一种汉语句子间相似度的度量算法及其实现

张民* 李生* 赵铁军* 周明**

(*)哈尔滨工业大学计算机系,哈尔滨,150001

(**)清华大学计算机系,北京,100084

摘要:本文主要论述了一种基于词的汉语句子之间相似度的计算方法,及其在汉外辅助翻译和辅助写作系统中的应用。首先对基于实例的机器辅助翻译及其所要解决的几个问题作一简单介绍。然后,详细描述了基于词的汉语句子之间相似度的计算方法。最后给出了算法的评价及其实现过程。

A Word—Based Approach for Measuring the Similarity between two Chinese Sentences

Min Zhang* Sheng Li* TieJun Zhao* Min Zhou**

(*)Department of Computer Science and Engineer, HIT, Harbin, 150001, P.R. China

(**)Computer Science Department, Tsinghua University, Beijing, 10084, P.R. China

ABSTRACT: This paper presents a brief description of Example—Based Machine Translation (EBMT) and addresses an important problem in EBMT—how to measure similarity between a sentence fragment and a set of stored examples. A new method is proposed that computes similarity according to the segmentation words and the thesaurus. Then, an evaluation is given by some experiments.

1. 引 言

八十年代以来,机译进入了空前繁荣的发展阶段,国际计算语言学界围绕机译的战略目标、理论方法等展开了激烈的讨论。一般来说,机译可分为经验主义(empiricism)和唯理主义(rationalism)两种方法[2]。传统的唯理主义方法是基于理性语言学(mentalinguistics),其试图通过建立一种语言学模型来表示、获取语言学中海量、模糊性知识;而经验主义则是基于心理语言学(psycholinguistics),通过对大规模真实语料的加工和分析来处理语言学中的复杂问题[2]。

传统的机译方法(rationalism, e.g. rule—based)中,知识的表示、知识的获取和知识的冲突、冗余其完备性都未能得到很好的解决。自然语言是个无限集,其单语或多语映射知识必然是海量和离散的,而知识自动获取是AI一直未能彻底解决的问题,因此,目前被称为

第二代的唯理主义方法暴露出很多弊端，很多学者对其提出尖锐批评[17]。而以基于统计(statistical－based)和基于实例(example－based)为代表的经验主义方法则避免了知识的手工编码和手工获取过程，从包含绝大多数语言现象的大规模真实文本语料中抽取翻译规律，这种方法被英国学者 John Hutchins 称为第三代 MT 方法[17]。本文所要论述的正是基于实例方法中的一个子问题。

本文所介绍的方法就是如何在大规模双语例句库中快速、准确地抽取和输入句子相似的例句，这也正是 EBAMT 所要解决的主要问题。

2. 基于实例的辅助翻译

基于实例的方法是通过模拟相似例句的翻译过程来实现的。在这种类型的翻译系统中，有一个大规模的双语或多语语料库作为知识源，输入一个待翻译的源语言的句子，从语料库中抽取和输入句最相似的例句，通过对例句的目标语言句子的调整来实现输入句的翻译过程。

一般说来，一个基于实例的机译系统由以下几个部分组成：

①一个双语语料库和双语对齐算法

②一部同义词(反义词、语义)词典

③最佳匹配抽取算法

④目标语调整算法

为了构造一个基于实例的系统，必须解决以下几个问题：

①建立双语对齐映射关系——这种对齐可以是在字符级、单词级、短语级、亚句子级或句子级。

②相似度计算(similarity calculation)实例抽取(example selection)和实例覆盖("cover" algorithm)——计算两个源语言句子之间的相似度，从例句库中抽取和输入句最相似的一些例句，并决定哪些句子组是对输入句的最佳"覆盖"。

③目标语的调整——通过对相似句目标语的调整实现输入句的翻译。

目前，关于这三个问题的研究很多。对于问题①，文献[13]、[14]给出了几种句子级的对齐算法；对于问题②文献[1,2,3,7,8,10,11]都给出了不同的算法，文献[4]给出了句子级和非句子级相似度的两种计算方法及其评价。但是，这三问题还远未彻底解决。本文针对问题②给出一种匹配算法。

3. 基于词的最佳匹配算法

3.1 匹配的层次

第二节问题②，即相似度计算和实例抽取，是基于实例的辅助翻译系统中最关键的问题。这个问题可细分为：

①匹配层次的确定——决定匹配是在句子级还是非句子级或称亚句子或部件级("text unit")。如果是在非句子级,就要有分割算法和覆盖算法("chunking" and "cover")。

