基于关联规则挖掘的汉语语义搭配规则获取方法

郑旭玲,周昌乐*,李堂秋,陈毅东
(厦门大学计算机科学系,福建 厦门 361005)

摘要:针对自然语言处理系统在短语分析时的词汇排歧和结构排歧需要,本文提出了一种基于语料库的汉语短语语义搭配规则自动获取方法.该方法以《知网》为语义知识资源,在标注了句法语义信息的汉语短语熟语料库基础上,先采用数据挖掘中元规则制导的交叉层关联规则挖掘方法,自动发现汉语短语的语义搭配规律,再根据统计结果自动优选后生成语义搭配规则库.实验结果表明该方法是切实可行的.运用该方法自动获取的语义搭配规则具有较好的排歧效果.

关键词:语义规则;语料库;关联规则;知网
中图分类号:TP 391.2 **文献标识码**:A **文章编号**:0438-0479(2007)03-0331-06

在机器翻译、信息检索、文本分类等诸多自然语言处理系统中,短语分析都是至关重要的一个环节,其分析质量的优劣直接影响系统的最终性能.歧义现象的存在给短语自动分析设置了巨大的无法回避的障碍.仅仅依靠词法和语法知识来消解短语分析中大量存在的词汇歧义和结构歧义是远远不够的.尤其是在分析像汉语这样形态特征较少而内涵却极为丰富的"意合"语言的短语时,更需要引入"具有什么语义的词语可以相互组合、以怎样的方式组合成怎样的短语"这样的语义搭配规则知识.另一方面,对汉语短语的语义搭配规则的研究,不仅能为短语分析提供有效指导,从而提高短语结构和语义分析的正确率,而且还能辐射对汉语词和句子的研究,促进汉语文本分析质量的全面改善.然而,这方面研究面临语义知识的形式化表示和语义搭配规则库的构造两大瓶颈问题.

近些年来,国内外在构建语义知识表示体系的工程实践方面取得了不少成果,开发出 WordNet、FrameNet、Hownet[1] 等计算机用语义词典.相比之下,语义搭配规则库的构造相对薄弱些.詹卫东[2]、俞士汶[3]、董振东和董强[4]等人对汉语短语的语义规则作了大规模的系统研究,并取得了一些成果.但这些规则主要是通过基于直觉的方法获得的,其准确性和完备性比较难以保证.

我们的研究工作是尝试利用语料库来自动获取汉语短语的语义搭配规则,从而验证已有的基于直觉的语义搭配规则和发现新的语义搭配规则.

1 语义搭配规则获取系统的架构

基于语料库的方法利用计算机对语料进行统计归纳,既减少了语言知识描述系统构造中的主观性,又降低了语言学家的工作难度和强度.故我们的系统采用基于语料库的方法,其架构及工作流程如图 1 所示.

我们的系统采用《知网》的语义知识表示体系[1].初始语料库是一个由大量真实文本中抽取的实例组成的汉语短语生语料库.首先,依据语言学家提供的初始语言知识,对其进行人-机交互的半自动加工,标注上各短语中每个词语的词性、义项[1]和词语间的句法结

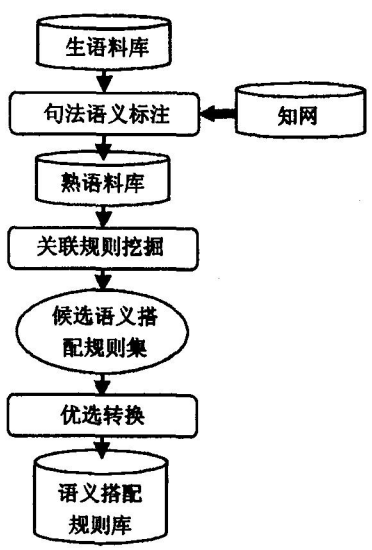

图 1 汉语语义搭配规则获取系统流程

Fig. 1 The work flow of Chinese semantic collocation rules acquisition system

收稿日期:2006-10-08
基金项目:国家自然科学基金(60373080)资助
*通讯作者:dozero @xmu.edu.cn

构、语义组合关系[4],使其成为熟语料库;然后,运用数据挖掘中元规则制导的交叉层关联规则挖掘方法来自动发现熟语料库中词语组合的语义规律,形成候选语义搭配规则集;最后,根据统计结果,经优选算法自动筛选后,转换生成短语分析排歧所需的语义搭配规则库.

