基于译后启发式信息的知识辅助获取系统: HIKAS

汪冰　黄河燕 陈肇雄

（中科院计算所机译中心　北京 100080）

摘要：IMT E/C 是以人工智能为基础的智能型机译系统。这种智能性的体现之一是系统的学习能力，即系统能不断地获取知识来完善自己。众所周知，知识获取是人工智能中的瓶颈问题，在本文中，我们提出了由译后反馈信息和启发式信息辅助获取知识的系统 HIKAS 。该系统根据用户对译文的修改，对源文语法分析树进行反向推导，帮助用户定位错误；根据译后启发式信息，辅助用户改进知识库。

关键字：人工智能 知识获取　机器翻译　启发式信息

Post-edit Heuristic Information based Knowledge Acquisition System: HIKAS

Wang Bing　　Huang Heyan　　Chen Zhaoxiong

(Intelligent Machine Translation Research Center, Institue of Computing Technology, The Chinese Academy of Science , Bei Jing 100080)

Abstract: IMT E/C is an intelligent machine translation system based on artifial intelligence. The intelligence is showed in part by the learning ability of the system. That is ,the system can acquire knowledge continuously to improve itself. It is well known that Knowledge Acquisition is a bottleneck problem in AI. In this paper, we present a Post-edit Heuristic Information based Knowledge Acquisition System: HIKAS. According to the user's revision of the translation, it can help to locate the error through a backword inference of the source sentence analysis tree ;it can also assist to revise the Knowledge Base according to post-edit heuristic information.

Keyword: Artifial Intelligence, Knowledge Acquisition , Machine Translation, Heuristic Information

一、引言

机译知识辅助获取是目前日益受重视的一个领域[6,7]。由机器辅助获取词典信息的方法很多，已进入实用阶段[8,9,10]。而辅助获取规则知识还处于探索性阶段，有归纳型、基于解释（例子）等方法[11,12]。通过译后知识反向推导，既可发现词条错误，也可发现规则错误，能够有效地实现知识辅助获取。我们设计、实现的 HIKAS 系统基于智能机译系统 IMT E/C，利用译后修改信息和启发式信息来辅助获取知识。下面首先介绍 HIKAS 的设计原理、实现算法，然后给出了实验结果和评价。

二、设计原理

IMT E/C 的知识库主要包括字典库和规则库。字典和规则均用 SC 规则表达[3][5]。字典形式和规则形式见[1]，翻译处理过程见[4]。翻译的错误主要是由词典和规则的错误造成的，参见[2]。

设计 HIKAS 时，主要基于以下考虑:

(1) 提高系统的智能化水平。计算机系统的智能化程度主要体现在系统的学习和自适应能力。通过译后反馈信息辅助获取知识，可以使系统具有一定程度的学习和自适应能力。

(2) 满足大规模工程化开发的需要。机译系统的开发是一个长期、艰巨的过程，尤其是其中的语言工程部分，要基于大规模的真实语料，且要多个人合作。提供了统一的调试工具后，有利于保持知识库的一致性。

(3) 提供更友好的集成调试环境。由于知识库的不断变化、更新，提供一个友好的调试环境，提高调试效率就及其重要了。 HIKAS 采用了下拉菜单形式，包含显示和输出两个窗口，具有以下特点:

(a)与 IMT/EC 的翻译过程相独立，不影响翻译速度。对翻译有误的句子，将其分析过程记录到一个文本文件上，作为 HIKAS 的输人。

(b)用图示的办法给出源文分析、译文生成过程，更为直观。

(c)与译后编辑相结合，提供了面向意段的修改命令。

(d)根据修改命令输出修改后的译文、帮助定位错误信息，并尽可能地提供修改建议。

(e)根据修改建议修改知识库。 HIKAS 中包含词典库、规则库的调试功能，使知识获取和知识库的完善一体化。

三、 辅助获取机制

机译知识辅助获取模块以源文分析树、译文生成树、词条规则、译后修改信息和启发式信息作为输人数据，对系统的字典库、规则库进行交互式修改，经过验证后，加人知识库中。下面具体说明这些输人数据。

(1) 词条规则，记录了源句中的词串、其语法、语义属性及对应的译文。

(2) 源文分析树集和译文生成树

源文分析树集记录了源句翻译的整个分析过程，其中包括所用到的所有源文分析树（完整或不完整），最后一个分析树(即源文终止分析树，下文简称源文分析树)满足了进一步的语法、语义检查而生成了最终译文。译文生成树记录了产生译文的过程。源文分析树和译文生成树是通过分析规则序列产生的。分析规则序列记录了生成译文时的所用到的所有规则，其数据结构略。

