一种基于"是一个"模式的下位概念获取方法*)

刘 磊[1] 曹存根[1] 王海涛[1] 陈 威[1,2]
(中国科学院计算技术研究所 中国科学院研究生院 北京 100080)[1]
(北京理工大学计算机系 北京 100081)[2]

摘 要 在文本知识获取中,上下位关系的获取是一个基本而又关键的问题。针对基于模式上下位关系获取中遇到的下位概念获取问题,本文结合我们的研究工作,给出一种从符合"是一个"模式的句子中获取下位概念的方法,这里主要利用半自动获取的词典和句型对"是一个"模式进行分析,然后根据不同的规则,分流获取下位概念。在实验分析中,此方法显示了较好的效果。

关键词 上下位关系,概念获取,关系获取,知识获取,信息抽取

A Method of Hyponym Acquisition Based on"isa" Pattern

LIU Lei[1] CAO Cun-Gen[1] WANG Hai-Tao[1] CHEN Wei[1,2]
(Institute of Computing Technology, Chinese Academy of Sciences, Graduate School of the Chinese Academy of Sciences, Beijing 100080)[1]
(School of Computer, Beijing Institute of Technology, Beijing 100081)[2]

Abstract The research on hyponymic relations acquisition is a basic and crucial problem in the field of knowledge acquisition of text. For the hyponym acquisition in hyponymic relation acquisition based on pattern, combined with our research work, we introduce a method of hyponym acquisition based on "isa" pattern. Firstly, we make use of the dictionary and sentence patterns that have been semi-automatically obtained to analyze "isa" pattern. Then we obtain hyponyms of different quantity according to different rules. In the experiment analysis, we got a good result by using the method.

Keywords Hyponymic relation, Concept acquisition, Relation acquisition, Knowledge acquisition, Information extraction

1 前言

随着互联网的迅速发展,知识获取的来源、数量和形式发生了根本的变化。庞大的网络信息源所蕴含的海量知识为自动知识获取提出了新的挑战和迫切需求。其中上下位关系获取是一个基本而又关键的问题。上下位关系构成了知识库的基础框架,从技术实现的角度看,上下位关系获取作为将非格式化信息转换为格式化信息的重要步骤,为其它信息的获取提供了基础性支持,可以对本体、知识库、词典进行正确性检测,并对其进行扩充和完善,以用于自然语言处理系统(如机器翻译、文本理解)、信息抽取、信息检索等领域[1~3]。

在上下位关系获取中,采用的知识源多种多样,通常分为3种:结构化文本(如数据库中的数据)、半结构化文本(如词典)和自由文本(如普通 Web 网页、百科全书)[4~7]。由于当前人类知识主要以自由文本的形式表示,处理真实自由文本已成为上下位关系获取的重点研究内容。

上下位关系的半自动/自动获取方法主要有2种:一种是基于模式的上下位关系获取,主要利用语言学和自然语言处理技术,通过词法分析和语法分析获取上下位关系模式,然后利用模式匹配(Pattern Matching)发现上下位关系,这是一种理性主义(Rationalism)的方法,又称为基于规则的方法;另一种是基于统计的上下位关系获取方法,主要基于语料库和统计语言模型,利用层次聚类(Hierarchical Clustering)和非层次聚类的算法,计算概念之间的关联度,从而获取概念间上下位关系,这是一种经验主义(Empiricism)的方法。

目前上下位关系获取主要以基于模式的方法为主,其中,isa 模式是非常重要的模式之一。与英文 isa 模式相对应的中文模式为"是一个"模式。在基于"是一个"模式的句子中,下位概念的获取是比较困难的问题。为此,我们给出了一种从符合"是一个"模式的句子中获取下位概念的方法,主要利用半自动获取的词典和句型对"是一个"模式进行分析,然后根据不同的规则,分流获取下位概念。在实验分析中,此方法显示了较好的效果。

2 相关工作

概念间的上下位关系可以简单表示为二元组:上下位关系(下位概念,上位概念),这里引入 WordNet 对上下位关系(hyponymy)的定义[1]。

定义 1 概念对具有上下位关系:如果给定概念 x、y,x 的同义集合为{x, x′, …},y 的同义集合为{y, y′, …},句子:"x 是一种|类|个 y"是可以接受的,其中 x 称为 y 的下位概念(hyponym),y 称为 x 的上位概念(hypernym),记作 hyponymy(x,y)。上下位关系是类与成员关系的一种,在概念的意义(meaning)相同条件下具有传递性。

在自由文本中可以找到含有上下位关系的句子,例如:

中国/是一个/伟大的国家。 ——hyponymy(中国,国

*)自然科学基金的资助(#60273019、60573064、60573063 和#60496326)和国家重点基础研究发展计划 2003CB317008 和 G1999032701 资助项目。**刘 磊** 博士研究生,主要研究方向为关系获取、本体学习、文本挖掘;**曹存根** 研究员,博士生导师,主要研究方向为知识获取与共享、文本挖掘、智能教学;**王海涛** 博士研究生,主要研究方向为文本挖掘、自动故事生成;**陈 威** 硕士研究生,主要研究方向为文本挖掘。

家)

Linux/是一种/开放式的操作系统。 ——hyponymy(Linux,操作系统)

