

文章编号:1007-130X(2005)11-0066-03

一种句型转换和近似机器翻译方法及算法*

A Syntactic Transfer and Approximate Machine Translation Method and Its Algorithms

杨宪泽, 雷开彬, 吴守宪, 张上游, 宁爱华

YANG Xian-ze, LEI Kai-bin, WU Shou-xian, ZHANG Shang-you, NING Ai-hua

(西南民族大学计算机科学与技术学院, 四川 成都 610041)

(School of Computer Science and Technology, Southwest University for Nationalities, Chengdu 610041, China)

摘 要:在机器翻译的研究中,混合式方法是一种好方法。本文讨论了句型转换的机器翻译和近似机器翻译,提出了规则索引算法和一种近似机器翻译的算法。

Abstract:In the research of machine translation, the hybrid method is efficient. The article discusses syntactic transfer and approximate machine translation, and puts forward the rule index algorithm and the approximate algorithm of machine translation

关键词:机器翻译;句型转换;规则索引算法;近似算法

Key words:machine translation; syntactic transfer; rule index algorithm; approximate algorithm

中图分类号:TP391.2

文献标识码:A

1 引言

基于转换的机器翻译方法在机器翻译界仍然占主导地位。机器翻译是 21 世纪要解决的科技难题之一,主要困难是自然语言在各个层次上的歧义性^[1,2]。为此,人们注意寻求简单的翻译方法。迄今为止,机器翻译主要包括有基于转换、基于中间语言、基于统计和基于实例四种方法^[3]。

基于转换的方法采取了一系列转换生成层次,各种分析多(如词法、句法、语义和语境等)。虽然这是最传统的方法,现在还不少使用,但它的复杂性制约了正确率,研究越深入,难度越上升,速度却下降。

中间语言的方法对于多语种的机器翻译作为中介可行,它为多语种对译创造了良好环境,大大减少了机器翻译的重复杂度。但是,中间语言的方法除了与基于转换的方法有相同缺点外,它还有人们迄今都怀疑的问题,即是否能够完整地构造各种不同的自然语言语法和语义?

基于统计的方法应用了语音识别的思路,以大规模的双语语料库为基础,对源语言和目标语言词汇的对应关系进行统计,根据统计规律输出译文。这种方法没有使用语言知识,却取得了一定的正确率,所以现在研究这种方法的

人较多。但是,它也有被怀疑之处——这种方法会不会由于本身的固有属性,不可能有很高的译文正确率。

基于实例的方法相对于其它翻译方法来说简单一些,它通过结构化的翻译例子直接把源语言的句子与目标语言的句子对应起来,即检索与输入句子最相似的例句,定义样本和新句间的相似度;然后计算每个相似例句的相似度,再选最优者。

近十多年来,对机器翻译的多方面研究使许多人认为,好的机器翻译系统应采用混合方法,因为就目前情况看,无论采用何种方法实现的机器翻译系统,都没有混合方法质量好。因此,在我们的藏汉英机器翻译资助项目中选择了混合方法,并作了改进。当然,这些改进的根本目的就是降低机器翻译的歧义处理难度。

改进的混合方法主要包括两大模块:(1)句型转换的翻译模块;(2)实例近似翻译模块。本文的工作就是讨论这种混合方法的模块实现以及相关的算法。

2 混合方法的模块实现

在实施机器翻译时,软件首先运行句型转换的翻译模块,若不能翻译,再运行近似翻译模块。混合方法的实施,较大地提高了机器翻译的成功率。

* 收稿日期:2004-02-08;修订日期:2004-04-07

基金项目:国家民委重点科研项目基金资助项目(234408)

作者简介:杨宪泽(1954-),男,四川成都人,教授,研究方向为数据结构和自然语言处理。

通讯地址:610041 四川省成都市西南民族大学 145 信箱;Tel:(028)85522317;E-mail:myjkxy@vip.sina.com
Address:Mail Box 145, Southwest University for Nationalities, Chengdu, Sichuan 610041, P. R. China

2.1 句型转换翻译模块实现简述

在基于转换的模块构造中,我们采用的方法是句型间的转换翻译。这样,将提高翻译的成功率^[4,5]。句型转换的翻译是把源语句的单词和句型结构映射到译文相对应的单词和句型结构。双语对译的核心是句型结构的表达式相同,所以我们建立了双语句型结构表达式集合模块。

句型转换的翻译主要有两种:

(1) 句型转换顺序相同的翻译。如汉译英中,“他们学习英语”和“我们研究汉语”句型完全相同,句型表达式只需建立一个:rvn;对应英文的语序也完全相同:r'v'n'。这样的句型翻译,只要自动分词正确,无语义问题,译文的质量就可以保证。并且,再多的相同句型也只需一个句型表达式,这无疑在完全实例翻译的基础上大大进了一步。

(2) 句型转换顺序不相同的翻译。如汉译英中,“我们是新中国的学生”,句型表达式为:rvan(1)pn(2);其译文是“We are students of New China”,调序生成译文的表达式是:r'v'n'(2)p'a'n(1)。

句型转换的翻译虽然在完全实例翻译的基础上大大进了一步,但自动分词、词义消歧、语义分析、译文转换等步骤不能省略。鉴于这种方法国际国内研究的时间最长,虽然仍面临许多困难,但也硕果累累。我们的研究更多的是借鉴,限于篇幅,不再赘述。

2.2 近似翻译模块实现简述

近似翻译是日本机器翻译专家 Nagao 在 80 年代提出的一种方法,用已经存在的翻译实例(双语文本)作为知识源,这种方法称为基于类比的翻译,后来普遍称为基于实例的翻译。现在,近似翻译思想已被广泛采用,它可以通过结构化的翻译例子直接把源语言的短语和句子与目标语言的短语和句子对应起来。近似翻译(基于实例的机器翻译)的实现过程简单概述如下:给定源语言输入句子 S,在双语语料库 C 中匹配查找一个最相近的句子 S',则 S'的译文 T 就被接受为 S 的译文。

近似翻译最重要的问题就是相似度的计算。相似度的计算有许多方法,基本上可分为相似程度计算和距离程度计算两类。计算的依据则各不相同,可以按照单词本身是否相同来计算,可以按照单词所具有的词类、语义类是否相同来计算,还可以按照形态变化、语义上下位关系等来计算。

相似度的计算按照距离程度计算,分值越小越好,此时的分值是罚分。系统选择分值最小的句型表达式做句型转换的翻译。

为了避免计算的复杂性,我们把句型表达式少一个基本元素或多一个基本元素赋予罚分,分值多少由这一元素的重要性决定:

(1) 少一个基本元素,若它是关键词可以高达 3 分;若它是助词、量词、语气词等,分值可以 0.1~0.3 分。当然,这可以事先由已经搜集的句型表达式确定。

(2) 多一个基本元素,其分值的确定只能按词性给出,相对来说还显得有些粗糙。

本文的第 4 节将给出一种包含有相似度计算的近似翻译算法。

3 规则索引算法

基于句型转换的翻译始终都要与规则集打交道。原文分析、原文译文转换、译文生成三个阶段的规则集中的规则数量庞大,规则的匹配时间势必成为影响系统运行速度的重要因素。针对这一点,我们近来依据文献[6,7]的基本思想,提出了产生式规则索引算法,经实验证明,效果较好。

3.1 规则索引原则

(1) 每条规则条件部分视为一个字符串 B_i , 含有单个字符 $b_{i1}, b_{i2}, \dots, b_{ik}$ (K 为字符串长度), 可分划:

$LEN(B_i \$)$; 求字符串长度

do j from 1 to k

$b_i \$j \leftarrow MID\$(B_i \$, j, 1)$; 切分字符串成单一字符。

(2) 在计算机内,每个字符可转换成 ASCII 码介于 0~255 之间的十进制数。

$C_{i1} = ASC(b_{i1}), C_{i2} = ASC(b_{i2}), \dots, C_{ik} = ASC(b_{ik})$

(3) 每条规则条件部分的十进制值为: $C_i = C_{i1} + C_{i2} + \dots + C_{ik}$ ($i=1, 2, \dots, N$)。

到此为止,规则 R_i 对应 C_i ($i=1, 2, \dots, N$)。 C_1, C_2, \dots, C_N 中有的数可能相同,但由于规则条件部分含有的字符较多,引起 $C_i = C_j$ ($i \neq j$) 的条件为:

① 字符串中单个字符都相同(如 abcd、bacd 等);

② $C_{i1} + C_{i2} + \dots + C_{ik} = C_{j1} + C_{j2} + \dots + C_{jk}$ 。这种情况出现的频率不可能太高,而且我们允许有一定的相同数,只要相同数的个数 $\ll N$ 。

3.2 规则索引算法描述与分析

算法描述:

A1: 设有 N 条规则,规则 R_i ($i=1, 2, \dots, N$) 条件部分视为字符串 C_i , 可以分划成单个字符 $b_{i1}, b_{i2}, \dots, b_{ik}$ (K 为字符串长度)。