在[1]中，定义了意段结构、基本意段结构等。源文分析树和译文生成树中的每一个结点均是一个意段结构，叶结点为基本意段，非叶结点为复合意段。分析规则序列中的结点号对应意段下标，结点串对应意段语法类。意段结构的引人为利用反馈信息获取知识提供了方便。下文中，将意段结构简称为意段。

(3)源、译文意段之间的关系

源文意段与译文生成意段之间的关系可用二元数组表示，

$$(ch\_ind_i, cor\_en\_ind_j) \quad i=1,2,\ldots,n; \quad j=1,2\ldots,m \tag{3}$$

其中 ch_ind_i $(i=1,2,\ldots,n)$是译文意段下标，$cor\_en\_ind_j$ $(j=1,2\ldots,m)$是 ch_ind_i 对应的源文终止意段下标。若某一译文意段没有对应的源文意段，即它是由规则转换生成的，则其对应的源文意段下标记作 -k(k=0,1,2 ...)， k 是当前的规约步数。

(4) 修改信息。修改是对译文生成树的结点（意段）进行的，修改信息包括修改操作、涉及的结点以及修改内容，数据结构略。

(5) 启发式规则序列。根据译后编辑信息对原有规则的修改记录到启发式规则序列中，第 i 步规约用到的规则记作规则 i，对这条规则的修改记录到第 i 条启发式规则中。数据结构略。

四、基于启发式信息的知识辅助获取

在翻译不出的情况下，主要靠人工获取知识，方法见[13]。下面着重介绍在可译出的情况下，基于启发式信息的知识辅助获取。

对译文做修改时，给出译文生成树，译后修改以意段为基本单位。有几种典型的译后编辑操作：删除、插入、替换、对换、调整（移动）。根据修改命令修改译文，并将所做的修改存入启发式规则序列中。为给下面的分析提供方便，要求修改时做到：(1) 词条规则不正确时，首先修改词条规则。 (2) 修改时自右向左进行，且使修改尽可能少。

1、 源、译文信息的获取

译后编辑是对译文进行的，要反推得出错误的词条或规则，必须将对译文的修改反映到分析规则序列中去。产生分析规则序列的方法如下：

(1) 根据分析过程，得到分析规则序列中各步规则的头部结点号、头部结点串、规约结点号、规约结点串、转换结点串、与转换结点对应的头部结点号。因为译文生成树中的叶结点数还未确定，所以译文生成结点号暂时定为： TEMP + k (k 为规约步数， TEMP 为一足够大的正整数，如 1000)。

(2) 分析结束后，递归地确定分析规则中转换体中的各个结点号。这一过程包括下面两步：(a)对最后一条规则的转换体 S （表示句子结点）寻找叶结点，设译文叶结点数为 trans_leaf_num，初始时 trans_leaf_num= 0。设某个转换体中的结点号为 trans_no，其对应的头部结点号为 trans_cor_lhs_no 。如果 trans_cor_lhs_no 是叶结点，则 trns_no= trans_leaf_num ， trans_leaf_num++，返回；否则，在分析规则序列中找规约生成 trans_cor_lhs_no 的规则，对这条规则的各个结点寻找叶结点。

(b)确定译文叶结点数后，将分析规则序列中大于 TEMP 的结点号减去 TEMP ，加上 trans_leaf_num，这样便得到了有正确结点号的分析规则序列。

2、 定位错误规则，产生启发式信息

分析所有的修改操作，某些修改可直接产生启发式信息，将其记录到启发式规则序列

中去。下文中，用 ch_ind 表示译文意段下标，用 cor_en_ind 表示它对应的源文意段下标。

(1) 删除操作

命令形式为： d ch_ind ，即删除 ch_ind 对应的译文字串。

(a)若 ch_ind 是基本意段且 cor_en_ind 为负或零，即此意段是通过规则生成的，取 cor_en_ind 的负值，得到规则号，将这条分析规则中的头部、规约部分、转换体记录到启发式规则序列中，其中转换体部分删除了该意段。

(b)若 ch_ind 是基本意段且 cor_en_ind 为正，则检查词条规则，如果词条规则没有错误，则从分析规则序列中找出头部含有 cor_en_ind 的规则，将修改记录到启发式规则序列中，与(a)的情况相同。

(c)若 ch_ind 是复合意段，从分析规则序列中找出头部含有 cor_en_ind 的规则，将修改记录到启发式规则序列中，与(a)的情况相同。

(2) 替换操作

命令形式为： c ch_ind content，即将 ch_ind 对应的译文字串改为 content 。

替换操作是对基本意段进行的。若 cor_en_ind 为负或零，即此意段是通过规则生成的，取 cor_en_ind 的负值，得到规则号，将这条分析规则中的头部、规约、转换体分记录到启发式规则序列中，其中转换体中该意段变成了 content；否则，检查词典库。