生物酶/是一种/良好的催化剂。 ——hyponymy(生物酶,催化剂)

许多研究人员通过分析文本语料中关系的实例,获取特定的语言模式,然后利用模式获取概念间的上下位关系。所谓模式包括句子的特殊习惯用法、句法、语义等。可以利用计算语言学和自然语言处理技术,对文本中的句子以及篇章进行分析后完成模式获取和模式匹配工作。

Hearst 是较早利用基于模式的方法研究上下位关系获取的学者之一,她认为可以从符合特定的词法-句法模式(lexico-syntactic patterns)的句子中提取上下位关系,主要利用了 4 个词法-句法模式,从 Grolier's Encyclopedia 百科全书中获取上下位关系,并与早期 WordNet(1.1 版本)进行了比较。此外,还探讨了利用已知上下位关系的概念对来提取更多词汇-句法模式的方法,但是没有给出具体的模式提取方法。Hearst 以 WordNet 作为标准,获取上下位关系的准确率是 61/106 (57.55%)。词法句法的模式示例如下,其中 NP 表示名词短语[8,9]。

…NP_1 is a NP_1… ——hyponymy(NP_1, NP_2)//isa 模式

…NP_1 such as NP_2… ——hyponymy(NP_2, NP_1) //such-as 模式

…NP_1{, NP_2} *{,} or other NP_3… hyponymy(NP_1, NP_3), hyponymy(NP_2, NP_3) //orother 模式

其他一些研究人员也尝试利用模式的方法获取上下位关系[10~12]。Moldovan 提出了一种基于模式匹配的领域概念及其概念关系提取方法。选取 5 个金融领域的种子概念,从 Internet 和 TREC-8 语料中选取了 5000 个句子作为测试集,发现了 264 个 WordNet 中没有定义的概念,同时基于 22 个不同的词汇-句法模式提取了 64 个与这些概念关联的关系[13]。Llorens 通过识别动词结构(如,"Be a kind of")来提取上下位关系。首先对文本进行词根还原、名词和动词短语识别,然后是动词结构识别,最后根据动词结构获取上下位关系[14]。

3 问题的提出和解决思路

3.1 问题的提出

从自由文本的模式匹配数量上看,在众多的上下位关系模式中,isa 模式是匹配最多的一个,但其复杂度也最高。与英文的 isa 模式相对应的中文模式为"是一个"模式,这里首先给出其定义。

定义 2 称一个句子符合"是一个"模式:如果给定句子 s,s 能与如下模式结构相匹配。

```
Define constant 常数  //定义一个常量:量词
{
    〈! 量词〉=〈种|个|名|篇|片|块|堆|群|批|章|节|环|部分|次|步|颗|套|本|条|张|幅|款|代|缕|位|卷|册|只|双|件|台|门|棵|株|朵||根|头|尾〉
}
Define pattern 是一个//定义一个模式:是一个
{
    模式:〈? C1〉〈是|为〉—〈! 量词〉〈? C2〉
}
```

其中"!"表示定义一个常量,"?"表示定义一个变量,"|"表示或者关系,"//"表示注释行,"〈? C1〉","〈? C2〉"都表示任意字符串,"〈是|为〉—〈! 量词〉"表示固定串集合{是一种,为一种,是一个,为一个,…}。若〈? C1〉中存在子串概念 c1,〈? C2〉中存在子串概念 c2,使得上下位关系 hyponymy(c1, c2)成立,则 c1 记作 $c_{下位}$,c2 记作 $c_{上位}$,称 s 为含有上下位关系 hyponymy($c_{下位}$, $c_{上位}$)的句子。例如:

{宪法规定,{约旦}$_{C下位}$}$_{〈?C1〉}$/是一个/{世袭的{阿拉伯君主立宪制国家} $C_{上位}$}$_{〈?C2〉}$

{此外,{甘露醇}$_{C下位}$ 还}$_{〈?C1〉}$/是一种/{自由基清除剂}$_{C上位}$}$_{〈?C2〉}$

{据此间媒体报道,{生物开矿技术}$_{C下位}$}$_{〈?C1〉}$/是一种/{利用微生物开矿的{湿式制铜技术}$_{C上位}$}$_{〈?C2〉}$

在利用"是一个"模式匹配后,对符合"是一个"模式的句子主要进行两步分析:(1)概念的获取和验证;(2)上下位关系的获取和验证。其中第一步如何从含有上下位关系的句子中提取上、下位概念,就是需要深入研究的问题。这主要由汉语本身的特点、知识源为通用领域自由文本、"是一个"模式的泛化程度高等原因所引起。

(1)汉语作为一种词根语,不适合直接采用其他语言处理的方法。因为[15]:

- 汉语缺乏形态变化,没有性、数、格的变化标志;
- 词序严格,词序不同,意义也随之不同;
- 词与词之间没有明显的界限,必须分词。

(2)通用领域自由文本的结构化程度很弱,表达方式灵活多样,难以给文本一个统一的结构化表示(不像词典),也不知道含有知识的范围(不像特定领域文本)。将其作为知识源进行模式匹配后,有许多非概念部分需要进一步处理。

