机器翻译中的智能规则系统

程学旗 唐泓英 姚天顺
（东北大学计算机科学研究所）

摘要：机器翻译系统(MTS)中，规则知识是必不可少的支撑部分.由于自然语言现象的复杂多样性，规则知识的爆炸和规则库的完善管理成为机译系统中的瓶颈问题. 本文基于知识的分层思想提出了一种智能型的规则库系统.该系统具有自我完善和自我学习的机制.它实质上是一种有效的自学习式的知识库系统.

关键字： 机器翻译　　智能规则系统　　自学习

An Intelligent Rule System In Machine Tanslation

Cheng Xue-qi ,Tang Hong-ying, Yao Tian-shun
(Department of Computer Science, Northeastern University)

ABSTRACT:
Rules are important Knowledge in a machine translation system(MTS). Because of the complexion of the natural language phenomena, the explosion of rule knowledge and the updating of rule base have become the bottlenecks in MTS. This paper , based on the hierarchical idea, proposes a model of an intelligent rule base system(IRBS), Using the check-reorganization techniques and learning algorithm,the system has the feature of self-learning and self-perfection. As the paper will show, the system is effective.

KeyWords: Machine Translation , Intelligent Rule System, Self-learning

1 引　言

机器翻译系统中，由于知识数量庞大,可靠性不一,难以归纳总结等原因,使规则库的完善和维护相当困难.目前各种实用或实验的机译系统中的规则管理都多少存在一些问题:

规则膨胀.规则库庞大,不好管理;

规则知识的更新和添加困难;

规则库内部的各规则之间优先级不当，存在矛盾和冗余等现象等，从而影响机译质量．

针对这些问题，人们提出了很多解决的办法，但效果往往不太好，并因此还会产生新的问题．本文针对我们的汉英双向翻译系统(CETRAN)的特点，设计了一种基于分层的智能规则库系统模型．该模型采用信息分层分流机制解决了因规则膨胀导致的问题．它最大的特点是带有智能的自我完善机制和自学习机制，能通过对加标的实例的学习来完善规则库．

2 规则体系及其形式化描述

规则系统采用信息分层分流机制，将整个规则系统划分成一些相对独立的规则子库．根据自然语言的特点，我们机译系统的翻译处理过程基本上按如下顺序进行：

源语言-->分词处理-->兼类处理-->词法分析-->句法分析-->

中间语形成-->线性化处理-->时态处理-->语序调整-->词形变换-->目标语生成

这个过程从源语言的表层一步步深入，直至构造成深层的语义网络，然后表达成独立于任意语种的中间转接语，再一步步构造至目标语的表层．我们的系统是基于规则驱动的，也就是说系统的每一步处理都是由规则制导的．翻译过程中的每一步都可表示为一个语言现象层．我们用来表达这些语言现象的规则知识也可对应划分成不同的层次．更进一步的同一层的规则知识又可据当前中心节点的词性再细划分．这就是我们所说的信息分层分流的思想．应用这种机制，系统中每个具体的规则库文件表达的是一具体的语言现象层内某个具体的词性节点所体现的一些语言现象．它们是静态独立的．在翻译过程中，系统的控制器按程序流程和当前被处理的语言单位的具体节点来检索具体的规则库文件．这样一个完整句子各个层次的语言现象是由控制器在动态翻译过程中匹配上的规则形成的规则流来体现的．规则体系的这种组织上的静态性、独立性和翻译过程中的动态语义相关性，不仅很方便地表达了完整的语言现象，而且很好地解决了规则膨胀问题，同时这又是我们建立智能规则系统的前提．关于规则体系的分层分流技术的详情请见文献[2]．

具体的每条规则的描述形式很简单，它实质上是一个条件、操作对：

R: P ==> M

P :- ∧Pi

M :- ∧Mj

P是规则的前件，由一些谓词公式Pi合取而成．M是操作序列，是一些动作Mj的合取．关于具体的规则描述语言的描述，请参阅文献[1]．

整个规则体系可形式地描述如下：

RB = ∪LRBi　　i=1,...,n;　　(n为约定的语言现象层数目)

LRB = ∪CLRBj　　j=1,...,m;　　(m为约定的自然语言词性的个数)

CLRB= ∪R　　(R为具体的一条规则)

由上可知n和m的值固定，我们完善规则库实际上就是完善每一个CLRB．我们的目标是使每一个初始的CLRB达到或趋向达到一理想的境界CLRB*，这种理想的规则库应该知识完备、无矛盾、无冗余．优先级正确．我们通过规则库的自我调整和学习机制来实现．

CLRB ---> CLRB*

理论上讲,只要语言现象有限,CLRB * 是可达的.

