文章编号: 1003－0077(2005)01－0021－09

基于词类串的汉语句子结构相似度计算方法①

王荣波, 池哲儒

(香港理工大学 电子及资讯工程系 多媒体信号处理中心, 香港)

摘要: 句子相似度的衡量是基于实例机器翻译研究中最重要的一个内容。对于基于实例的汉英机器翻译研究, 汉语句子相似度衡量的准确性, 直接影响到最后翻译结果的输出。本文提出了一种汉语句子结构相似性的计算方法。该方法比较两个句子的词类信息串, 进行最优匹配, 得到一个结构相似性的值。在小句子集上的初步实验结果表明, 该方法可行, 有效, 符合人的直观判断。

关键词: 人工智能; 机器翻译; 基于实例机器翻译; 汉英机器翻译; 句子相似度衡量; 自然语言处理

中图分类号: TP391 文献标识码: A

A Similarity Measure Method of Chinese Sentence Structures

WANG Rong-bo, CHI Zhe-ru

(Center for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, China)

Abstract: Example-based machine translation(EBMT) is an important branch of machine translation that has been studied extensively for about twenty years. So far, some progresses have been gained because of researchers' hard work. Sentence similarity measure certainly is one of the most important problems addressed in EBMT. For EBMT from Chinese to English, the performance of similarity measure of Chinese sentences affects directly final translation result of an input sentence. In this paper, we proposed a similarity measure method of Chinese sentence structures for example-based Chinese to English machine translation. In this method, the algorithm performs the optimal matching between the word type sequences of two compared sentences. The preliminary experimental results show that the measure method works well when it is tested on a small dataset.

Key words: artificial intelligence; machine translation; example-based machine translation; Chinese-English machine translation; sentence similarity measure; natural language processing

1 引言

自从计算机的发明, 人们就希望它能实现自动翻译, 因此, 机器翻译的研究被提出来。到目前为止, 机器翻译研究已经经历了半个多世纪, 有很多的进展, 但更多的是挫败。基于实例机器翻译(EBMT)的提出, 使机器翻译的研究多了一个很重要的分支, 同时, 也给MT研究带来新的希望[1~4] 越来越多的研究人员投入到EBMT的研究当中。

基于实例机器翻译的基本原理是: 当输入一个待翻译的句子时, 系统从存有海量句子实例的双语(汉英)句子库中搜索得到与输入句最相似的句子, 再以该句子的译文为模板, 根据找出的该句子与输入句子间的差异, 经过一些转换生成步骤, 最后生成的结果, 就是输入句子的翻

① 收稿日期: 2004－05－26
作者简介: 王荣波(1978－), 男, 博士生, 主要研究方向为计算语言学, 机器翻译, 模式识别.

译。在 EBMT 中，句子相似度的衡量是一个非常关键的步骤，直接影响得到的模板的质量，最后影响译文的正确性。

在基于实例的机器翻译中，句子相似度衡量被用于识别双语语料库中与源句子最相似的句子。到目前为止，有很多的衡量方法被提出来。不同的方法很大程度上依赖于汉语句子的不同表示形式[5]。可以把这些方法分为两类：基于字符串匹配的方法和基于句法/语义信息的方法。前者的衡量方法，主要采用字符串匹配，进行少量或者不进行句法分析。例如，在两个待比较的句子中，计算相同的词数与两个句子总词数的比例。基于句法/语义的方法更多地考虑句子的句法结构和语义信息，通过对每个句子进行句法结构和语义分析，得到句子的结构框架。

在文献[7]中，作者提出了一种英语句子相似度计算方法。在该方法中，作者定义了一个距离函数，通过计算两个句子的最小编辑操作数来衡量两个句子的距离。编辑操作包括插入，删除，替换单个单词项等。也就是说，从第一个句子的词序列变换到第二个句子的词序列需要多少次编辑操作，编辑操作的数目体现了距离的大小。该方法适用于英语句子及与英语同属于一个语系的句子，但它并不适用于汉语句子，因为汉语句子中的词并没有形态和时态的变化，很难用编辑操作来实现从一个句子词序变换到另一个句子的词序。

对于汉语句子相似度衡量，到目前为止，也有很多的方法被提出来。其中有通过计算语句中词语间的相似度来得到整个语句间的相似度[6]。对于词语相似度的计算，在文献[8]中，作者利用上下文的词汇向量空间模型来近似地描述词汇的语义，再在此基础上定义词汇的相似关系。两个字符串字符相似度的定义也能用于两个句子的相似度计算[9]。另外，穗志方博士提出了一种基于骨架依存分析的方法[10]。在该方法中，首先对汉语句子进行谓语中心词的识别。如果两个句子的谓语中心词相似，再判断是否它们的直接支配成分之间是一一对应的。如果是，再进一步计算谓语中心词之间以及对应的直接支配成分之间的相似度之和，作为两个待比较句子的相似度值。但是，如果对所有句子进行骨架依存分析，比较耗时，并没有统一的标准，需要很多人为参与。

