第6卷第3期
1995年3月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 6, No. 3
Mar. 1995

一种智能译后编辑器的设计及其实现算法*

黄河燕　　陈肇雄

（中国科学院计算技术研究所智能机器翻译研究开发中心，北京 100080）

摘要　译后编辑是改进机器翻译译文质量的主要手段.本文提出一个智能译后编辑器的设计原理和实现算法.该编辑器以意段为基本处理单位,既可以形成适于反向推理的译后编辑反馈信息,为机译系统知识的自完善提供处理依据,又可以实现源译文句子级和意段级的多窗口同步显示.同时还利用智能机译系统对句子/短语的多解译文和单词的多义查询能力,使用户只要在误译文的多个候选译文中选择正确译文,从而大量减少人工删除误译文和插入正确译文的操作,并通过设置多个意段译文位置的自动调整机制,提高译后编辑的效率.

关键词　机器翻译,人工智能,译后编辑.

由于机译译文质量不可能达到人工翻译的水平,因而常常需要对机译译文进行译后编辑,以便得到较准确的译文.可以说,机译系统的效率由翻译效率(翻译处理时间和译文正确率)和译后编辑效率决定,译后编辑处理智能的关系到整个机译系统的实用水平.

通常的纯正文编辑软件是对字串进行操作的,没有考虑到译后编辑主要对意段操作的特点,未提供适合于译后编辑的操作,而且由于这些编辑软件不可能有与机译系统中翻译转换模块和知识处理模块的接口,因此不能充分利用翻译转换模块的功能给译后编辑提供可供选择的信息,也不可能形成合适的对知识处理模块的反馈信息,以改善机译系统的知识.

在本文中,我们提出一个智能译后编辑器,该编辑器以意段作为编辑修改的基本单位,既可以给知识处理模块形成适合于进行反向推理的译后编辑修改反馈信息,给机译知识的自动获取提供处理依据,从而提高机译系统的学习和自适应能力,不断改善机译系统的知识;又可以实现源译文句子级和意段级的多窗口同步显示,给译后编辑带来很大的方便性.同时还利用智能机译系统对句子翻译的多种译文和对单词的多义查询能力,使用户只要在误译文的多个候选译文中选择正确译文,从而减少人工删除误译文和插入正确译文的操作,并通过设置多个译文意段位置的自动调整机制,提高译后编辑的效率.下面,我们给出这种智能译后编辑器的设计原理和实现方法.

1　设计原理

译后编辑的任务是对机译错误译文进行编辑修改.机译译文中最常出现的两种错误译

* 本文 1994－01－13 收到,1994－04－09 定稿

本研究得到国家自然科学基金的资助.作者黄河燕,女,1963年生,副研究员,主要研究领域为机器翻译,面向对象程序设计,大型智能应用系统.陈肇雄,1961年生,研究员,主要研究领域为机器翻译,人工智能.

本文通讯联系人:黄河燕,北京 100080,中国科学院计算技术研究所智能机器翻译研究开发中心

文是:①译文中某些结构成分的顺序不合适;②译文中某些结构成分的译文不正确或不确切.这两种情况都是要对以结构成分为单位的译文片段进行编辑修改操作.这些结构成分一般都对应于译文结构分析树中节点所对应的译文片段,我们把这种译文片段称为一个意段.因此,我们在设计译后编辑器时,充分考虑到译后编辑的这一特点,以意段作为编辑处理的基本单位,并基于以下几点基本考虑:

(1)提高机译系统的智能化水平

可以说,计算机应用系统智能化的主要特征是系统是否具有学习和自适应能力,即系统是否能够根据自身的错误或外界对系统错误纠正的反馈信息获取正确的知识,改善系统的性能.而对机译译文的译后编辑就是人工对机译系统译文错误的一种纠正.因此,为了提高整个机译系统的智能化程度,我们把译后编辑器对译文的修改,形成适合于进行反向推理的译后编辑反馈信息,提供给机译的知识处理模块以给机译知识的自动完善和获取提供处理依据.