3 智能规则库系统模型及实现

规则库系统模型如下图所示:

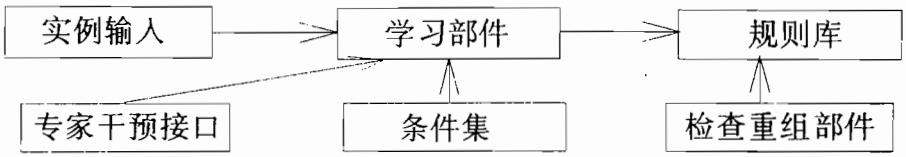

图1. 规则库系统模型

图1中"-->"表示数据流向.系统的最终目标是完善规则库, 主要包括对规则库内的规则进行重组调整和添加新的规则.系统中,这两种功能基本上都是由机器来智能实现完成.

3.1 概念定义

为了完整地描述规则系统的实现,我们先阐述几个概念:

一. 状态空间 (state space)

在翻译的每一个处理层次,被处理的语言单位都具有一种或多种初始状态(Initial State)、中间状态(Middle State)和终止状态(Final State). 所有这些状态总和我们称为整个翻译的状态空间(State Space).机译实际上就是利用规则信息驱动,将语言单位从源语初始态一步一步构造直至目标语的终止状态.

二. 条件集 (Condition Set)

规则的前件包括以下两方面的信息:

1. 节点本身的属性(如当前节点及相邻节点的词性CCAT,词法信息MOR,语义码REST等);

2. 节点之间的语义格关系(如施事AGT.受事OBJ,等格关系).

所有这些条件的总和我们称之为条件集(Condition Set, CS *):

$$CS^* = A^* \cup R^*$$

$$A^* = \cup Ai \quad (Ai是具体的属性)$$

$$R^* = \cup Rj \quad (Rj是格关系)$$

针对某一具体的规则库(CLRB)而言,它的条件集CS是CS^*的一个子集.我们定义为:

$$CS = F(i) \circ G(j) \circ CS^* \quad (i为规则系统的层次标志,j为具体的词性编号)$$

此处函数F和G是用来求子集的,它们分别与具体的层次和词性有关.这些函数是预先给出的,它们是一种经验函数.每一个CLRB 的条件集(CS)又可描述成如下的n元组:

$$CS = \langle CS1, CS2, \ldots CSn \rangle \quad CSi为某一类属性或格关系$$

这样，每个规则库(CLRB)对应的条件集很小.如针对我们机译系统中的名词兼类规则库(dkind_n).它的条件集只包含当前及左右节点的词性，下位词性，语义码等.可表示如下：

CS=<^CCAT,^SUBCAT,^REST,1^CCAT,1^SUBCAT,1^REST,^1CCAT,^1SUBCAT,^1REST>

三．动作函数（Action Function）

规则的后件是一个动作函数(M).翻译的规则驱动，实际上就是被处理的语言单位在某一处理层次，当规则的前件为真时，经动作函数作用于初始态，最后构造至终止态.

$$Sbegin \xrightarrow{M} Send$$

3.2 规则库系统的实现

如图1所示，条件集部件是相对静止的，它在一次学习过程中是预先给出的.我们智能规则库系统实现的关键是完成检查重组部件和学习部件的构造．下面我们将分别讨论之：

3.2.1 检查重组部件的实现

我们可以从单个CLRB的角度来考虑检查重组部件的实现.从三个方面来论叙：

一．规则间矛盾的消除：

我们认为状态之间的构造应是一种最优构造，其解是确定的．这样虽然有时得不到最好的解，但不需回溯，系统更加实用．正因如此，如果在同一规则库(CLRB)中存在如下情况：

P => M1

P => M2

系统认为存在矛盾，能利用实例来去伪存真.

二．冗余的消除

若CLRB中存在如下两条规则：

P1 => M (1)

P2 => M (2)

动作函数相同而前件不同，但前件之间是包容关系，即P1是P2的子集，则存在冗余.