2 语义搭配规则的形式化表示

针对《知网》知识词典描述语言 KDML[1]和汉语短语构造的语义规律的特点,我们设计了如下形式的语义搭配规则,形式化地描述短语分析排歧所需的关于汉语中具有什么语义的词语可以相互组合、以怎样的方式组合成怎样的短语的语义知识.

< Rule > ::= (< POS-Serial > < Syntax-Struction-Type >
< Semantic-Relation-List >
< DEF-Pattern > { < DEF-Pattern > }+)
< POS-Serial > ::= (< POS > { < POS > }+)
< POS > ::= n| v| a| p| m| ……
< Syntax-Struction-Type > ::= < Tag >
< Tag > ::= {0| 1| 2| …| 9}+
< Semantic-Relation-List > ::= ({ < Semantic-Relation > }+)
< Semantic-Relation > ::= 合成| 修饰| 限定| ……
< DEF-Pattern > ::= (< First-Item-Pattern > { < Sememe-Pattern > }*)
< First-Item-Pattern > ::= (FIRSTITEM < First-Item >)
< First-Item > ::= ([* HYP *] < Main-Sememe >)|
< Variable >
< Main-Sememe > ::= { Hownet 中概念的主要特征}
< Sememe-Pattern > ::= (< Relation > { < Sememe-or-Variable > }+)
< Relation > ::= Partof| Material| Feature | ……
< Sememe-or-Variable > ::= (*DEFVAR * < Varialbe >
< Domain >)| < Sememe-Item >
< Domain > ::= ({ < Sememe-Item > }+)
< Sememe-Item > ::= < Sememe > | ([* HYP *]
< Sememe >)| < Variable >
< Sememe > ::= < Main-Sememe > | < Secondary-Sememe >
< Secondary-Sememe > ::= { Hownet 中概念的次要特征}
< Variable > ::= var1| var2| var3| ……

一条语义搭配规则是由一个词性序列(即 < POS-Serial >)、一个句法结构类型编码(即 < Syntax-Struction-Type >)、一个语义关系列表(即 < Semantic-Relation-List >)以及若干个语义模式(即 < DEF-Pattern >)构成. 若短语由 n 个成分组成,其对应的规则应包含 n 个语义模式,且称这样的规则为 n 元规则.

词性类序列标明了适用该规则的短语中各成分的词性. 句法结构类型编码指明了该短语的句法结构. 语义关系列表给出了该短语各成分之间的语义关系. 而义项模式则描述了该短语各成分应当具有的语义特征. 为增强描述能力,我们在 KDML 基础上引入了前缀 * HYP * 和受限变量. 带前缀 * HYP * 的义原表示该位置上可以取义原分类树上该义原的任一子孙义原,它增强了规则的概括能力. 用指定了义原取值范围的受限变量代替具体义原,使规则可以更概括地描述短语各成分间在语义上的联系.

3 语义搭配规则的自动挖掘

3.1 数据预处理

考虑到语义搭配规则的运用是以句法分析为基础的,描述不同词性序列的短语组合规律的语义搭配规则之间并不存在交叉和抵触,为减少规则挖掘中不必要的计算,我们将语料库中的短语根据词性序列不同划分子语料库. 由于同一子语料库的所有语料在词性构成序列上均相同,因此对各个子语料库分别进行规则挖掘时就无需考虑词性问题了. 我们对各个子语料库所采用的规则挖掘和优选算法相同,为方便叙述,本文仍用“语料库”一词来指称任意一个子语料库.