其中一种基于字符串匹配的方法是计算两个句子中相同的词数与总词数的比例。计算公式定义如下：

$$\frac{2c}{m+n} \tag{1}$$

其中 m，n 分别表示两个句子的词数。而 c 是两个句子中相同词的数目。该方法的主要优点是计算简单。它可以用于汉语和英语句子的相似度计算。而它的主要缺点也是由过于简单引起的，那就是它基本没有考虑句法和语义信息，不能区别在语法或语义上相似的句子。也就是说，如果两个汉语句子结构上非常相似，但是在它们之间并没有相同的词，那么其相似值等于零。这是该计算公式的一个难以避免的缺点。这说明，我们在计算两个汉语句子的相似度的时候，需要同时考虑句子的结构信息，这样会使相似度衡量更加合理准确。

本文提出了一种汉语句子结构相似度衡量方法。该方法比较两个汉语句子对应的词类串，找到最优的词类之间对应关系，利用这种对应关系计算两个句子的结构相似性。初步实验表明，该方法是可行的，也是有效的。

2 本文的方法

2.1 信息要求

汉语句子结构相似性的计算，要体现以下几种信息。

(1)连续性

句子结构成分具有连续性,才能说明句子具有某种结构。因此,在计算句子的结构相似性的时候,需要考虑并体现连续性信息。

(2)整体性

句子结构具有整体性,只有句子中的成分前后呼应,才能说明句子具有某种结构。因此,在计算句子的结构相似性的时候,需要考虑并体现整体性信息。也就是说,把句子结构作为一个整体来考虑。

2.2 主要思想

我们提出的方法的主要思想是:对两个汉语句子的词类序列,结合词类的权值信息,进行匹配,得到最优的匹配结果,即最后的结果使两个待比较句子的词类序列相似度值最大。

2.3 算法

定义 **1**:合并条件(Joint Condition):如果两个词类相同,那么它们可以被合并。

定义 **2**:合并结点(JointNode):在两个词类序列中,如果符合合并条件的两个结点合并后,该结点称为合并结点。

定义 **3**:当前结点 1(CurrentNode1):在较短的词类序列中,当前正在被处理的结点。

当前结点 2(CurrentNode2):在较长的词类序列中,当前正在被处理的结点。

定义 **4**:当前合并结点(CurrentJointNode):在反向匹配调整过程中,前一个(最近一个)被合并的结点称为当前合并结点。

定义 **5**:有向图表示(Direct Graph Expression):对于汉语句子的词类序列,我们用一个带有词类标记的结点和一条到该结点的有向连线来表示一个词类。该连线赋予对应该结点词类标记的权值。起始结点用“ *”表示,终点用“ #”表示,其连线赋予权值为 0 的 w_0。

算法的系统流程图如下:

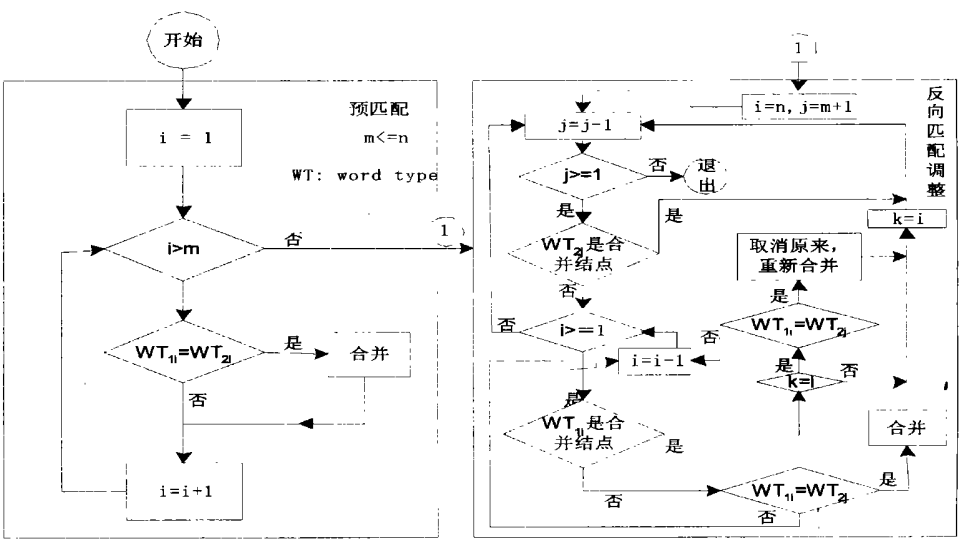

图 1 系统流程图

首先,假设我们有以下两个词类序列($n \geqslant m$),分别对应句子 S_1 和 S_2。

S_1: s_1 s_2 s_3 … s_i … s_{n-2} s_{n-1} s_n.