(2)提供面向译后编辑的操作

机器翻译译后编辑的两种典型操作是:(1)译文结构成分的顺序调整;(2)译文某些成分的修改.对于第一种情况,如果按通常编辑器的操作方式,逐个地调整它们的顺序,则操作比较繁琐.为此,我们设置了译文多意段位置的自动调整机制,可以使这种译文顺序调整操作相对简单方便;对于第二种情况,一般编辑器的操作是删除该成分的误译文,同时插入其正确译文或进行串替换.这一方面需要删除和/或插入操作;另一方面,如果用户对一个成分的译文不能肯定时,就需要查阅字典找出正确译文.为此,我们在译后编辑器中提供了对智能机译系统中翻译转换模块的接口,利用智能机译系统对短语/句子的多解翻译和对单词的多义查询能力,通过对所选择的误译译文意段相对应的源文意段调用智能翻译转换模块,并在辅助窗口上逐个地显示对其翻译出来的多个解,选择一个合适的译文替换所选择的误译文,从而可以减省人工查阅字典的时间和对误译文的删除及对正确译文的插入操作,减少人的工作量.

(3)多窗口源译文同步显示

机译译后编辑在对译文进行编辑修改时,常要对照源文的内容.如果在对译文进行修改时,能够把相应的源文内容同时显示出来,并且使源文显示的内容随着译文修改位置的变化而改变,即二者之间的显示及光标位置保持一种对应关系,则会大大方便于译文的修改编辑过程.因此,我们采用多窗口源译文同步显示方式,一个用于显示与所编辑译文相对应的源文内容;一个用于显示所要编辑的译文内容;还有用于执行其它命令和用于显示编辑过程中所需要的一些辅助信息,如某个源文片段的多个可供选择的译文等的临时辅助窗口.并且在源译文的窗口显示中,使当前正在编辑的源译文句子显示和句子中相应意段显示同步对应.

2 实现方法

根据上述设计原理所设计的智能译后编辑器的总体功能结构如图1所示.其中,编辑主模块主要对用户的编辑命令进行解释执行;文件I/O模块、键盘输入模块的功能与通用编辑器的功能基本相似;正文显示模块和正文编辑模块除了实现一般通用编辑器的功能外,还针对译后编辑的特点,设置了一些适合译后编辑的操作功能;物理屏幕模块提供一些最基本

的屏幕操作功能,让正文显示模块调用;反馈信息获取模块根据译后编辑对译文的修改,形成一定形式的反馈信息传递给知识处理模块;知识处理模块和翻译转换模块分别是机译系统中实现知识管理[1]和翻译处理[2,3]的主要机制.下面,我们给出智能译后编辑器的数据表示形式和实现算法.

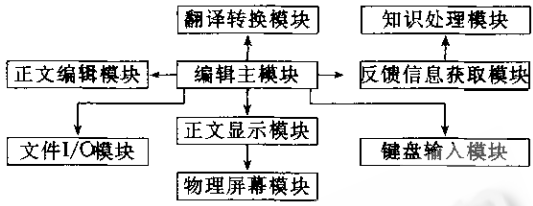

图1 译后编辑器的功能模块结构

2.1 源译文意段结构表示

对机译错误译文进行人工编辑修改是用户对机译系统的一种极其重要的纠错信息.如果机译系统能够充分利用这些信息进行反向推导,找出在译文分析过程中所使用的导致错误译文的字典和/或规则库中的错误规则或所缺少的规则,进而根据这些信息自动地对错误规则进行修改或是归纳出正确的规则,即对已有机译知识进行自我完善,将会大大提高机译系统的智能化程度.

利用译后编辑反馈信息进行知识完善和获取的最有效方式是把对译文的修改反推到对SC源文结构分析树[4,5]中相应节点的转换生成规则[6]的修改,并根据这种反推改善或创立机译系统的规则.因此,为了能在译后编辑阶段形成对知识获取有用的反馈信息,我们的翻译处理机制给译后编辑器两种输入数据:一种是源译文正文文件,另一种是把机译的源译文按系统对句子进行翻译时形成的结构分析树给出源译文对应于结构分析树的各节点的意段表示及意段间的对应关系.译后编辑器在对译文进行修改编辑时,同时也对句子的源译文的意段结构及对应关系作相应的修改,并把这些修改了的意段结构及其对应关系反馈给知识处理模块.因为译文意段结构及其对应关系的改变能够清楚地反映出结构分析树中各结构成分的转换生成模式的改变,从而给反向推理提供充分的处理依据.