结合语料库中标注信息的特点以及关联规则挖掘算法实现的需要,我们将熟语料库中的数据转换成如下形式来存储:

(< ExampleID > , < No > , < Info-Type > ,
< Info >)

其中: < ExampleID > 为实例在语料库中的编码(本文将同一实例的不同标注方案看成是不同的实例,给予不同的编码); < No > 为信息标注的对象编码,0 表示标注的是整个实例的信息(如短语的句法结构、语义关系等),1,2, …, n 分别表示标注的是短语中第 1、2、…、n 个成分的信息(如对应的词语、概念定义中出现的义原及其前缀符等); < Info-Type > 为信息的类型编码,例如,WC 代表构成短语的汉语词语,NF 代表其对应概念定义中无前缀语义符的首位义原,PS 代表其对应概念定义中带前缀语义符“ % ”的非首位义原; < Info > 为具体的信息,如汉语词语、语义关系、义原等等.

语义搭配规则需要在不同的义原概念层次上进行概括,而《知网》提供的义原分类树描述了义原之间的上下位关系,正是进行关联规则挖掘所需的体现数据自底向上概括和自顶向下特殊化关系的概念层次.

3.2 问题描述

经上述数据预处理后,从语料库中自动获取候选语义搭配规则的问题就可以转换为元规则制导的布尔型单维交叉层关联规则挖掘问题[5-7],其形式化描述

如下:设项目全集 $I=\{i_1, i_2, \cdots, i_m\}$, I 中任意一个项目 $i_j(j=1,2,\cdots,m)$ 是一个有序 3 元组 (n_j, y_j, a_j),用于代表语料库中的一个具体标注信息,其中:$n_j \in \mathbf{N}$, $\mathbf{N}$ 为语料库中所有信息标注对象编码构成的集合;$y_j \in \mathbf{Y}$, $\mathbf{Y}=\{\text{cw}, \text{sr}\} \cup \mathbf{F} \cup \mathbf{O}$ 为语料库中所有信息类型编码构成的集合,cw、sr 分别对应于句法结构、语义关系,$\mathbf{F}$ 对应于《知网》概念定义中首位义原的所有可能的语义前缀符构成的集合,$\mathbf{O}$ 对应于《知网》概念定义中非首位义原的所有可能的语义前缀符构成的集合;$a_j \in \mathbf{A}$, $\mathbf{A}=\mathbf{Q} \cup \mathbf{R} \cup \mathbf{S}$ 为语料库中所有标注信息构成的集合,$\mathbf{Q}$ 为句法结构类型编码全集,$\mathbf{R}$ 为《知网》中定义的语义关系序列全集,$\mathbf{S}$ 为《知网》中定义的义原全集.

语料库中所有编码不同的实例构成数据库事务全集 D,其中每个事务(即实例) T 形如:{ExampleID, $<T_1, T_2, \cdots, T_k>$}, $T_i \in I(i=1,2,\cdots,k)$.

由于只有项目全集 I 中对应于义原标注信息的那部分项目需要考虑概括,因此概括项目全集 $GI=\{A \mid A \in (\mathbf{N}-\{0\}) \times (\mathbf{F} \cup \mathbf{O}) \times \mathbf{GS}\}$,其中 $\mathbf{GS}$ 是 $\mathbf{S}$ 中各义原的概括(作标记以区别于 $\mathbf{S}$ 中的义原).

考虑到语料库中的每个实例均应具有句法结构类型、语义关系序列和各个成分的首位义原这三类标注信息,而且这三类标注信息也是描述词语组合规律时必不可少的,它们共同构成了我们所需挖掘的语义搭配规则的基本框架. 对应于这三类标注信息的 3 个必要项目集分别记作 β_1、β_2 和 β_3,其中 $\beta_1=\{0\} \times \{\text{cw}\} \times \mathbf{Q}$, $\beta_2=\{0\} \times \{\text{sr}\} \times \mathbf{R}$, $\beta_3[j]=\{j\} \times \{\text{nf}\} \times (\mathbf{S} \cup \mathbf{GS})$, $\beta_3=\cup_{j \in (\mathbf{N}-\{0\})} \beta_3[j]$. 此外,规则中不允许出现重复项,项之间也不应存在祖孙关系(即某一者 I_1 是另一者 I_2 的概括,记作 $I_1 \propto I_2$). 因此,我们将这些约束条件作为元规则(规则模板),用于约束和指导挖掘过程,提高挖掘效率. 元规则 P 是形如