②相似度的定义和计算——如何定义两个句子之间的相似度及其计算方法。③最小时间花费——即相似度计算和实例抽取的效率。

3.1.1 匹配"部件"的确定

文献[2,6,9,10,12]给出了几种"chunking"和"cover"算法。

句子的边界是非歧义的,很好确定,因此,一般把整句作为匹配的部件。但是,句子包含信息非常丰富,并且句子太长。匹配部件越长,完全匹配的概率就越低,这样系统的柔韧性和鲁棒性就比较差。

相反,如果选择"亚句子"作为匹配部件,虽然完全匹配的概率增大了,但"亚句子"的边界很难确定,特别是汉语短语的边界。这样,边界的歧义性增大,译文质量就会下降,另外,相应的"覆盖"算法和"调整"算法还很不成熟,系统实现比较困难。

在本系统中,匹配是在整句级进行的。这一方面是由系统的"辅助性"特点决定的,更重要的是由算法的特点决定的。

输入句和例句的长度往往相差很大,单词的个数不一致,因此,输入句的每个单词和例句中的每个单词都可能是相互相关的,这种相关性可通过相对位置进行加权处理。这种词一级的相关性、词(同义词)的同现(类似于 n-gram),评价值的向后叠加以及算法的递归定义[参见 3.2 节]决定了,算法不仅可以抽取出句子级相似例句,而且"亚句子"级相似也会给出一个很高的评价值,所以也可抽取出非句子级相似的例句。本算法可根据评价值的不同选出任何层次的相似例句。

3.1.2 算法所需知识

文献[1~12]给出了几种匹配算法及其评价。文献[4]给出了等价类、加权值和所需背景知识的 12 级分类。

基于词的计算方法目前是最流行的,依据词的形态变化、同义词、反义词及更进一步的语义距离来判断孤立单词之间的相似度,再通过这种词间相似度的不同组合来确定句子之间的相似度[4]。另外,还有很多其他算法,例如,语法规则驱动的[1,9,10,12],基于字符的[6],各种混合式和纯净式算法[2,3,4,11]。日本学者 Sato 在其 CTM 系统中实现了一种基于字符的算法,非常简单,但很有效[6]。语法规则驱动的方法似乎应该很有效,但是源语言的分析又引入了基于规则的方法,变得非常复杂,于是有人提出了基于模式的方法。

对于两个句子之间的相似度至今没有人给出一个令人信服的准确定义,其实也很难定义。一般说来,在语义或更高层的相似是指人对两句的理解结果是相同或相似的;在译法层次上,指的是整句或亚句的语法或译法结构相似,这两种相似都非简单的字符串相似。但计算两个句子之间的相似度是否一定要引入语法、语义知识呢?能否运用一种数学方法计算字符串相似来判断整句相似[例如,汉语自动分词中的最大匹配法]? 基于此,本文提出一种经验型的评价公式。

本算法吸收了字符驱动和单词驱动算法的优点,不需要任何语法、语义信息。它利用同义词表[15]计算两个单词的语义距离,再利用词的相似计算整句的相似,本算法具有如下优点:

①不需要任何词法、语法、语义分析。

②例句库仅需作分词标记,即便是词性标记都是不需要的。

③词典是必要的，但词典是供分词用的，即仅需汉语词条，其他信息均不用填写，只需一部分词词典即可。

④对输入句类型没有要求，可以是整句，也可以是短语、搭配、惯用语等等。

3.2 最佳匹配算法

3.2.1 汉语的自动分词

双语语料中的每个汉语句子都进行了分词处理，输入句也要进行分词处理，系统采用一种基于评价的快速分词算法[16]。

3.2.2 基于词的最佳匹配算法

定义 1：输入句

$$inp-str=a_1\sim a_x$$

分词之后：

$$inp-str=/a_1\cdots a_i/a_{i+1}\cdots a_k/\cdots/a_m\cdots a_x$$
$$=WDI_1+\cdots+WDI_n$$

定义 2：例句

$$exa-str=b_1\sim b_y$$

分词之后：

$$exa-str=/b_1\cdots b_j/b_{j+1}\cdots b_l/\cdots/b_w\cdots b_y$$
$$=WDE_1+\cdots+WDE_n$$

定义 3：单词 i 的语义描述

$$Sense—Word(i)=x_1\cdots\cdots x_n|\cdots\cdots|y_1\cdots\cdots y_m$$

详见文献[15]

定义 4：单词 i 的语义分类个数

$$Num-Sense-Class-Word(i)=\begin{cases}1 & \text{if 单词 i 为单义词}\\ n & \text{if 单词 i 为多义词}(n>1)\end{cases}$$

