
ZZ	73.38%	72.32%	*PO*	75.49%	74.56%	*AD*	83.33%	60.61%

表 5　各匹配类型的 *PP*/*PR*

匹配类型	*PP*	*PR*
词与词	93.21%	89.02%
词与短语	69.96%	70.83%
短语与短语	60.08%	55.74%

* 测试环境：Celeron 2.8 GHz CPU，512 M RAM，WinXP。

(下转 159 页)