$$A_1 \wedge A_2 \wedge \cdots \wedge A_k \Rightarrow B_1 \wedge B_2 \quad (k \geq 2)$$

的 n 元规则模板,其中 $A_1, A_2, \cdots, A_k, B_1, B_2 \in I \cup GI$,且同时满足:

(1) $\exists ! A_{ij} \in \{A_1, A_2, \cdots, A_k\}$, $A_{ij} \in \beta_3[j]$, $j=1,2,\cdots,n$;

(2) $B_1 \in \beta_1$, $B_2 \in \beta_2$;

(3) $\neg(\exists i,j(A_i \propto A_j))$, $1 \leq i, j \leq k$);

(4) $\{A_1, A_2, \cdots, A_k\} \cap \{B_1, B_2\} = \varnothing$.

我们的候选语义搭配规则自动获取问题就是在上述定义的事务全集 D、项目全集 $I \cup GI$ 和概念分层上挖掘所有满足元规则 P 的形如"$A \Rightarrow B$"的交叉层强关联规则,其中 $A, B \subset I \cup GI$ 且 $A \cap B = \varnothing$.

依据关联规则挖掘方法,此问题可划分成两个子问题:(1) 根据最小支持度找出 D 中符合元规则 P 约束条件的所有交叉层频繁项集;(2) 根据最小置信度和(1) 中的频繁项集产生满足元规则的强关联规则.

3.3　发现频繁项集

为了表述方便,定义以下相关符号:

$L[l,k]$:频繁 l 层 k 项集;

L_k:频繁 k 项集,即 $\cup_l [l,k]$;

L_k^P:符合元规则 P 约束的全部频繁 k 项集;

C_k^P:符合元规则 P 约束的全部候选频繁 k 项集;

D_k:运用 L_k^P 进行过滤得到的压缩事务数据库;

针对语义搭配规则自动获取的需要,我们对频繁项集发现算法——Apriori 算法[8] 作了一些变形,以适应我们的交叉层关联规则挖掘和元规则 P 约束的需要. 变形后的算法如下:

算法 1　面向语义搭配规则自动获取的频繁项集发现算法

输入:事务数据库 D;最小支持度计数阈值 min_sup;概念层次的最大深度 max_level;淘汰率阈值 $\sigma(0<\sigma<1)$.

输出:事务数据库 D 中符合元规则 P 约束的全部频繁项集 L.

处理过程:

```
L_k^P = L_1 = find_frequent_1-itemsets(D, min_sup, max_level, θ);
D_1 = get_filtered_transaction_talbe(L_1^P, D);
for (k=2; L_{k-1}^P ≠ Ø; k++) do begin
  C_k^P = gen_candidates(L_{k-1}^P);
  for each transaction t ∈ D_{k-1} do begin
    C_t = subset(C_k^P, t);
    if C_t ≠ Ø then begin
      for each candidate c ∈ C_t do
        c.sup_count++;
      t' = filter_items(C_t, t);
      add t' to D_k;
    end
  end
  L_k^P = {c ∈ C_k^P | c.sup_count ≥ min_sup};
  if (|C_k^P| - |L_k^P|)/|C_k^P| ≥ σ then
    D_k = get_filtered_transaction_talbe(L_k^P, D_k);
end
return L = ∪_k L_k^P;
```

算法 1 使用逐层搜索的迭代方法,首次搜索是通过调用 find_frequent_1-itemsets 函数来找出全部频繁 1 项集 L_k^P. 随后的搜索均包括两个阶段:首先,使用在第 $(k-1)$ 次搜索中找到的符合元规则 P 约束的频繁 $(k-1)$ 项集 L_{k-1}^P 和 gen_candidates 函数产生符合元规则 P 约束的候选频繁 k 项集 C_k^P;然后,扫描数据库,

计算 C_k^P 中各个候选的支持度,继而找到符合元规则 P 约束的频繁 k 项集 L_k^P. 为了提高算法效率,我们在搜索的同时对事物数据库做了适当的压缩.