(3) 对换操作

命令形式为： v ch_ind1 ch_ind2，即对换 ch_ind1 和 ch_ind2 对应的译文字串。

ch_ind1 、 ch_ind2 对应的源文意段分别为 cor_en_ind1 、 cor_en_ind2 ，在分析规则序列中找到头部既含有 cor_en_ind1 又含有 cor_en_ind2 的规则，将这条分析规则中的头部、规约、转换体记录到启发式序列中，其中的转换体对换了 ch_ind1 、 ch_ind2 对应的部分。

(4) 调整操作

命令形式为： m ch_ind1 h ch_ind2，即将 ch_ind1 对应的译文移到 ch_ind2 对应的译文之后。从分析规则序列中找出左部含有 cor_en_ind1 的规则作为错误规则。

(5) 插入操作

命令形式为： i ch_ind content，即在 ch_ind 对应的译文之前插入 content 。

(a)若插入的为标点，如果 ch_ind 与 ch_ind-1 在同一规则的左部，将此规则记录到启发式规则序列中，其中的转换体 ch_ind 对应的部分之前加入了标点；否则建立 ch_ind 的父意段下标表：(f1,f2, …, fn)，其中 f1 是 ch_ind 的父意段下标， fi 是 $f_{i-1}(i>1)$的父意段下标。同理，建立 ch_ind-1 的父意段下标表：(d1,d2, …, dm)。找到最小的 i 、 j ，使 fi 、 dj 是平行意段，确定既含 fi 又含 dj 的分析规则，将此分析规则记录到启发式规则序列中，其中的转换体在 ch_ind 对应的部分之前加入了标点；

(b)若插入的为汉字串，如果 ch_ind 与 ch_ind-1 在同一规则的左部，将此分析规则记录到启发式规则序列中，其中的转换体在 ch_ind 对应的部分之前加入 content；否则，找出左部含有 cor_en_ind 的规则，将此规则记录到启发式规则序列中，其中的转换体在 ch_ind 对应的部分之前加入 content 。

3 、 获取启发式规则

若词典内容需要修改，则直接修改词典内容，否则，通过启发式规则修改规则库。因为

翻译过程中，后面的分析受前面分析产生的状态信息的影响，所以启发式规则序列中的规则并不一定都可以作为启发式规则，同时启发式规则需要由用户确认是否正确，若不正确，提示用户输入启发式规则，形式为:

<规则号> <结点号 1> <语法类 1> ... <结点号 n> <语法类 n> --> <规约语法类> * <转换体>

整个获取启发式规则的步骤如下:

(1) 从启发式规则序列中取出第一条规则，显示这条规则，由用户确认是否为启发式规则，若不是，提示用户输入。建立父意段集合{f1,f2, ... fn}，其中 fi(i=1,2, ...,n)是此启发式规则头部各意段的父意段。转(2)。

(2) 若启发式规则序列不为空，取下一条规则，判断其头部的各意段是否在父意段集合中，若在，忽略此规则；否则显示这条规则，由用户确认是否为启发式规则，若是，将此规则头部各意段的父意段添加到父意段集合中。转(3)。

(3) 重复此过程，直至启发式规则序列为空为止。转(4)。

(4) 提问用户是否还需要输入其他启发式规则，当不需要时，结束。

4 、 完善规则库

根据启发式规则序列对规则库的完善包括三种情况: (1)创建规则，即构造一条新规则; (2)规则一般化，即去掉条件或减少限制，以扩大规则的使用范围; (3)规则特殊化，即增加新条件或对原有条件做更多限制。设启发式规则为 Heur_rule ，分析中所用的规则为 Orig_rule 。比较这两条规则，可提示用户修改或产生新的规则，步骤如下:

(1) 若二者仅<转换体>部分不同，检查规则库中是否有 Heur_rule 类规则。

(a)若无，则应补充此类规则: 设在当前的规约状态下，与 Heur_rule 头部对应的各个结点为: $H_1(S_1)$,..., $H_2(S_2)$ $H_n(S_n)$，其中 S_i(i=1,...,n)表示当前状态下的语法、语义属性，取出 S_i 中的某些属性作为 Heur_rule 相应头部的属性，取出 Orig_rule 的<上下文条件>中满足当前条件的部分作为 Heur_rule 的 <上下文条件>，从而得到一条新的规则，用户根据这条规则的意义对其进行修改，之后加入规则库;