(3)为了保证上下位获取的查全率,我们初始定义的"是一个"模式的泛化程度比较高,没有增加额外的限制规则,例如规定句子长度要小于某个阈值、〈? C1〉不能含有逗号、"的"字等。因此增加了概念获取的困难。

从中文语料的"是一个"模式匹配结果上分析,上位概念 $c_{上位}$ 的获取相对容易,例如只要获取〈? C2〉中"的"字后面的内容,效果就比较好。而下位概念 $c_{下位}$ 的获取就比较困难,其关键是在确定 $c_{下位}$ 的左右边界时,必须尽量消除可能产生的歧义(组合歧义和交叉歧义)和识别未登录词。这是下位概念获取的难点所在。

3.2 解决思路

由于"是一个"模式已经限定了下位概念所出现的语境,语境具有一定的特殊性;上位和下位概念获取后,上下位关系的获取和验证也需要下位概念所在语境的语义信息。因此,我们没有采用常规的概念获取方法,如自然语言处理技术(如句法分析),统计技术(如互信息、聚类、似然估计等),而是采用了一种比较简单的先概念外层剥离,后按规则分流的策略来获取下位概念。

所谓概念外层剥离策略是基于这样的假设:在"是一个"模式中蕴含下位概念的〈? C1〉部分,其非概念成分的构成是比较固定的。因此可以通过半自动的方式获取和分析这些非概念成分,将其转化为词典或者句型。然后利用词典和句型对〈? C1〉部分进行处理,从其左右两边向内层层剥离掉非概念部分。同时词典和句型也能提供上下位关系获取时所需要的语义信息。

所谓按规则分流的策略是指:〈? C1〉部分经过剥离处理后,剩余部分为概念的可能性,可以根据剥离程度来判断。因此我们依次利用 3 个规则:

1)〈? C1〉剥离处理后的剩余部分是否有逗号;

2)〈? C1〉剥离处理后是否有剥离标记;

3)〈? C1〉剥离处理后的剩余部分是否含有"的"字。

对剥离处理后的句子进行分析,将其分流成多组语料。含有 $c_{下位}$ 可能性高的句子被分流到一起,以方便进一步处理。从试验结果看,处理的效率和正确率都比较好。

4 下位概念的获取

根据上述研究思路,我们首先从 Web 上获取自由文本,构建训练语料和测试语料,然后利用训练语料半自动获取词典和句型,再利用剥离处理算法和分流规则对测试语料进行处理,最终获取下位概念。其获取框架如图 1 所示。

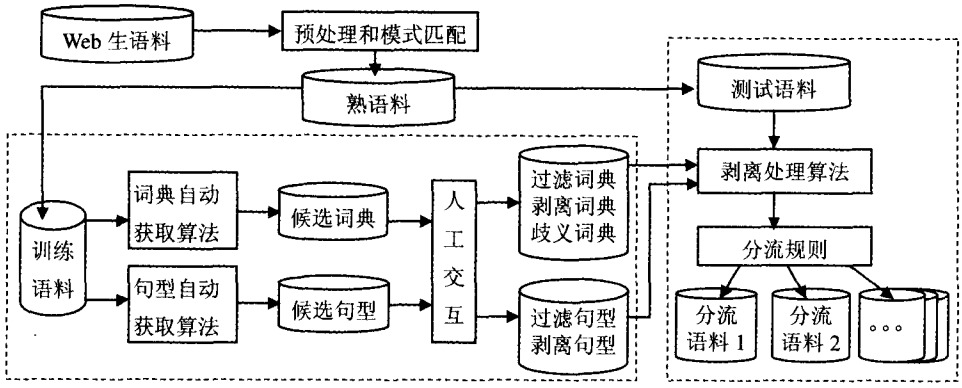

图 1 下位概念的获取框架

4.1 语料库构建

首先利用 Web 文本构建语料库,步骤如下:

(1) 从 Web 中获取网页文本,组成约 30G 生语料;

(2) 通过预处理,去掉无用字符、Web 标记,检查和纠正 Web 文本中的错误,按照标点符号(句号、问号、感叹号)断句;

(3) 利用"是一个"模式进行模式匹配,符合模式的句子构成熟语料库;

(4) 将熟语料库划分为两部分:80% 为训练语料,记作 $T_{训}$,用于词典和句型的获取;20% 为测试语料,记作 $T_{测}$,用于下位概念获取方法的验证。

4.2 词典的建立

词典根据功能的不同,分为 3 类:过滤词典、剥离词典、歧义词典。

定义 3 称词 w 为过滤词:如果对符合"是一个"模式的句子 s,词 w 满足条件:

(1)词 w 出现在〈? C1〉部分的指定位置;

(2)因词 w 的出现而使 s 中不存在上下位关系。

含有过滤词的句子称为可词过滤的,记作 $s_{w过滤}$。由过滤词组成的词典称为过滤词典,记作 $Dic_{过滤}$。

〈? C1〉含有过滤词,并不表示没有概念存在,而表示句子中上下位关系不存在,即没有上、下位概念存在。$Dic_{过滤}$ 的示例如下:

```
//位置标记 a:〈? C1〉前后部都可  h:〈? C1〉前部  t:〈? C1〉后部  u:〈? C1〉中任意位置
传说中  a  //{传说中}w过滤的天蝎座/是一只/大毒蝎
忽然 u  //谢玲眼泪未干,{忽然}w过滤又/是一个/电话
周围 t  //水面上变的阴暗了,{周围}w过滤/是一片/灰色
今天 h  //{今天}w过滤重庆又/是一个/大阴天
```