(1) find_frequent_1-itemsets 函数　通过对事物数据库 D 的一次扫描,统计出每个项目的支持度计数,然后由最底层向上逐层计算出所有概括项目的支持度计数,就可根据最小支持度确定出所有层次上的全部频繁 1 项集.

(2) gen_candidates 函数　gen_candidates 过程分连接和剪枝两大步骤,用于产生符合元规则 P 约束的候选频繁 k 项集,具体算法如下:

算法 2　符合元规则 P 约束的候选频繁 k 项集 C_k^P 产生算法

输入:符合元规则 P 约束的频繁 $(k-1)$ 项集 L_k^P.

输出:符合元规则 P 约束的候选频繁 k 项集 C_k^P.

处理过程:

```
for each itemset A ∈ L^P_{k-1} do
  for each itemset B ∈ L^P_{k-1} do
    if ((A1 = B1) and ··· and (A_{k-2} = B_{k-2}) and (A_{k-1} < B_{k-1}))
      and (not (A_{k-1} ∝ B_{k-1}) or not (B_{k-1} ∝ A_{k-1}))
      and ((∀i(B_{k-1} ∉ β_i)) or (∀j(B_{k-1} ∈ β_j and A_{k-1} ∉ β_j))
        or (∀h(B_{k-1} ∈ β_3[h] and A_{k-1} ∉ β_3[h])), 1 ≤ i ≤ 3,
        1 ≤ j ≤ 2, 1 ≤ h ≤ n) then begin
      c = join(A, B);
      if has_infrequent_subset(c, L^P_{k1}) then
        if has_ancestor_itemset(c, E_t) then
          delete c;
        else begin
          E_t = remove_progeny_itemset(c, E_t);
          add c to E_t;
        end
      else add c to C^P_k;
    end
for each candidate c ∈ C^P_k do
  if has_ancestor_itemset(c, E_t) then
    remove c from C^P_k;
return C^P_k;
```

算法 2 的连接步(调用 join 函数之前)的逻辑表达式描述了元规则 P 所要求的项之间不存在祖孙关系、不存在同类的必要项目等约束条件,确保仅将 L_k^P 与 L_{k-1}^P 的连接中符合元规则 P 约束的 k 项集作为候选. 候选项集是频繁项集的超集,这个集合可能很大. 为压缩候选项集,通常使用 Apriori 性质[8]进行剪枝. 在交叉层上进行约束挖掘,须将 Apriori 性质推广到交叉层和约束性挖掘上,可得到如下两个推论.

推论 1　符合元规则 P 约束的频繁 k 项集的所有非空子集必须也都是符合元规则 P 约束的频繁项集.

推论 2　使用一致支持度时,频繁项集的祖先必须也都是频繁项集;反之,非频繁项集的任何子孙必定是非频繁项集.

算法 2 的剪枝步就是分别使用上述两个推论对 C_k^P 作两轮剪枝. 第 1 轮剪枝是在连接步产生候选的同时运用推论 1 进行的. 将第 1 轮修剪下来的候选项集收集起来(即 E_t),就可在所有连接完成之后,运用推论 2 和 E_t 对 C_k^P 做第 2 轮剪枝. 为了减少模式之间重复的祖孙关系判定,E_t 只需要保留所有已知的非频繁 k 项集的最小覆盖集合.