三．优先级的调整

本质上讲，规则的分层组织实际上就是对规则确定了优先级，但这是一种粗分．对同一个规则库CLRB内部而言，规则之间也存在优先级问题.我们从规则的前件部分入手：

<1> 条件的约束从属关系

当两条规则的后件不同而前件存在从属关系时：

若 $P1 \wedge P2 \Rightarrow M1$ (1)

$P1 \wedge P2 \wedge P3 \Rightarrow M2$ (2)

则 (2)的优先级高于(1).考虑如下两条规则:

```
^'本'+CCAT.n    => ~^1,1^\^__(SEMRELA.MOD)          (1)
CCAT.m+^'本'+CCAT.n => ~1^,^1\^__(SEMRELA.NUM)      (2)
```

有两个短语:"一本书","本国".我们知道"一本书"的"本"是一个量词,而"本国"中是用来修饰"国"的.如果以上的两条规则中(1)的优先级高于(2),则这两个短语都将匹配上规则(1),此时无法判定第一个短语中的"本"是书的量词.反之这两个短语都会被正确理解.

<2> 属性的语义从属关系

若 P1∧P2 => M1 (1)

P3∧P2 => M2 (2)

且P3是P1的下位属性或P3是P1的值,即P3与P1从语义上讲是具体与抽象,特例与泛例的关系,则(2)的优先级应高于(1).让我们看如下两条规则:

```
n+'的'+^n => !1^,1^\^__(SEMRELA.MOD)          (1)
(nn2)+'的'+^n => !1^,1^\^__(SEMRELA.POS)      (2)
```

针对两个短语"天津的包子"和"教授的书".我们知道教授拥有书(语义格关系是POS),而天津是修饰包子的(语义格关系是MOD).因为教授和天津都是名词(CCAT.n),不同之处在于前者的下位词性是nn2(CASUBCAT.nn2). 此时若规则(1)优先于规则(2),则分析会出错.

<3> 属性的语义先后关系

我们翻译器中分析是从表层到深层,生成是从深层到表层的构造.这样反映不同层次的属性之间也存在着先后关系.如在分析时反映词法信息的属性应优先于反映句法信息的属性.我们看如下两条规则:

```
v-^(CCAT.v,MOR.kxn)+n => ~1^,^1\2^__(SEMRELA.MOD),^1\^_ (SEMRELA.OBJ)   (1)
v+^(CCAT.v,SYN.pon)+n => ~1^,^2\^1__(SEMRELA.OBJ),^1\^__(SEMRELA.CON)    (2)
```

这两条规则都是处理 "动词1+动词2+名词" 这种情况的. 规则(1)表示当动词2可以修饰名词时(词法属性是mor.kxn),则后面的名词作为动词1宾语,动词1修饰该名词. 规则(2)表示当动词2可以带名词宾语时(句法属性是syn.pon),后面的名词作为该动词的宾语,然后这两个动词形成一个连动关系. 由于(1)的前件中包含词法信息而(2)的前件中包含句法信息,所以(1)的优先级应高于(2).

<4> 语义格的先后关系

我们知道在标准的英语中时间状语应位于地点状语之后,类似这种规律很多.也就是说反映语法信息的语义格关系也应有一种被处理的先后顺序.我们举例说明:

下面是一个由中间语表达的深层理解结构

```
! 0: ( SENTENCE,,,,SENTENCE)
  !--SEN-->1: ( live,v,vv1)
          !--EXP-->2: ( we, r, rr1)
          !--LOC-->3: ( Beijin,n,nn2, in)
          !--TIM-->4: ( 1994,n,nn3,in)
```

经过线性化处理之后,理想的形式应是: We live in Beijin in 1994

它主要是匹配两个规则:

^(CCAT.v),^__(SEMRELA.TIM) =>^__\^1　　(1)

^(CCAT.v),^_ (SEMRELA.LOC) =>^ \^1　　(2)

如果规则(2)优先于规则(1)则会出现错误结果:　We live in 1994 in Beijin

3.2.2 学习部件

学习部件主要是通过对输入实例的学习产生新的规则知识.它的输入是一组语言单位在某一层次的初始状态和结束状态对:E=<Sbegin,Send>. 输出是规则或关于规则的提示.