定义 4 称词 w 为剥离词:如果对符合"是一个"模式的句子 s,词 w 满足条件:

(1)词 w 出现在〈? C1〉部分的指定位置;

(2)词 w 属于〈? C1〉中的非概念部分。

含有剥离词的句子称为可词剥离的,记作 $s_{w剥离}$。由剥离词组成的词典称为剥离词典,记作 $Dic_{剥离}$。

除了考虑剥离词出现的位置,还要考虑其语义(如肯定否定、时间)对所获取上下位关系的影响,$Dic_{剥离}$ 的示例如下:

```
//位置标记:a:〈? C1〉前后部都可 h:〈? C1〉前部 t:在〈? C1〉后部
//肯否标记:+:部分肯定 -:部分否定 1:完全肯定 -1:完全否定 0:中性
//时间标记:p:过去 f:将来 c:当前
譬如   0  c//{譬如}w剥离,内存控制器通常/是一个/单独的芯片
并不   t  -1  c  //实际上,KX13{并不}w剥离/是一种/全新的芯片组
即将   t  0  f//按照规划开发的泸沽湖{即将}w剥离成/为一个/世界级旅游景区
以前   a  0  p//上海{以前}w剥离只/是一个/小渔村。
可能   t  +  c//据此推测,CETP 缺陷者{可能}w剥离是一种/复合异型合子
未心   t  -  c//美国{未必}w剥离/是一个/十全十美的国家
毫无疑问  a  1  c//P{毫无疑问}w剥离,中国作/为一个/发展中国家。
```

定义 5 称词 w 为歧义词:如果对符合"是一个"模式的句子 s,词 w 满足条件:

(1)词 w 出现在〈? C1〉部分的指定位置;

(2)词 w 有时属于〈? C1〉所含下位概念的一部分,有时属于非概念部分。

含有歧义词的句子称为有歧义的,记作 $s_{w歧义}$。由歧义词组成的词典称为歧义词典,记作 $Dic_{歧义}$。

歧义词出现于概念的左右两边,造成歧义的词,多为单字词,需要进行消岐处理。歧义词可以看作是一种特殊的剥离词,Dic 歧义的示例如下:

```
//位置标记:a:〈? C1〉前后部都可  h:〈? C1〉前部  t:〈? C1〉后部
真  t  //  中国{真}w歧义/是一个/伟大的国家
       //  互调失{真}w歧义/是一种/测量非线性失真的方法
```

Dic 过滤、Dic 剥离、Dic 歧义是通过半自动的方法获取的,这里给出其简要的获取步骤:

(1) 读入语料 $T_{训}$,截取每个句子的〈? C1〉部分。

(2) 以逗号、顿号等标点为分隔符将所有句子的〈? C1〉分隔,分隔结果记作 S={s1,s2…sn},任意 s∈S 都是〈? C1〉的分隔串。

(3) 对S中的所有分隔串，两两对串的开头和结尾部分求最长公共串。例如，对任意 $s_1, s_2 \in S$，$s_1 = a_1 a_2 a_3 \cdots a_n$，$s_2 = b_1 b_2 b_3 \cdots b_m$（a，b 表示单个字符），若存在 $a_1 a_2 \cdots a_i = b_1 b_2 \cdots b_i$，$a_{i+1} \neq b_{i+1}$，$i \leq n$ 且 $i \leq m$，则称 $a_1 a_2 \cdots a_i$ 为（s_1，s_2）的公共前串；若存在 $a_{n-j} \cdots a_{n-1} a_n = b_{m-j} \cdots b_{m-1} b_m$，$a_{m-j-1} \neq b_{m-j-1}$，$j < n$ 且 $j < m$，则称 $a_i \cdots a_{n-1} a_n$ 为（s_1，s_2）的公共后串。

(4) 将所求公共串加入到候选词典 $Dic_{候选}$ 中，并记录位置(前串|后串)和出现次数。

(5) 对 $Dic_{候选}$ 的公共串按规则自动筛选(串的长度＜规定阈值，串的次数＞规定阈值)。

(6) 通过人工交互方式，$Dic_{候选}$ 划分为 $Dic_{过滤}$、$Dic_{剥离}$、$Dic_{歧义}$，增加缺少的标记。处理原则：可以保留组合词(如组合词"并不单纯"由"并不"和"单纯"组成)，这会减少歧义，也能提高处理效率；单字词尽量放入 $Dic_{歧义}$；词的前后位置要区分，有些词在前置时为过滤词，在后置时却为剥离词。

最终获取词典结果：$Dic_{过滤}$ 含词 1880 个，$Dic_{剥离}$ 含词 3516 个，$Dic_{歧义}$ 含词 242 个。

4.3 句型库建立

利用词典只能处理〈? C1〉结构比较简单的情况。对于结构复杂的〈? C1〉，则需通过句型去掉非概念部分。例如一些时间、条件短语需要句型来识别。根据功能不同分为两类：过滤句型库和剥离句型库。由于句型出现歧义的情况比较低，没有划分出歧义句型库。