S_2: t_1 t_2 t_3 …t_j …t_{m-2} t_{m-1} t_m.

其中, s_i ($1 \leqslant i \leqslant n$)表示句子 S_1 中的第 i 个词类标记, t_j ($1 \leqslant j \leqslant m$)表示句子 S_2 中的第 j 个词类标记。

为了便于说明算法, 以上句子(S_1, S_2)词类序列的有向图表示如下。

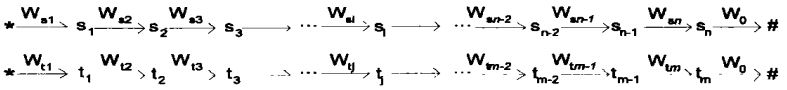

图 2　对应句子 S1 和 S2 的词类序列(包含词类权值信息)

基于图 2, 我们可以计算两个句子结构的相似性。其计算过程可以分解成两个方向的子过程。其一是从左到右进行预匹配, 其二是从右到左进行反向匹配调整。

其具体步骤如下:

(1)从左到右预匹配。

首先, 对相同位置的词类进行比较。如果相同, 那么合并其对应的节点, 使其成为合并结点。否则, 继续比较下个结点, 直至有其中一个句子结束, 即较短的句子结束。在该过程完成以后, 有可能存在一个或多个环结构, 如下图所示。

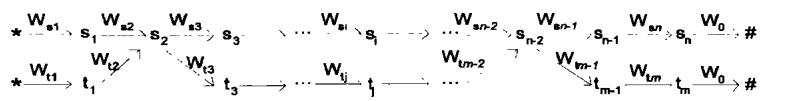

图 3　完成预匹配后状态

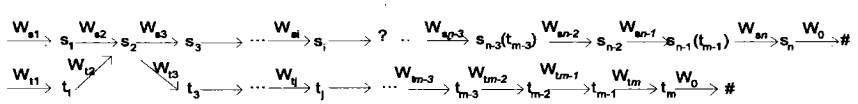

图 4　完成预匹配后状态

(2)从右到左反向匹配调整。

在完成预匹配后, 接着进行反向匹配调整, 使最后的相似度值最大。

a. 把较短的词类序列(S_2)中的倒数第一个词类结点(结点“ #”除外), 作为当前结点 1, 把较长词类序列(S_1)中最后结点(结点“ #”除外)作为当前结点 2, 并把 S_1 中结点“ #”作为当前合并结点。如果当前结点 1 不是合并结点, 开始反向搜索过程。否则, 不进行反向匹配过程。

b. 如果当前结点 1 为结点“ *”, 则结束。

如果当前结点 1 是合并结点, 则把该结点同时作为当前合并结点和当前结点 2, 并进行步骤 e。

c. 如果当前结点 2 是结点“ *”, 则进行步骤 e。

如果当前结点 2 是合并结点, 但非当前合并结点, 则重新选当前合并结点作为当前结点 2, 并进行步骤 e。

如果当前结点 2 是当前合并结点, 并且当前结点 1 等于当前结点 2, 那么取消原来的合并, 把当前结点 1 与当前结点 2 合并, 并仍然把当前结点 2 作为当前合并结点, 进行步骤 e。

如果当前结点 2 是当前合并结点, 但是当前结点 1 不等于当前结点 2, 那么进行步骤 d。

如果当前结点2不是合并结点，但是当前结点1等于当前结点2，那么合并着两结点，标记当前结点2为当前合并结点，进行步骤e。

如果当前结点2不是合并结点，并且当前结点1不等于当前结点2，进行步骤d。

d. 考虑当前结点2的左边第一个结点，使它为当前结点2，进行步骤c。

e. 接着考虑当前结点1的左边第一个结点，使它为当前结点1，进行步骤b。

现在以图4为例，说明以上算法。

图4中的句子有向图表示是完成预匹配以后的状态。从图中可知，我们假设第一个句子中的 s_{n-3} 与第二个句子中的 t_{m-3} 相同，s_{n-1} 与 t_{m-1} 相同。但由于它们在句子中的位置序号不相同，所以在预匹配阶段没有实现匹配，而要在反向匹配调整中确定是否能实现匹配。在从右到左反向匹配调整过程中，我们首先把较短句子(S_2)的最后一个结点(t_m)，作为当前结点1。因为它不是合并结点，所以进行反向搜索过程。接着把较长句子(S_1)中最后结点(s_n)，作为当前结点2，并把 S_1 中结点“ #”作为当前合并结点。