A2: (在计算机内,每个字符都可以转换成 ASCII 码介于 0~255 之间的十进制数)。求以下转换: $C_{i1} = ASC(b_{i1}), C_{i2} = ASC(b_{i2}), \dots, C_{ik} = ASC(b_{ik})$ 。

A3: $C_i = C_{i1} + C_{i2} + \dots + C_{ik}, i=1, 2, \dots, N$ 。

A4: 扫描一遍已转换成数值的集合 C , 求 C_{max} 、 C_{min} (C_{max} 和 C_{min} 为集合 C 的最大和最小值)。

A5: 变换 $M = INT((C_i - C_{min})/a)$, 其中 a 为大于等于 1 的正整数, $i=1, 2, \dots, N$ 。

A6: 开辟一个数组空间 F , 容量为 $INT((C_i - C_{min})/a)$, 建立规则 $R_i \rightarrow F(M_i)$ 索引, $i=1, 2, \dots, N$ 。若有规则 R_i 和 R_j ($i \neq j$) 使 $F(M_i) = F(M_j)$, 则采用链接方式将它们链接起来。

匹配操作时,有关事实(假定词组、词组特征等)仍视为字符串,利用一程序进行如下步骤:

(1) 将事实字符串分划成单个字符;

(2) 求 $C_{i1}, C_{i2}, \dots, C_{ik}$ 和 C_i ;

(3) 计算 $M_i = INT((C_i - C_{min})/a)$;

(4) 让 $M_i \rightarrow F(M_i)$ 的地址,这时若无规则,失败退出;若为唯一规则,成功;若为链,则按一般方法继续链上的匹配操作。

一般的规则匹配实际上是字符串的匹配操作,切分成单个字符与索引匹配,方法是共同的。不同的部分是:一般匹配将进行单个字符的一一比较和传送;索引匹配有求字符串长度、 C_i 和 M_i 的操作。据指令流方式推算,这两种匹配方式指令执行时间相差无几。但是,一般匹配成功最多次数为 N (N 为规则数),最小次数 1,平均次数 $(1+N)/2$

2;而索引匹配成功最大次数为 j (j 为 C_1, C_2, \dots, C_N 相同数的个数, $j \leq N$), 最小次数 1, 平均次数 $(1+j)/2$ 。

4 近似翻译算法

4.1 准备

近似翻译的实现首先需要建立大规模语料库,即用大量的翻译实例(双语文本)作为知识源。对选择的汉语实例的单词进行如下处理(假定做汉英翻译):

- (1) 切分出汉语实例中的单词;
- (2) 单词分值确定;
- (3) 单词总分值确定;
- (4) 单词个数确定;
- (5) 汉语句子个数统计。

4.2 算法概述

近似翻译的实质是近似检索(匹配),即检索与输入句子最相似的例句,定义样本和新句间的相似度,然后计算每个相似例句的相似度,再选最优者。

近似翻译算法构成思路:

- (1) 待译句自动分词;
- (2) 单词个数计算, $E=L, I=1, Y=1$ 。

(3) 待译句子与目标语句比较,比较原则是两句正确率应该在 75% 以上,算法如下:

① 目标句的总分值乘 75%, 确定这个句子允许“出错”的分值, 定为 FZ 。初值 $J=1, D=1$ 。

② $P=D$, 待译句的单词与目标句的单词比较, $D=D+1$, 直至 $D=K(I)$, 成功按序转③, 类推第二, ..., 第 $K(I)$ 单词 ($K(I)$ 为单词个数); 不成功有以下两种情况:

a $D=D+1$, 直至 $D=K(I)$, 此单词找不到, 扣出这一单词分值, 实际上这是多余单词, 暂定为总分值的 1%, 若 $FZ>0$, 继续, 即从下一单词开始, 转(4); 否则, 转(6)。

b 单词找到, 但不在相应位置, 那么这一单词前的那些为减少的单词, 扣出相应的分值, 即, $P1=D-1, FZ=FZ-DF(I, C)$ (C 从 P 开始, 直至 $P1$), 然后若 $FZ>0$, 继续, 即从下一单词开始, ..., 转(5), 否则转(6)。

③ 目标句比较的单词已经是最后一个吗? 是, 有两种情况:

a 待译句还剩有单词 $T=E-J$, 扣出分为总分值的 $T\%$, 若 $FZ>0$, 转 b; 否则, 转(6)。

b $T=E-J=0$, 为后选语句, 确定 $JM(Y)=HW(I)$, 此外, 这一句的现有分值记为 $JF(Y)$, $Y=Y+1, N=Y$, 转(6); 否则, $J=J+1, D=P+1$, 转(2)。