(3) 压缩事物数据　我们在算法 1 中采取了 3 个事务压缩措施:1) 在发现所有频繁 1 项集后,根据 L_1^P 删除事务中不被任何频繁 1 项集包含的项和不包含任何频繁 1 项集的事务,得到压缩事务数据库 D_1;2) 在扫描 D_{k-1} 以确定 L_k^P 的同时,删除事务中不被任何候选 $(k-1)$ 项集包含的项和不包含任何候选 $(k-1)$ 项集的事务,从而得到压缩事务数据库 D_k;3) 确定了 L_k^P 之后,只有当候选项集的淘汰率大于设定的阈值时,才根据 L_k^P 再次压缩 D_k.

3.4　生成候选规则

根据元规则的约束,算法 1 所发现的任何一个频繁项集 c 中必定有且仅有一个标注句法结构类型的项和一个标注语义关系序列类型的项,不妨记作 b_1 和 b_2 $(b_1 \in \beta_1, b_2 \in \beta_2)$,若将 c 中的其它项分别记作 $a_1, a_2, \cdots, a_k$,那么由频繁项集 c 生成的候选规则 r 形如:$a_1 \wedge a_2 \wedge \cdots \wedge a_k \Rightarrow b_1 \wedge b_2$.

其置信度可按如下公式求得:

$$\text{confidence}(r) = P(\{a_1, a_2, \cdots, a_k\} \mid \{b_1, b_2\}) = \frac{\text{sup\_count}(c)}{\text{sup\_count}(\{b_1, b_2\})}.$$

显然,c 和 c 的子集 $\{b_1, b_2\}$ 都是符合元规则 P 约束的频繁项集,即都是算法 1 的求解结果集合 L 中的成员,故它们的支持度计数都已知. 而且由于 c 是频繁的,故只要 r 满足最小置信度,r 就一定是强规则.

至于规则形式的转换,只要按照数据预处理的原则逆向转换,并在代表义原的概括的项之前加上 * HYP * 前缀,即可得到对应的候选语义搭配规则.

4　语义搭配规则的自动优选

在挖掘交叉层关联规则时,由于项之间的祖孙关系以及项集之间的子集/超集关系,挖掘出来的候选规则集(记作 GL)中混杂了大量冗余规则. 如果将其直接用于我们的短语分析排歧,不仅会增加计算量,还会带来许多干扰和误导. 为了删除冗余规则[9-10],我们根据规则之间支持度和置信度的近似程度及最强限制

原则,在 GL 上定义了一种覆盖关系.

对于 GL 中的任意两条规则 p 和 q, p 为 $X \Rightarrow Y$, q 为 $A \Rightarrow B$. 若将 q 中的项用它在概念分层中的祖先替换能够得到 p,则称 p 是 q 的祖先, q 是 p 的子孙. 若规则 p 和 q 的右部相同即 $Y = B$,且它们的左部满足 $A \subset X$,则称 q 是 p 的子规则.

对于 GL 中同时满意下列条件的任意两条规则 p 和 q:

(1) p 和 q 的置信度近似相等,即| confidence(p) - confidence(q)| $\leq \varepsilon$(趋近于 0);

(2) p 是 q 的祖先规则或子规则,或者存在 p 的子孙规则是 q 的子规则.

若 p 和 q 的支持度也近似相等,即

$$\frac{\text{sup\_count}(q)}{\text{sup\_count}(p)} \geq \mu,$$

则称 q 覆盖 p,记作 $p \preccurlyeq q$;反之,则认为 p 的支持度较明显地大于 q 的支持度,称 p 覆盖 q,记作 $q \preccurlyeq p$.

当 ε 和 μ 都趋近于 0 时,可近似地认为集合 GL 上的这种覆盖关系具有传递性,即可近似地将(GL, $\preccurlyeq$)看成一个偏序集. 受偏序集的 Hasse 图的启发,我们采用如下方法来绘制 GL 上的覆盖关系图:

(1) 省略关系图中每个结点上的自环;

(2) $\forall p, q \in GL$,若 $p \preccurlyeq q$,则在 p 和 q 之间存在一条从 q 指向 p 的有向边;若存在 $r \in GL$ 且 $p \preccurlyeq r \preccurlyeq q$,则 p, q 之间的这条有向边可以省略.