(b)若有 Heur_rule 类规则，如果 Heur_rule 在 Orig_rule 之前，说明 Heur_rule 的优先级高，即分析过程中曾尝试过这条规则，但其不满足进一步的语法、语义检查，因此提示对 Heur_rule 类规则减弱限制; 否则，说明 Heur_rule 的优先级低，即分析过程中首先满足了 Orig_rule 的要求而没用到 Heur_rule ，因此应加强对 Orig_rule 类规则的限制。

(2) 若二者<规约>部分不同，处理办法与(1)相似。

(3) 若二者<头部>不同，如果规则库中不存在 Heur_rule 类规则，由用户根据当前结点状态对 Heur_rule 的<头部属性>、<规约属性>和<上下文条件>加以限制，得到这一类的规则; 否则，根据翻译处理过程“长规则优先”和“自右向左”的原则，比较二者的优先级，若 Heur_rule 的优先级高，则提示对 Heur_rule 类规则减弱限制; 若 Heur_rule 的优先级低，提示加强对 Orig_rule 类规则的限制。

5 、验证修改，完善知识库

对当前知识库的修改要经验证，只有在没有不良影响时，才可确认修改。 IMT E/C 系

统中有一个基本的语料库，其中收录了比较典型的句子（约 5 万句）及其译文，这些句子在当前知识库下是可正确翻译的。以这个基本语料库中的句子为标准，检验所做的完善对知识库的完整性和一致性是否有损害。若对词条规则做了修改，则从基本语料库中抽出所有含此词条的句子；若对规则做了修改，则抽出所有符合此规则的句子，对比这些句子修改前后的翻译结果，确认未受影响时，才可将修改加入知识库。

五、 实验结果及评价

我们从实际的调试语料中选取了一些句子，对 HIKAS 系统做了检验。词条规则的错误往往一看便知，例如:

1The 2car 3drinks 4gas 5. (1 这 2 辆 3 小汽车 4 喝 5 气体 6 。)

修改操作: c 4 耗

c 5 汽油

在 drink 中加入“耗”这个义项，在 gas 中加入上下文条件，以便区分“气体”与“汽油”。规则出错的情况比较复杂，具体例子见[13]。HIKAS 在实际调试过程中的使用证明它基本达到了设计要求，使 IMT/EC 具有更高的智能，并且提高了调试效率。

参考文献

[1]. 黄河燕，陈肇雄. 一种智能译后编辑器的设计及其实现算法. 软件学报 1995,Vol6,No3

[2]. 黄河燕，陈肇雄. 基于译后反馈信息的知识辅助获取. 人工智能新进展 1994

[3]. 陈肇雄，高庆狮. 智能化英汉机译系统 IMT/EC .中国科学（A 辑） 1989(2)P186-192

[4]. 陈肇雄,陈强. 智能机器翻译进展 P419 1992

[5]. 陈肇雄. SC 文法功能体系.计算机学报 1992,15(11),P801~805

[6]. Binot, J.L., and L.Jensen, A Semantic Expert Using an Online Standard Dictionary, In Proceedings of IJCAI-87,P709~714

[7]. Nirenburg S., J.Carbonell, M.Tomita, K.Goodman, Machine Translation:A Knowledge Base Approach, Morgan Kaufman Publishers, 1992

[8]. Bonnie J. Dorr, Joseph Garman, Amy Weinberg, From Syntactic Encoding to Thematic Roles : Building Lexical Entries for Interlingual MT, MT Vol 9 No.3-4 94/95 P221~250

[9]. Bran Boguraev, Ted Briscoe, Large Lexicons for Natural Language Processing: Utilising the Grammar Coding System of LDOCE, 1987. CL,Vol 13, No3-4

[10]. Deryle Lonsdale, Teruko Mitamurn, Eric Nyberg, Acquisition of Large Lexicons for Practical Knowledge-based MT, MT Vol 9, No.3-4 94/95 P251~283

[11]. Hussein Almuallim, Yasuhiroo Akiba, Takefumi Yamazaki, Akio Yokoo, Shigeo Kaneda, A Tool for the Acquisition of Japanese-English Machine Translation Rules Using Inductive Learning Techniques, IEEE 10th Conference on Artificial Intelligence for Application, 1994, Los Alamitos, California

[12]. Robert E., Simmons, Yeong-Ho Yu. The Acquisition and Use of Context-Dependent Grammar for English, CL Vol 18 ,No. 4,1992

[13]. 汪冰. 智能机译系统的知识辅助获取.计算所硕士论文.1997,6