定义 6 称句型 p 为过滤句型：如果对符合"是一个"模式的句子 s，句型 p 满足条件：

(1)句型 p 能与〈? C1〉相匹配；

(2)因句型 p 的出现而使 s 中不存在上下位关系。

含有过滤句型的句子称为可句型过滤的，记作 $s_{p过滤}$。所有过滤句型的集合称为过滤句型库，记作 $Pat_{过滤}$。

$Pat_{过滤}$ 的示例如下：

```
define pattern 过滤句型 011
{
  句型:〈? w1〉〈按|依据|按照|依照|根据|据|经|从|经过|依|
  通过|已有|有|有关|国内|国外|国内外〉〈? w2〉〈传闻|估计|
  猜想|预测|预计|预言〉〈? w3〉
  基本规则:notcontain(〈? w2〉,〈,|。|.|? |! |,|.|? |! |;
  |;))
  //notcontain(a,b)表示 a 中不含有 b 的内容,〈? w1〉〈? w2〉〈?
  w3〉表示任意串
}
//例句:{据有关专家预言}p过滤,新的世纪将//创新策划的时代
```

定义 7 称句型 p 为剥离句型：如果对符合"是一个"模式的句子 s，句型 p 满足条件：

(1)句型 p 能与〈? C1〉相匹配；

(2)句型 p 所包含的部分为〈? C1〉非概念部分。

含有剥离句型的句子称为可句型剥离的，记作 $s_{p剥离}$。所有剥离句型的集合称为剥离句型库，记作 $Pat_{剥离}$。

$Pat_{剥离}$ 的示例如下：

```
    define pattern 剥离句型 016
    {
        句型:〈? w1〉〈如|就如|正如|恰如|正像|像|就像|正向
|正像〉〈? w2〉〈评价|所述|所说|一样|所言|所示|所讲|所见〉〈,
|。|.|? |! |;|;〉〈? w3〉
        基本规则:notcontain(〈? w2〉,〈,|。|.|? |! |,|.|?
|! |;|;))
    }
    //例句:{正如页框一样}p剥离表格也/是一种/容器对象
```

$Pat_{过滤}$ 和 $Pat_{剥离}$ 的获取方法与词典的获取比较类似，其简要获取步骤如下：

(1) 读入语料 $T_{训}$，截取每个句子的〈? C1〉部分。

(2) 以逗号、顿号等为分隔符，将所有句子的〈? C1〉分隔，分隔结果记作 $S = \{s_1, s_2, \cdots, s_n\}$，任意 $s \in S$ 都是〈? C1〉的分隔串。

(3) 对S中的所有分隔串，两两对串求公共子串。若满足以下条件，则可以放入候选句型 $Pat_{候选}$。①两串的公共子串数目＞=2；②公共子串出现顺序一致，不能交叉；③存在一个公共子串位于串的开始或者结尾；例如，对任意 $s_i, s_j \in S$，$s_i = a_1 a_2 a_3 \cdots a_n$，$s_j = b_1 b_2 b_3 \cdots b_m$，若只存在 $a_1 = b_2$ 一个公共子串，则不满足条件①；若存在 $a_1 a_2 = b_4 b_5$，$a_5 = b_2$，虽满足①，但出现顺序出现交叉，不满足条件②；若存在 $a_2\ a_3 = b_2\ b_3$，$a_{n-1} = b_{m-2}$，虽满足①、②，但不满足条件③；若 $a_1 a_2 = b_2 b_3$，$a_{n-1} = b_{m-2}$ 则满足所有条件，记作句型：〈? w_1〉〈$a_1\ a_2$〉〈? w_2〉〈a_{n-1}〉〈? w_3〉。

(4) 将所得到句型加入到候选句型库 $Pat_{候选}$ 中，并记录出现次数。

(5) 对 $Pat_{候选}$ 的句型按规则自动筛选(公共串长度＜规定阈值，句型次数＞规定阈值)。

(6) 通过人工交互方式，$Pat_{候选}$ 划分为 $Pat_{过滤}$、$Pat_{剥离}$，并根据句型效果增加额外规则。

(7) 对相似的句型进行合并和泛化。

最终获取句型结果：$Pat_{过滤}$ 含句型 28 个，$Pat_{剥离}$ 含句型 85 个。

4.4 下位概念获取算法

根据已经建立的词典和句型库，按先概念外层剥离、后规则分流的策略给出概念获取的算法如下：

Step1：读入词典、句型库、测试语料。

Step2：从测试语料中顺序读入未处理句子。若存在未处理句子，则按照 Step3～Step5 处理；若不存在未处理句子，则转 Step6。

Step3：利用 $Dic_{过滤}$、$Dic_{剥离}$、$Dic_{歧义}$、$Pat_{过滤}$、$Pat_{剥离}$ 对句子的〈? C1〉部分进行标记处理。

(i) 利用 $Dic_{过滤}$，从〈? C1〉指定位置(前部、后部、任意)查找是否含有过滤词。如果有，则增加过滤词标记，然后转 Step4；如果没有，则转(ii)处理。