定义 5：单词 i，单词 j 的语义包含关系：

$$Inclusion-Sense(i,j)=\begin{cases}1 & \text{if}\begin{cases}Num—Sense—Class—Word(i)=1\\ Num—Sense—Class—Word(j)\neq 1\\ Sense—Word(i)=\text{one of }Sense—Word(j)\end{cases}\\ 0 & \text{else.}\end{cases}$$

定义 6：孤立单词 i，j 的相似度

$$SIM-WORD(i,j)=\begin{cases}1 & \text{if } Sense-Word(i)=Sense-Word(j)\cup wdI_i=wdE_j\\ 0.6 & \text{else if } Inclusion-Sense(i,j)=1\ \cup Inclusion-Sense(j,i)=1\\ 0.4 & \text{else if}\begin{cases}Num-Sense-Class-Word(i)\neq 1\\ Num-sense-Class-Words(j)\neq 1\\ \text{one of }Sense-Word(i)=\text{one of }Sense-Word(j)\end{cases}\\ 0.2 & \text{else if strcmp }[Sense-Word(i),Sense-Word(j)]>3\\ & \text{(借用 C++中的函数)}\\ 0 & \text{else}\end{cases}$$

算法 1：位置加权后的单词 i，j 的相似度

$$Cor—SIM—Word(i,j)=\begin{cases}0 & \text{if } i=-1 \cup j=-1\\ Cor—SIM—Word(i-1,j-1)+SIM—WORD(i,j)\times U & \text{if}(0\leqslant i\leqslant n,0\leqslant j\leqslant m)\end{cases}$$

$$U=\begin{cases}1 & \text{if}\left|i-\frac{j}{|b|}\times|a|\right|<4\\ 0.7 & \text{else}\end{cases}$$

算法 2：节点 i，j 的相似度

$$①\quad S—node(i,j)=\begin{cases}0 & \text{if } i=-1 \cup j=-1\\ \max\begin{cases}S—node(i-1,j-1)+\min(Cor—SIM—Word(i,j),W),\\ S—node(i-1,j),\\ S—node(i,j-1)\end{cases} & \text{if } (0\leqslant i\leqslant n,0\leqslant j\leqslant m)\end{cases}$$

$$②\quad S—node(i,j)=\begin{cases}0 & \text{if } i=-1 \cup j=-1\\ \{S—node(i-1,j-1)+\min[Cor—SIM—Word(i,j),W]\}/3+\\ \quad\{S—node(i-1,j)+S—node(i,j-1)\}/3 & \text{if } (0\leqslant i\leqslant n,0\leqslant j\leqslant m)\end{cases}$$

$W=3$

算法 3：句子之间的相似度

① $SIM—SENT—\text{Ⅰ}=S—node(n,m)$

② $$SIM—SENT—\text{Ⅱ}=\frac{\sum_{i=0}^{|a|}\sum_{j=0}^{|b|}\cdot S—node(i,j)}{|a|\cdot|b|}$$

通过以上给出的定义和算法，算法 3 最终给出句子之间相似度计算的两种计算公式。算法 3 是两个经验型公式，这种算法只能用于辅助翻译。

算法 3－①是对连续词串最大相似值的评价，适合于亚句子或短语级相似计算；算法 3－②是对整句相似的均值计算，适合于整句相似度的计算。在算法 3－①中，相对位置加权参数 U 恒等于 1。

算法 1 对连续匹配单词相似值向后累加，从第一组一直交叉累加到最后一组，特别适合于算法 3－①。但考虑到算法③－2，可对算法 1 改进如下：

$$Cor—SIM—Word(i,j)'=Cor—SIM—Word(i,j)MOD\ \underline{Q}$$

$Q=W+1$

在本算法中，使用了三个经验型参数“$U=\begin{cases}1\\ 0.7\end{cases}$”，“$W=3$”，“$Q=W+1$”。“U”反映了两个句子之间单词的相对位置；“W”是对连续匹配单词的最大加权值；“Q”类似于“W”，但“W”是从整句角度考虑的连续匹配最大加权值，“Q”是从局部角度给出的连续匹配最大加权值。

3.2.2 算法的预处理

例句库中存有 10 万条例句，考虑到时间的因素，不可能计算输入句和每个例句的相似度。本系统采用一种类全文检索技术通过词索引，从大例句库中选择出一个子集，这个子集包含了算法所要确定的最相似例句。预选过程如下：

①建立例句库的词索引表。

1) $\begin{cases} word_1: No.\ of\ Example_1 \cdots\cdots No.\ of\ Example_n. \\ \vdots \\ word_k: No.\ of\ Example_1 \cdots\cdots No.\ of\ Example_n. \end{cases}$