源译文的意段结构用形为如下的二元组的数组表示,:

$$(ind, str) \tag{1}$$

或

$$(ind, (ind_1, ind_2, \cdots, ind_n)) \tag{2}$$

其中 ind 表示意段的下标,(1)表示一个基本意段,str 表示这个基本意段所对应的字串;(2)表示一个复合意段,$(ind_1, ind_2, \cdots, ind_n)$是组成这个复合意段的所有子意段的下标组成的有序表.对于源文复合意段,它们是结构分析树中归约形成相应于 ind 结点的各结构成分所对应结点的意段下标;对于译文复合意段,它们分别对应于结构分析树中相应于 ind 结点的 SC 规则转换体 $T_1, T_2, \cdots, T_n$ 中的一个转换生成成分.复合意段的字串是由下标为$(ind_1, ind_2, \cdots, ind_n)$的意段的字串并联形成.意段间的对应关系用形为如下的二元组数组表示,

$$(a_i, b_j) \qquad i=1,2,\cdots,m,\ j=1,2,\cdots,n$$

其中 a_i 是源文意段的下标;b_j 是译文意段的下标.该二元组表示源文中下标为 a_i 的意段与译文中下标为 b_j 的意段是相互对应的.如果源译文的某个意段没有对应的译、源文意段,则其对应的译、源文意段下标设为-1.例如,英汉翻译中英文句子"I read a book."的结构分析树如图2,则源译文的意段表示分别为:

((1,I),(2,read),(3,a),(4,book),(5,(3,4)),(6,(2,5)),(7,(1,6)))

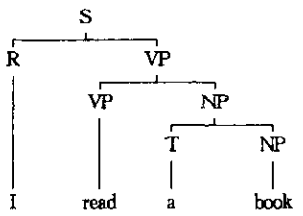

图2

((1,我),(2,读),(3,一),(4,本),(5,书),(6,(3,4,5)),(7,(2,6)),(8,(1,7)))

意段对应关系为:

((1,1),(2,2),(3,3),(−1,4),(4,5),(5,6),(6,7),(7,8))

为了下面描述的方便,我们给出如下定义:

定义 1. 一个意段$(ind,cont)$是另一个复合意段$(ind',(ind_1,ind_2,\cdots,ind_n))$的一个子意段,如果存在$j(j=1,2,\cdots,n)$使得$ind_j=ind$成立,并称后者为前者的父意段.

推论 1. 一个意段只能有一个父意段,即一个句子的意段结构表中最多只有一个意段的意段下标表含有这个意段的下标.

定义 2. 一个意段$(ind,cont)$是另一个意段$(ind',cont')$的一个平行意段,如果ind和ind'属于同一个复合意段的意段下标表中,否则称它们为不平行意段.

定义 3. 一个意段$(ind,cont)$也称为ind意段.

这里$cont$和$cont'$可以是字串形式或者是下标表形式.

2.2 智能译后编辑操作

为了适合译后编辑的特点,我们的编辑器以意段作为编辑的基本处理单位.同时为了形成给知识处理模块的译文意段结构及源译文意段对应关系改变的反馈信息及实现源译文意段的同步显示,这里的译后编辑操作不仅要对译文的正文文件内容以意段为单位进行相应的修改,还要对译文意段结构和意段对应关系数组作相应的修改以反映修改后的译文的结构.另外,考虑到译后编辑的特点,有些通用的编辑操作功能是不能适用的,如删除光标所在行;删除当前行头至光标位置的字串;删除光标位置到当前行尾的字串;当前定义块写入文件等.而且,除了通常的正文编辑功能外,我们还提供了几种典型的译后编辑操作:

①意段删除:删除译文某一意段中的部分字串或全部字串;

②意段插入:插入新的目标语言字串到译文某一意段中;

③意段替换:用新的目标语言字串替换原有译文某一意段中的部分或全部字串;

④意段对换:对换译文中两个意段字串的位置;

⑤意段调整:调整译文中某些意段字串的位置.

它们相应的实现方法如下:

(1)意段删除操作

对译文的意段删除操作,一方面要把译文正文文件中某个意段字串删除,同时还要对译文意段结构及意段对应数组作修改,分为两种情况:

①所删字串str_1是一个基本意段:(ind,str)字串str中的一部分,则只要把该意段的字串str中的相应部分字串str_1删除即可.