(ii) 利用 $Pat_{过滤}$，从〈? C1〉查找是否含有能匹配的过滤句型。如果有，则增加过滤句型标记，然后转 Step4；如果没有，则转(iii)处理。

(iii) 利用 $Pat_{剥离}$，从〈? C1〉查找是否含有能匹配的剥离句型。如果有，则增加剥离句型标记，这里〈? C1〉中匹配的剥离句型可能会有多个。无论〈? C1〉是否含有剥离句型，都转(iv)。

(iv) 依次利用 $Dic_{剥离}$、$Dic_{歧义}$、$Dic_{过滤}$ 词典，按词的长度优先原则，对〈? C1〉的未标记部分从前部和后部由外向内，递归查找，层层标记。当满足如下条件之一时停止：(a)发现过滤词时停止；(b)连续找到两个歧义词时停止，这时只标记第一个歧义词；(c)不能再发现任何标记词。

Step4：根据句子中〈? C1〉部分所含有的标记，按照如下原则处理：

• 若〈? C1〉中含有过滤词或过滤句型标记，直接将句子分流到过滤语料。

• 若〈? C1〉完全被剥离词、歧义词、剥离模式所标记，不含有任何未标记部分，则将句子分流到完全标记语料。

• 若〈? C1〉中前部或后部标记的最内层为歧义词标记，则对〈? C1〉进行分词；如果歧义词与其它词分到一起，则去掉歧义词标记；如果歧义词被单独分开，则认为歧义词标记是一种剥离标记，不再有歧义。

Step5：根据3个分流规则：(i)〈? C1〉未标记部分是否含有逗号；(ii)〈? C1〉是否有标记；(iii)〈? C1〉未标记部分是否有"的"字；将语料划分为5部分：

- 无逗号，无标记，无的字——分流语料
- 无逗号，无标记，有的字——分流语料
- 无逗号，有标记，无的字——分流语料
- 无逗号，有标记，有的字——分流语料
- 有逗号——分流语料

Step6：最后所有的分流语料都作为候选概念语料，用于以后的概念验证。

5 试验结果及分析

5.1 试验结果

我们选取的测试语料 $T_{测}$ 的句子规模为23万。经过上述算法处理后，语料被分流为多个部分。对分流语料的评价主要依据3个评价指标：查全率(Recall)、查准率(Precision)、剥离正确率。

查全率＝分流语料内含有下位概念的句子数目/总语料含有下位概念的句子数目

查准率＝分流语料内含有下位概念的句子数目/分流语料所有句子数目

剥离正确率＝分流语料中含有下位概念的句子中〈? C1〉剥离正确的句子数目/分流语料中含有下位概念的句子数目。

其详细数据统计如表1所示。

表1 测试语料处理结果分析

输入	词典：过滤词：1880个　剥离词：3516个 歧义词：242个				
	句型：过滤句型：28个　剥离句型：85个				
	测试语料：235625 句				
输出结果	分类语料	数目(比例)	准确率	召回率	剥离正确率
	总语料	235625(100%)	50.6 %	100 %	/
	过滤掉语料	39028(16.5%)	2.5 %	0.8 %	/
	完全标记语料	14758(6.3%)	1.8 %	0.2 %	/
	部分标记和无标记语料	181839(77.2%)	/	/	/
	无逗号	155593(66.0%)	/	/	/
	无逗号－无标记	55081(23.4%)	/	/	/
	无逗号－无标记－无的	48162(20.4%)	86.3 %	34.9 %	98.9 %
	无逗号－无标记－有的	6919(2.9%)	55.2 %	3.2 %	74.6 %
	无逗号－有标记	100512(42.7%)	/	/	/
	无逗号－有标记－无的	78717(33.4%)	75.2 %	49.6 %	93.0 %
	无逗号－有标记－有的	21795 (9.2%)	36.5%	6.7 %	49.6 %
	有逗号	26246 (11.1%)	20.5 %	4.5 %	/

5.2 结果分析

(1)过滤掉语料和完全标记语料

这两部分语料都不再进一步处理。从表1中查全率(0.8%和0.2%)和查准率(2.5%和1.8%)可以看出，这两部分语料中含有下位概念的句子非常少。分流出这两部分语料，对提高其他分流语料的质量起到了重要作用。例句如下：

过滤语料：
电梯{左边}$_{w过滤}$/是一个/大工作室
中国二十多年的改革走{的就}$_{w过滤}$/是一条/先探索后规范的道路
完全标记语料：
{当然}$_{w剥离}${现在}$_{w剥离}${还}$_{w歧义}${只}$_{w歧义}$/是一种/意向
{其实}$_{w剥离}${，许多问题归结起来，}$_{p剥离}${还}$_{w歧义}$/是一个/是否尊重投资者权益的问题

(2)无逗号－无标记－无的字语料

这部分语料的质量是最好的，其查全率和查准率分别达到了34.9%和86.3%，而且〈? C1〉的剥离正确率98.9%，即为100个含有下位概念的〈? C1〉，有约99个〈? C1〉直接就是没有非概念部分的下位概念。例如：