(4) $J=J+1, D=P$, 转(2)。

(5) $J=J+1, D=D+1$, 转(2)。

(6) $I=I+1, I>DM$ 转(7); 否则, 转(1)。

(7) $Y=1?$ 是, 转 a。

(8) $Y=2?$ 是, $I=1$, 转(10)。

(9) 求 $JF(Y)$ 的最大值, 设 $Y=1, 2, \dots, N-1, I=1, J=J+1$;

a 比较 $JF(I)$ 与 $JF(J)$, 若 $JF(I)>JF(J)$, 转 c

b $I=J$, 转 c。

c $J=J+1, J<N$ 转 a。

(9) 近似译文输出, 即输出 $JM(I)$ 对应的 $YH(I)$ 。

(10) 输出“此句暂时不能翻译”。

4.3 近似翻译算法中使用的符号说明

$JM(I)$: 最后的翻译语句;

$JF(I)$: 近似翻译一个语句可以扣除的分(剩余部分);

DM : 双语实例的条数;

$JM(Y)$: 满足近似翻译要求的后选语句;

$HW(I)$: 汉语例句;

$YH(I)$: 英语例句;

T : 待译句与目标语句相比剩余单词个数;

E : 待译句的单词个数;

Y : 满足条件的近似目标句个数;

FZ : 待译句可以扣除的最大分;

$DF(I, C)$: 待译句的单词分值, 其中 I 为某一汉语例句, C 为例句中某一单词。

5 结束语

文中涉及的算法是我们工作的核心, 因此作以下讨论:

(1) 在规则集中可能出现 AND 条件和 OR 条件, 由于所有条件满足才可认为规则匹配, 因此所有 AND 条件作为一个字符串最后变换成 M_i 。对于 OR 条件, 由于只要其中一个条件满足即可认为规则已匹配, 所以每一 OR 条件单独作一个字符变换成 $M_{i1}, M_{i2}, \dots, M_{id}$ (d 为 OR 条件个数), 采用链接方式。

(2) $M_i = \text{INT}((C_{\max} - C_{\min})/a)$, 公式压缩索引空间, $C_{\max} - C_{\min}$ 首先把索引空间压缩为 $0 \sim C_{\max} - C_{\min}$ 个。倘若还大, 取 $a>1$ 进一步压缩, 但这可能使多条规则因 $F(M_i) = F(M_j)$ 而采用链接方式。

(3) 规则索引算法经过对 500 余条规则集的实验, 比规则的顺序匹配方式平均速度快 5 倍以上, 这说明了该算法的高效率。

(4) 再对 $R_i, C_i, M_i, F(M_i)$ 四者关系作一简要说明: R_i 是规则集中任一规则, C_i 是相对的十进制值, M_i 又是 C_i 相对的压缩值, 而 $F(M_i)$ 是一维数组记录索引。例如, 某条规则条件部分算出 $M_i=50$, 它对应 $F(50)$ 这条索引给出的规则结论; 反之, 若某条规则算出 $M_i=500$, 就对应 $F(500)$ 这条索引给出的规则结论。

(5) 关于实例近似翻译算法, 经过对 500 个语句的实验, 参考对象是 CMU 的 EBMT 实验系统相似度的计算方法, 效率相差无几, 但我们的算法中词汇特征的计算简单化非常明显, 计算的复杂度大大降低。

参考文献:

- [1] Yael Karov, S Edelman. Similarity -Based Word Sense Disambiguation[J]. Computational Linguistics, 1998, 24(1): 41-59.
- [2] Sergei Nirenburg, Constantine Domasheny, Dean J Grannesl. Two Approaches Matching in Example-Based Machine Translation[A]. Proc of TMI-93[C]. 1993, 47-57.
- [3] 赵铁军. 机器翻译原理[M]. 哈尔滨: 哈尔滨工业大学出版社, 2001.
- [4] 杨宪泽. 自动翻译的词处理及其算法[J]. 计算机工程与科学, 2003, 25(4): 69-71.
- [5] 杨宪泽. 基于实例的机器翻译处理方法[J]. 计算机工程, 2003, 29(21): 51-53.
- [6] 杨宪泽. 子域映射快速排序法研究[J]. 科学通报, 1990, 35(15): 1119-1200.
- [7] 杨宪泽. 长记录位置不变的排序算法[J]. 软件学报, 1993, 4(2): 48-52.