②所删字串str_1是一个意段:$(ind,cont)$的全部字串,这又分两种情况处理:

·$cont$为字串形式str,且$str_1=str$,则只要把ind从ind的父意段的意段下标表中删除,并从译文意段结构数组中把这个意段删除,同时把意段对应数组中形为:(ind_1,ind)的对偶改为:$(ind_1,-1)$即可;

• $cont$ 为下标表形式$(ind_1, ind_2, \cdots, ind_n)$，并且：

$$str_1 = pstr(ind_1) \| pstr(ind_1) \| \cdots \| pstr(ind_n)$$

$pstr(i)$表示意段下标为 i 的意段字串. 这种情况要把 ind 从其父意段的下标表中删除，并从意段结构表中把下标为 $ind_1, ind_2, \cdots, ind_n$ 的意段及它们的子孙意段都从译文意段数组中删除，同时把意段对应数组中译文下标为上述所删除意段下标的对偶中相应译文下标改为－1 即可.

(2)意段插入操作

在译文正文文件的某一意段插入一个新的目标语言字串，同时对译文意段结构及意段对应数组作相应修改，也分为两种情况：

①在译文某一基本意段的字串中间插入新的字串，则只要在该意段的字串中加入相应新字串即可.

②在译文意段的字串的边界插入新的字串，这又有两种情况：

• 在两个平行意段$(ind_1, cont_1)$和$(ind_2, cont_2)$之间插入一些新的字串 str，则在译文意段结构中这两个意段之间加入一个新的基本意段：

$$(ind, str), ind = \max(b_j) + 1$$

并在 ind_1 和 ind_2 的父意段的下标表中，在 ind_1 和 ind_2 之间加入下标 ind. 同时还在意段对应数组中加入一个二元组：$(-1, ind)$；

• 在两个不平行意段$(ind_1, cont_1)$和$(ind_2, cont_2)$之间插入一些新的字串 str，在译文意段结构中插入成分可以是在第一个意段的后面或第二个意段的前面，这时由光标的位置来确定把插入的字串加到哪个意段中，如果光标在第一个意段的字串上，则在译文意段结构中第一个意段的后面加入一个新的基本意段：

$$(ind, str), ind = \max(b_j) + 1$$

并在 ind_1 的父意段的下标表中 ind_1 的后面加入 ind，同时还在意段对应数组中加入二元组：$(-1, ind)$.

如果光标在第二个意段的字串上，则在译文意段结构中第二个意段的前面加入一个新的基本意段：

$$(ind, str), ind = \max(b_j) + 1$$

并在 ind_2 的父意段的下标表中 ind_2 的前面加入 ind，同时还在意段对应数组中加入二元组：$(-1, ind)$.

(3)意段替换操作

对某个译文意段的替换操作对应于译文意段结构的改变可以有两种情况：一种是用新的字串替换译文某一基本意段中原有字串的一部分或全部字串，此时，只要对基本意段字串作相应的替换操作即可；另一种是用新的字串替换译文某一复合意段的全部字串，这时要把复合意段改为基本意段，其字串即为新的字串，这一复合意段的下标表变为空，并把其下标表中的意段及其子孙意段都从译文意段数组中删除，把意段对应数组中的相应译文意段下标改为－1.

(4)意段对换操作

这种操作除了对正文文件作修改外，对意段结构及其对应关系数组的改变也只能有如

下两种情况：

①两个平行意段(ind_1,$cont_1$)和(ind_2,$cont_2$)之间字串的对换，则只要把这两个意段的父意段中意段下标表中的 ind_1 和 ind_2 的位置对换即可.

②两个不平行意段(ind_1,$cont_1$)和(ind_2,$cont_2$)的字串位置的对换，则用 ind_1 替换 ind_2 的父意段下标表中的 ind_2，用 ind_2 替换 ind_1 的父意段下标表中的 ind_1.

(5)意段调整操作

调整意段字串位置操作对意段结构及其对应数组的改变也有两种情况：

①把一个意段(ind,$cont$)的字串 str 调整到其父意段的某一位置，这时只需把其父意段的子意段下标表中 ind 的相应位置进行调整即可.

②把一个意段(ind,$cont$)的字串 str 调整到不是它的父意段的另一个复合意段(ind',(ind_1,ind_2,…,ind_n))中的某一位置，这时需要确定调整到复合意段 ind' 中的位置，然后在 ind' 的下标表中的适当位置加入 ind 即可.

在上述的意段编辑操作中及下面对意段的重翻译和意段同步显示中，如果用户所选择的字串不是一个意段的完整字串，则由编辑器自动将其扩展为一个包含所选择字串的最小完整意段的字串，使编辑器能够返回合理的译文意段结构表示.