联合防区外武器/是一种/无动力滑翔武器

氢氟酸/是一种/强烈的腐蚀剂

(3)无逗号－无标记－有的字语料

此部分语料含有上下位关系的查准率也较高，但是必须对〈? C1〉做进一步处理，确定下位概念是否存在，是位于"的"字之前还是之后，可通过对〈? C1〉为标记部分进行内部词法分析来判断。〈? C1〉主要包含3类结构：

(i)〈名词短语〉〈的〉〈! 名词短语〉
例：{伦敦}$_{名词}$的{牛津街}$_{名词}$/是一条/非常繁华的商业街道
上下位关系为 hyponymy(牛津街，商业街道)，"伦敦"对"牛津街"起限定作用。
(ii)〈! 形容词短语〉〈的〉〈! 名词短语〉
例：{美丽富饶}$_{形容词}$的{海南}$_{名词}$/是一座/历史悠久的岛屿
上下位关系为 hyponymy(海南，岛屿)，"美丽富饶"对"海南"起修饰作用。
(iii)〈! 名词短语〉〈的〉〈动词短语〉
例：{数据仓库技术}$_{名词}$的{开发}$_{动词}$/是一个/复杂的过程
这时〈? C1〉常表示一种过程。虽然〈? C1〉含有概念，但没有上下位关系存在。

(4)无逗号－有标记－无的与无逗号－有标记－有的语料

这两部分语料的〈? C1〉都含有剥离标记。剥离标记是通过 $Dic_{剥离}$、$Dic_{歧义}$、$Pat_{剥离}$ 处理得到的，下位概念的获取直接受剥离标记的影响。在无逗号－有标记－无的语料中，剥离正确率达到93.0%；在无逗号－有标记－无的语料中，剥离正确率达到49.6%。例如：

效果好的剥离：
{研究已发现，}$_{p剥离}$巯基乙酸的{的确}$_{w剥离}$/是一种/慢性致癌物质
{例如}$_{p剥离}$清末的任伯年{便}$_{w歧义}$/是一位/雅俗共赏的大画家
{不过}$_{w剥离}${，}$_{w剥离}${和{所有海洋性气候下的}$_{p剥离}$城市一样，}$_{p剥离}$诺丁汉{也}$_{w歧义}$/是一座/多雨的城市
效果不好的剥离：
地中海式饮食虽然含脂肪相对较高{，}$_{w剥离}${但仍}$_{w剥离}${不失}$_{w剥离}$/为一种/保健食谱
{{与软件公司及软件业面临的}$_{p剥离}$情况有所不同，}$_{p剥离}$IT硬件业正以较快的速度{成}$_{w歧义}$/为一个/微利行业
信息电话是现{有普通电}$_{p剥离}$话机的升级换代产品{，}$_{w剥离}${也}$_{w歧义}${可以看}$_{p剥离}${作}$_{w歧义}$/是一种/功能非常丰富的新型电话机

出现效果不好的剥离主要是因为〈? C1〉的结构比较复杂，在没有进行语法分析的基础上，难以剥离正确，而且剥离词典和句型不可能做到非常完善，剥离时还存在剥离冲突和歧义问题。对〈? C1〉的未标记部分，可以按照(2)、(3)语料的方式继续分析处理。

(5)有逗号语料

由于〈? C1〉的复杂，只用剥离词典和句型处理不了。〈? C1〉非标记部分仍然含有逗号等分隔符号。在这种情况下，再按照前面的有无标记、有无的字划分语料就没有意义了。虽然这部分语料占的比例非常小，而且查全率(4.5 %)和查准率(20.5%)也不高，但是仍有一些含有上下位关系的句子。例如：

新加坡环境清洁美丽，管理井然有序，社会安定文明/是一个/现代化的城市国家

亚彻一行几经辛苦终于到达了费展，费展/是一座 /因矿业而发达的都市

针对这种情况，目前我们考虑以逗号为分隔符，将〈? C1〉分隔成多个部分。每个部分与〈? C2〉组成分隔句，然后对分隔句重新执行 Step3～Step5，对每个分隔句进行剥离处理。这里可以召回一些失去的下位概念。例如：

原句：亚彻一行几经辛苦终于到达了费展，费展/是一座/因矿业而发达的都市

分隔句 1：亚彻一行几经辛苦终于到达了费展/是一座/因矿业而发达的都市

分隔句 2：费展/是一座/因矿业而发达的都市

在所有分流语料中，无逗号－无标记－无的语料和无逗号－有标记－无的语料是最重要的语料，两者之和占总语料的 53.8%，查全率为 84.5%，查准率(86.3 %和 75.2 %)、剥离正确率(98.9 %和 93.0 %)也都非常高。因此上下位关系可以先从这部分语料中获取和验证，然后利用所获取上下位关系帮助其他分流语料中上下位关系的获取和验证。

存在问题与进一步工作 作为上下位关系获取研究的一部分，本文主要讨论了一种从符合“是一个”模式句子中获取下位概念的方法。该方法通过半自动获取的词典和句型对“是一个”模式进行分析，根据不同的规则，分流获取下位概念，实验结果令人满意。该方法还可用于其它上下位关系模式的概念获取，例如“such as”、“or other”等模式。但仍然存在一些需要解决的问题，例如标记时歧义和冲突的处理、词典和句型语义信息的合理使用等。在下位概念获取的基础上，我们将结合一些方法(如词法分析、网页 html 标记、已获取概念等)对下位概念做进一步验证。