学习是从不同的层次来进行的,它是一个逐步求精过程.在某一层的翻译过程实际上是将语言单位从初始态构造至结束态.这种构造可分两种情况.一种是初始态中只有一个节点经规则驱动,一步构造至结束态.我们称之为一步构造(One-Step Construction);另一种是初始态中多个结点依次匹配上各自的规则,经历多个中间状态的构造,直到结束态.我们称之为多步构造(Multi-Step Construction).下面我们将分别叙述:

3.2.2.1 一步构造(One-Step Construction)

一步构造的算法(OSCA)如下:

(1).接收一组反应同类语言现象的实例 E1,E2,,,En.　Ei=<Sbegin,Send>;

(2).确定规则学习的层次同时确定将要学习的规则库及条件集:

CS=<CS1,CS2,...,CSn>;

(3).由Sbegin--->Send 求出的动作函数 M .除去 M 异常的实例,优化学习的例子空间,使得 E=< E1,E2,....Em>;

(4).求出Sbegin在CS上的映射.然后概念抽取,找出将要产生的规则的前件P.即

P=Π(Si★CS) ,(i=1,...,m), '★'用于映射运算,'Π'是概念抽取函数.

(5).利用 P => M,对实例空间E进行验证,同时修改使更精确.

(6).利用检查重组部件检查 P => M.确定是否添加到规则库中或给出产生新规则的提示,由人工参予修改,产生新的规则添加到规则库中.

该算法中最关键的是第4步.我们学习的质量由新产生的规则的前件P的质量决定.一次优质的学习.P中应包含了该类语言现象的所有必要前提,且冗余信息特别少.同时学习算法的实现不应太繁.所以在算法(OSCA)中,对条件集CS空间的选取和规则的前件P的概念抽取的方式很重要. 为了简化学习,我们实际系统(CETRAN)中,CS主要选取当前节点,左右节点以及当前节点的父子节点的词性(CCAT). 语义编码REST),格关系等为其属性元素.经我们对实际规则系统的统计发现. 80%以上的共性规则的前件都满足这种条件, 系统中对概念的抽取基本上采取语义聚类的方式.如针对两个词: 老师 (CCAT,n.REST.11112), 学生(CCAT,n,REST 11114) 则可抽取其共同属性为 (CCAT ,n,REST.<(1111)).

对于远距离节点之间的语义关联现象,可利用人工加标方式来扩充本次学习的条件集(CS)的元素空间,采用同样的算法来完成学习.

3.2.2.2 多步构造(Multi-Step Construction)

多步构造类似一种图遍历,从初始状态开始一步步构造到终止状态,其模型如下图

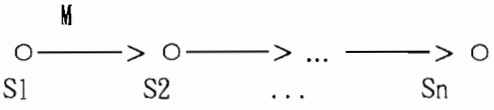

图2. 多步构造模型

图中的箭头表示动作函数(M),圆圈表示结点状态,系统先由输入的初始态S1和结束态Sn,比较被处理的语言单位中各语义节点之间的变化,一一对应地生成多个中间状态,然后在相邻状态之间调用一步构造策略,生成一系列原始的规则,将它们反馈到图2中的各个状态点之间的构造中.如果存在矛盾则重组各个状态,产生一新的状态图.再重复上次的动作,直至达到较满意的结果,不再存在矛盾,则本次学习即告成功. 这实际上是一个带回溯的多遍学习的过程.它的复杂度与被处理的语言单位的复杂度成正比.

4 实验结果与结论

我们使用前面的规则库检查重组部件对我们现有的汉英双向机器翻译系统(CETRAN)中的三千多条规则进行了处理之后,检查出了多处矛盾并调整了规则之间的优先级,使系统运行的准确率提高了很多.我们用100多个加标注的实例来验证学习部件的可行性.结果学习生成了十多条词性兼类规则和词法处理规则.这些规则经验证80%正确.这证明我们的策略是可行的.

总的来说,我们采用信息分层的思想,建成了一个智能规则系统模型.该模型不仅能方便的表达语言现象,有效的解决了规则膨胀问题,最主要的是它具有一些智能机制,克服了规则库的自我完善和规则库的维护困难的一些问题.它实际上是一个有效的知识库系统.

我们的工作还在继续,该项目受到国家自然科学基金的支持.

参考文献

〔1〕 Wang Baoku, Zhang Zhongyi, Yao Tianshun, Rule description language CTRDL in machine translation system, In: Proceeding of 1991 International Conference on Computer Processing of Chinese and Oriental Languages, Taipei,1991, 264-269.

〔2〕 唐泓英、姚天顺, 基于搭配词典的词汇语义驱动算法, <<软件学报>>, VOL.6 1995,78-85

〔3〕 Oliviero Stork, Natural Language in Multimodal Human-Computer Interface.IEEE Expert, April 1994

〔4〕 徐立本主编, <<机器学习引论>>. 吉林大学出版社, 1992