2)第一个例句的文件指针，…… 最后一个例句的文件指针

②对于切分好的输入句(含有 N 个单词)，查找每个单词及其同义词在例句库中所出现的句子号。合并每个单词及其同义词的句子号，这样形成 N 组句子标号(去掉重复的句子号，共有 m 个句子号)，利用全文检索的"与"搜索技术，计算输入单词序列在 m 个句子中的同现值。这个同现值是在忽略单词顺序后的输入句和例句的相似值，称其为预选值。

③按照预选值的大小，选取前 100 个例句作为从大例句库中选出的子集。这样，只计算这个子集中的例句和输入句的相似度即可。

4. 系统和实验及其评价

系统是用 Visual c++ 语言在 windows 下实现。系统由以下几个部分组成：

- 知识库：
  - ①同义词典：约 5 万词[15]
  - ②分词词典：约 6 万词[16]
  - ③例句库：约 10 万双语例句
- 算法：
  - ①最佳相似例句抽取算法
  - ②类全文检索的预处理算法
- 人机界面：
  - 辅助翻译平台

本文仅举一例说明系统的实验结果：

- 输入句："我当然愿意了解她们的要求。"
- 输出相似例句：
- "我认为我当然愿意了解她们的要求。"

I think that, of course, I want to know what their opinion is .

SIM－SENT＝22.00， W＝4， Q＝5

- 当然我想知道你的意见。

Of course, I want to know what your opinion is.

SIM－SENT＝20.00， W＝4， Q＝5

- 我很想知道他的决定是什么?

I' m anxious to know what his decision is ?

SIM－SENT＝13.00， W＝4， Q＝5

下面给出系统的评价：

①平均抽取时间：

a. 预处理时间：

pre－time ＝(每句的平均单词数×平均同义词数)×读一个单词句子号时间＋计算单词同现时间＋排序时间

＝15×5×10＋200＋100＝1050(millisecond)＝1.05(second)

b. 相似计算时间：

SIM－time＝N×10＝100×10＝1000(millisecond)＝1.00(second).

c. 总的时间消耗：

Tol time ＝pre－time＋SIM－time＋读取相似的时间＋排序时间

＝1050＋1000＋100＋100＝2250(millisecond)＝2.25(second).

②抽取质量：

一般说来，从抽取例句的质量来对这种辅助翻译系统进行评价是很困难的，不同的用户对翻译同一句存在不同的疑问，当用户对系统选择的例句认为有用时，则认为是有效的；否则，则认为是无效的。

但是，我们认为可以从评价值最高的前20句中至少可获得70%的有用信息。

参 考 文 献

[1] Osamul FuRusE and Hitoshi IIDA (1992), An Example－Based Method for Transfer－Driven MT, TMI－92 pp. 139～148

[2] Daniel Jones (1992), Non－hybrid EBMT Architectures, TMI－92 pp. 163～170

[3] Sato(1991), Example－Based Translation Approach, Telephony Research Laboratories, pp1～16, 1991

[4] Sergei Nireaburg (1993), Two Approaches to Matching in EBMT, TMI—93 pp. 47～57

[5] Peter F. Brown, John cocke. (1990), A Statistical Approach to MT, Computational linguistics Volume 16, Number 2, June 1990.

[6] Satoshi SATO (1992), CTM: An Example－Based Translation Aid System, COLING－92 pp. 1259～1263.

[7] Sato, S and Nagac M (1990), Toward Memory－Based Translation, COLING－90

[8] Nagao, M. (1984), A framework of a mechanical translation between Japanese and English by analogy principle, Artificial and Human Intelligence, NorthHollaud, pp. 173－180

[9] Lambros CRANIAS. etc. (1994), A Matching Technique in Example－Based Machine Translation, COLING－94 pp. 100～104.

[10] Osamu FURUSE etc (1994), Constituent Boundary Parsing for Example－Based Machine Translation, COLING－94 pp. 106～111.

[11] Hideo Watanable (1992), A similarity－Driven Transfer system, COLING－92 pp. 770～774.

[12] Hiroshi Maruyama (1992), Tree Cover Search Algorithm for EB Translation, TMI－92 pp. 173～184.

[13] P. Fung, K. W. chuch (1993), K—vec: A new Approach for Aligning Parallel Texts, COLING－94 pp. 1096～1104.

[14] kennetch church (1994), Aligning Parallel Texts: Do methods Developed for English－French generalitzeto Asia Language? Reported from Tsinghua University

[15] 梅家驹(1983),《同义词词林》,上海人民出版社,1983

[16] 张民,李生,赵铁军等(1994),CEMT－Ⅲ汉英机器翻译系统的研究,情报学报,Vol. 13,NO. 1

[17] John Hutchins (1993), Latest Developments in Machine Translation Technology: Beginning a New Era in MT Research, MT Summit IV 1993. 7