参 考 文 献

1 Miller G. WordNet: An On-line Lexical Database. International Journal of Lexicography, 1990, 3(4)
2 Beeferman D. Lexical discovery with an enriched semantic network. In: Proceedings of the Workshop on Applications of WordNet in Natural Language Processing Systems, ACL/COLING, 1998
3 Richardson S D, Dolan W B, Vandervende L. Mindnet: acquiring and structuring semantic information from text. In: Proc. of COLING-ACL'98, 1998. 1098～1102
4 Cao Cungen, Shi Qiuyan. Acquiring Chinese Historical Knowledge from Encyclopedic Texts. In: Proceedings of the International Conference for Young Computer Scientists, 2001. 1194～1198
5 Dolan W, Vanderwende L, Richardson S D. Automatically Deriving Structured Knowledge Bases From On-Line Dictionaries. In: Proceedings of the Pacific Association for Computational Linguistics. Vancouver, British Columbia, 1993. 5～14
6 Shinzato K, Torisawa K. Acquiring hyponymy relations from web documents. In: Proceedings of HLT-NAACL 2004. 73～80
7 宋柔，许勇. 基于语义的百科辞典知识提取实验. Computational Linguistics and Chinese Language Processing, 2002, 7(2): 101～112
8 Hearst M A. Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th International Conference on Computational Linguistics. Nantes, France, 1992
9 Hearst M A. Automated Discovery of WordNet Relations. To Appear in WordNet: An Electronic Lexical Database and Some of its Applications, Christiane Fellbaum (Ed.). MIT Press, 1998
10 Imasumi K. Automatic acquisition of hyponymy relations from coordinated noun phrases and appositions: [Master's thesis]. Kyushu Institute of Technology, 2001
11 Morin E, Jacquemin C. Automatic acquisition and expansion of hypernym links. Computer and the Humanities, 2003
12 Ando M, Sekine S, Ishizaki S. Automatic extraction of hyponyms from newspaper using lexicosyntactic patterns: [IPSJ SIG Technical Report 2003-NL-157]. 2003. 77～82
13 Moldovan D, Girju R, Rus V. Domain-Specific. Knowledge Acquisition from Text. In: Proceedings of the sixth conference on Applied natural language processing. Washington, 2000. 268～275
14 Lloréns J, Astudillo H. Automatic generation of hierarchical taxonomies from free text using linguistic algorithms. In: Advances in Object-Oriented Information Systems, OOIS 2002 Workshops, Montpellier, France, Lecture Notes in Computer Science 2426, 2002. 74～83
15 张春霞，郝天永. 汉语自动分词的研究现状与困难. 系统仿真学报，2005, 17(1): 138～143

(上接第 106 页)

(2)ListBox，当控件只允许选择一个值时，与(1)同；当允许选择多值时，其值以一个集合传送，特别当 ListBox 没有选中值时，“form”提交串中没有该控件的位置；

(3) CheckBox/CheckListBox 其值传送非常特别，在“form”提交串中按照检查框的选项被选中的个数占位：ID 串按照格式“ID:序号”占位，值以符号“on”占位，最后加一个“List”符号结束。

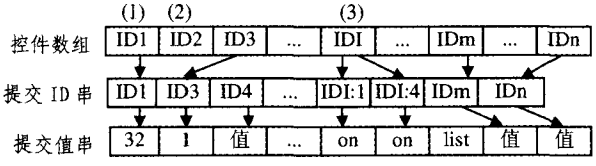

图 4 “form”提交串分析示意图

根据以上分析，本文作者通过在 asp. Net 组件的 Page 对象的 ViewState 属性上记录后台应用数据结构的格式串，对 Form 体提交的格式流设计一个“标尺”算法，成功地实现了动态页面后台数据与前台界面操作的通讯，在系统表示层和应用层的可视组件中大量运用了该算法，如通用编辑组件、查询组件等。

结束语 通过对 WEB MIS 进行应用数据与系统功能松散耦合设计构建系统框架结构，一方面提高了 Web 系统设计的集成度，显著提高系统的稳定性和可靠性；另一方面，由于采用了组件技术和面向对象方法，组件的重用度可达到 70%左右，在框架系统的支持下，可以实现应用系统的半自动/自动生成，大大提高了系统开发效率。本文涉及的 Web 信息开发平台的基本组件已在. NET 环境下全部完成，并在此应用平台下生成了一个仓库群的 Web 应用系统，其运行效率与常规开发的 Web 系统运行效率基本一致，而在系统稳定性和扩展性上都有显著的提高，具有较高的应用价值。

参 考 文 献

1 孙宏伟，张树生，王静. 组件化松耦合企业应用集成环境关键技术研究[J]. 计算机应用，2002, 22(4): 4～8
2 赵会群，王国仁，高远. 软件体系结构抽象模型[J]. 计算机学报，2002, 25(4): 730～736
3 舒忠梅，左亚尧. 软件体系结构与组件技术[J]. 微机发展，2002, 4: 31～33
4 Leinecker R C 著. COM+技术大全. 高智勇，等译. 北京：机械工业出版社，2001. 16～30