2.3 意段重翻译/单词多义查询操作

由于智能机器翻译系统可以提供对句子/短语的多解翻译，在一般情况下，最合适的译文是系统给出的第一个解，但也有一些情况最合适的译文不是系统给出的第一个解，在进行译后编辑时，用户可以通过调用翻译转换模块对源文句子/短语进行重翻译，用户可以在给出的多个解中选出一个最为合适的译文，用以替换译文中的某一部分或插入到译文中的某一位置，从而避免对句子/短语不合适译文的编辑修改.

另外，在译后编辑要对某一译文基本意段进行修改时，既可以用上述的字串替换操作来完成，同时为了省免用户在不能清楚地给出正确译文时查询字典的工作，我们还提供了对源文意段单词的多义查询操作，并在辅助显示窗口上给出其多个可供选择的译文，然后选择一个合适的译文替换该单词所对应的原译文或插入到译文的某一位置，并对译文意段结构中相对应意段中的字串作相应的替换或插入一个新的译文基本意段.

2.4 多窗口源译文同步显示

多窗口源译文同步显示操作就是使机译源文和译文显示窗口的显示同步，这包括两个方面，一方面是指源译文对应句子的同步显示，即当编辑一个译文句子时，把其相应的源文显示窗口的源文句子进行反显，这相对比较容易实现；另一方面是源译文句子意段显示的同步，即当编辑修改译文句子的一个意段的译文时，对其相应的源文意段的源文内容进行再反显，从而方便了编辑过程.

源译文句子意段的同步显示，由于其对应关系的复杂性，实现起来比较困难. 具体的方法是，当对译文中的某个字串进行了选择，则根据这个字串查找到所选择译文对应的译文意段，查找算法为：

①如果所选择的译文属于一个基本意段，则所对应的意段即为这个基本意段；

②如果所选择的译文属于一个复合意段，且没有比该复合意段更小的复合意段包含所选择的字串，则所对应的意段即为这个复合意段. 然后根据译文的意段下标 ind，在意段对

应结构中查找译文下标为 *ind* 的二元组：(*X*,*ind*)，得到其相应的源文意段下标 *X*，并把源文中编号为 *X* 的意段的字串内容在屏幕上进行再次反显.

3 结　语

本文，我们提出了一个智能译后编辑器的设计原理及其实现算法. 这个译后编辑器以意段作为基本的编辑显示单位，可以形成给知识处理模块的编辑反馈信息，以进行反向推理，给机译知识的自动获取提供处理依据，提高机译系统的学习和自适应能力，不断改善机译系统的知识. 同时利用智能机译系统对句子/短语翻译的多种译文和对单词的多义查询能力，使用户只要在误译文的多个候选译文中选择正确译文，从而减少人工删除误译文和插入正确译文的操作. 此外，通过设置多个译文意段位置的自动调整机制和源译文多窗口同步显示技术，提高了译后编辑的效率.

参考文献

1 Chen Zhaoxiong, Gao Qingshi. IMT—KB: a knowledge base system for machine translation. Proc. of the Inter. Conf. for CPCOL, Toronto, 1988.

2 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)：186—194.

3 陈肇雄主编. 机器翻译研究进展. 北京：电子工业出版社，1991.

4 陈肇雄. SC 文法功能体系. 计算机学报，1992，**15**(11)：801—808.

5 黄河燕，陈肇雄. 智能机器翻译研究. 见：吴泉源，高文主编，中国计算机智能接口与智能应用前沿研究 1993，北京：国防工业出版社，1994.

6 黄河燕，陈肇雄. 机器翻译译文生成算法. ICCC'94, Singapore, 1994.

DESIGN AND IMPLEMENTATION OF AN INTELLIGENT POST—EDITOR

Huang Heyan　Chen Zhaoxiong

(*Intelligent Machine Translation Research Center, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing* 100080)

Abstract　In this paper, an intelligent post—editor is proposed. In the editor, feedback messages suitable to backward inference can be formed to the knowledge acquisition mechanism to improve the knowledge of MT system. In addition, the number of delete and insert operations can be greatly reduced by using the multiple solution ability of the system's translation mechanism, making the users select correct result from several candidate strings rather manually deleting error string and insert correct string. Furthermore, the efficiency of post—editing can be enhanced by providing automatic adjusting the order of parts in output and the synchronous display of source text and target text.

Key words　Machine translation, artificial intelligence, post—editor.