我们需要从 GL 中优选的正是这样一张关系图上入度为 0 的结点所对应的规则. 因此,所采用的语义搭配规则优选算法的主要思路为:首先,按候选规则的长度(即对应的频繁项集长度)非递减顺序扫描 GL,根据覆盖关系的定义,计算各个候选规则的入度(实际上只需要判定入度是否为 0);然后,从 GL 中选出入度为 0 的所有候选规则. 具体算法如下:

算法 3　n 元语义搭配规则优选算法

输入:n 元候选规则集 GL ($GL = \cup_k GL_k$, $min\_len \leq k \leq max\_len$,其中 GL_k 是规则长度为 k 的所有 n 元候选规则, min_len 和 max_len 分别为 GL 中候选规则的最小和最大长度);覆盖率阈值 $\mu(0 < \mu < 1)$;置信度近似阈值 $\varepsilon(0 < \varepsilon < 1)$.

输出:n 元语义搭配规则集合 R.

处理过程:

```
for (k = min_len; k ≤ max_len; k++) do
  for each candidate q ∈ GL_k do
    q.indegree = 0;
for (k = min_len + 1; k ≤ max_len; k++) do
  for each candidate q ∈ GL_k do
    for (j = min_len; j ≤ k; j++) do
      for each candidate p ∈ GL_j do
        if (p.indegree = 0) or (q.indegree = 0) then
          if (| conf(p) - conf(q)| ≤ ε) and
            is_sub_or_ancestor_set(p, q) then
            if (q.sup_count/ p.sup_count ≥ μ) then
              p.indegree++;
            else q.indegree++;
for (k = min_len; k ≤ max_len; k++) do
  R_k = { q ∈ GL_k | q.indegree = 0};
return R = trans_pattern_to_rule( ∪_k R_k).
```

5　实验结果及分析

目前,我们构造的汉语短语语料库包含从《人民日报》、《读者》等真实文本中抽取的“n + n”、“adj + n”、“m + n”、“adj + adj”、“m + adj”、“v + n”、“n + v + n”、“adj + n + n”等八大类短语共 8 516 个. 将本文提出的语义搭配规则获取方法运用该语料库,共获得 379 条语义搭配规则.

为考察本文方法的实际效果,我们将语料库所涉及的《结构库》中的信息结构模式直接改写得到的所有语义搭配规则作为规则集 Ⅰ,将运用候选语义搭配规则挖掘算法发现的所有候选语义搭配规则作为规则集 Ⅱ,将规则集 Ⅱ经优选算法筛选后剩余的所有语义搭配规则作为规则集 Ⅲ,采用相同的测试集和评价标准(要求完全匹配),分别对这三个规则集作封闭测试和开放测试. 封闭测试使用的是语料库中的所有短语实例,而开放测试使用的是从真实文本中另外抽取的 400 个短语实例(已确定词性). 实验结果如表 1.

表 1　不同规则集的测试结果

Tab. 1　Results of different rule-sets

规则集	规则数	封闭测试/ %		开放测试/ %	
		召回率	正确率	召回率	正确率
Ⅰ	215	80.4	69.5	76.2	64.8
Ⅱ	3266	74.9	84.1	76.7	83.5
Ⅲ	379	73.2	86.1	74.5	83.2

其中,使用的两个指标:

$$\text{召回率} = \frac{\text{可识别(即存在完全匹配规则)的短语个数}}{\text{总短语个数}} \times 100\ \%,$$

$$\text{正确率} = \frac{\text{正确识别的短语个数}}{\text{可识别的短语个数}} \times 100\ \%.$$

上述实验结果表明,尽管运用自动获取方法得到的规则集 Ⅱ和 Ⅲ在召回率上不及《结构库》的原有规则

集 I,但精确率却显著提高了,这也正是基于语料库的方法比基于直觉的更全面客观的体现. 对比 3 个规则集的规模,规则集 III比规则集 I多了 100 多条规则,但分析正确率却显著提高了,由此增加的时间消耗还是可以接受的;而规则集 II与规则集 III的召回率和精确率相差不大,但规模却是规则集 III的 8.6 倍,在封闭测试时其耗费的时间相当可观. 由此说明,我们提出的语义搭配规则挖掘算法和优选算法是合理的,运用数据挖掘方法从语料库中自动获取语义搭配规则的思想是可行的.

6 结 论

本文针对汉语短语分析排歧的需要,提出了一种将基于规则和基于语料库相结合的语义搭配规则获取方法. 它先采用关联规则挖掘方法来自动发现熟语料库中的语义搭配规律,再根据统计结果经优选算法自动筛选后转换生成语义搭配规则库. 实验结果表明该方法是切实可行的.

本文提出的语义搭配规则自动获取方法不仅适用于采用《知网》语义知识表示体系的短语语料库,而且也适用于类似的采用显式标注词语语义概念或语义属性的短语语料库. 此外,只需稍作修改,该方法也可用于融合句法和语义的短语搭配规则的获取.

我们已将采用本文方法获取的汉语短语语义搭配规则应用到我们的机器翻译系统中并取得了较好的排歧效果. 由该方法获取的语义搭配规则还可应用到诸如命名实体识别、文本分类、文本过滤等需要词义或结构排歧的自然语言处理领域.

参考文献:

[1] 董振东,董强. 知网[EB/OL]. (2000-10-25)[2006-10-07] http://www.keenage.com/zhiwang/c_zhiwang.html.

[2] 詹卫东. 面向中文信息处理的现代汉语短语结构规则研究[M]. 北京:清华大学出版社,2000.

[3] 俞士汶. 现代汉语短语结构知识库规格说明书[J]. 汉语语言与计算学报,2003,13(2):215 - 226.

[4] 董振东,董强. 关于知网-中文信息结构库[EB/OL]. (2000-10-25)[2006-10-07] http://www.keenage.com/html/c_index.html.

[5] Han J, Kamber M. Data mining: concepts and techniques[M]. San Francisco: Morgan Kaufmann Publishers, 2001.

[6] Han J, Fu Y. Discovery of multiple-level association rules from large databases[C]//Proceedings of 21th International Conference on Very Large Data Bases. Zurich: Morgan Kaufmann Publishers, 1995: 420 - 431.

[7] 欧阳为民,蔡庆生. 大型数据库中多层关联规则的元模式制导发现[J]. 软件学报,1997,8(12):920 - 927.

[8] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases[C]//Proceedings of the 1993 ACM-SIGMOD International Conference on Management of Data. Washington: ACM Press, 1993: 207 - 216.

[9] Aggarwal C C, Sun Z, Yu P S. Online generation of profile association rules[C]//Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining. Florida: AAAI Press, 1998: 129 - 133.

[10] Aggarwal C C, Yu P S. A new approach to online generation of association rules[J]. IEEE Transactions on Knowledge and Data Engineering, 2001, 13(4): 527 - 540.

Automatic Acquisition of Chinese Semantic Collocation Rules Based on Association Rule Mining Technique

ZHENG Xu-ling, ZHOU Chang-le*, LI Tang-qiu, CHEN Yi-dong

(Department of Computer Science, Xiamen University, Xiamen 361005, China)

Abstract: The semantic collocations play important roles in parsing Chinese phrases. It is useful for both semantic disambiguation and structural disambiguation. In this paper, a corpus-based method was proposed to automatically acquire semantic collocation rules from a Chinese phrase corpus, which was annotated with semantic knowledge according to HowNet. Moreover, a metarule-guided algorithm for mining cross-level association rules was developed to acquire semantic collocation rules from the corpus. And an optimized algorithm was developed to filter these rules. The experiment results showed the effectiveness of the proposed method. Disambiguation performance of the automatically acquired rules was quiet well.

Key words: semantic rules; corpus; association rules; HowNet