当考虑 t_m 时，直到当前结点2为结点 s_2，都没有找到相同的结点，结束此循环。接着考虑 t_m 前面一个结点 t_{m-1}。由于存在与 t_{m-1} 相同的结点(s_{n-1})，所以在这两点之间建立连接，并把 s_{n-1} 作为当前合并结点。类似地，结点 t_{m-2} 也不能在 S_1 中找到与之相匹配地结点，而 t_{m-3} 能与结点 s_{n-3} 合并，建立连接。最后的匹配调整结果如下所示。

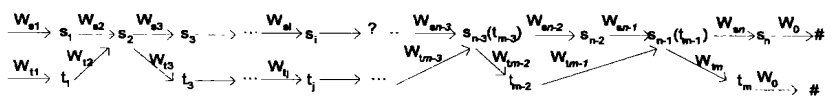

图5　完成反向匹配调整后状态(对应于图4)

算法描述如下：

```
k=m; l=n; CurrentNode1= t_k; CurrentNode2= s_l CurrentJointNode=“ #”
IF(CurrentNode1==jointNode)THEN STOP;
ELSE{
    WHILE(CurrentNode1 !=“ *”)
    {
      IF(CurrentNode1==JointNode)
      {CurrentNode1= CurrentNode1; CurrentNode2=CurrentNode1; }
      ELSE{
          TAG=1;
          WHILE(CurrentNode2!=‘ *’ && TAG==1)
            {
              IF(CurrentNode2==JointNode &&
                 CurrentNode2!=CurrentJoint Node)
              {CurrentNode2= CurrentJoinNode; TAG=0;}
              ELSE IF(CurrentNode2==CurrentJointNode&&
                       CurrentNode1==CurentNode2)
                       {
                         Cancel Joint(CurrentNode2);
                         Joint(CurrentNode1, CurrentNode2);
```

```
                CurrentJointNode= CurrentNode2;
                TAG= 0;
               }
        ELSE IF(CurrentNode2 ! = JointNode&&
                CurrentNode1= = CurrentNode2)
               {
                Joint(CurrentNode1, CurrentNode2);
                CurrentJointNode= CurrentNode2; TAG= 0;
               }
        ELSE IF((CurrentNode2= = CurrentJointNode ||
                CurrentNode2 ! = JointNode) &&
                CurrentNode1 ! = CurrentNode2)
               { l= l-1; CurrentNode2= s_l; }
          }
      }
    k= k-1; CurrentNode1= t_k;
   }
  }
```

其中: 函数 Cance1Joint( ), Joint( )分别为取消合并函数和 合并函数

以上是汉语句子结构相似性计算中, 进行最优匹配的算法。在获得最优匹配后, 其结构相似度值的计算公式如下:

$$StruSim(s_1, s_2) = \frac{2\sum_{i=1}^{C} \frac{1}{D_i} W_i}{1 + \sum_{j=1}^{i} W_j} \Big/ \sum_{k=1}^{E} W_k \tag{2}$$

在公式(2)中, C 是两个词类序列中相同结点(合并结点)的数目。而 D_i 是一个环中, 非合并结点的数目。在匹配过程中, 如果碰到重复匹配, 需要替换的, 总是用前面的结点代替后面的结点, 这样能保证最后的匹配结果相似度值最大。值得注意的一点是, 在计算相同结点数目的时候, 我们不考虑初始结点“ *”。但在计算环内非结点数目的时候, 把它作为第一个环的起点。E 是两个词类序列中总的结点数(词类数, 重复计算)。W_i 是第 i 个相同结点(词类)的权值。W_j 是环中第 j 个结点的权值。W_k 是所有结点(词类)中第 k 个结点的权值。

考虑两种特殊的情况。

(1)两个句子结构完全相同。

如果两个待比较的句子结构完全相同, 有 $m= n$, $D_i=0$, $C= m$, $E=2m$。所以公式(2)等于 1, 表示两个句子结构完全相同。

(2)两个句子结构完全不同。

如果两个待比较的句子结构完全不相同, 那么, 两个词类序列中, 合并结点数目为 0, 即公式(2)等于 0, 表示两个句子结构完全不同。

2.4 举例

为了说明以上算法, 举例如下。

例 1　原句 1: 鲁迅浙江绍兴人。　　切分后: 鲁迅/n 浙江/n 绍兴/n 人/n 。/w

　　原句 2: 鲁迅浙江人。　　切分后: 鲁迅/n 浙江/n 人/n 。/w

对应的词类序列如下。

S_1: *n n n n w*.

S_2: *n n n w*.

以上词类序列的有向图表示形式如下图所示。

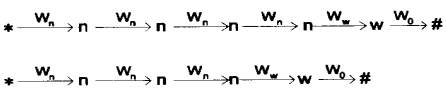

图 6　例 1 两个句子词类序列的有向图形式

经过预匹配后的结果如图 7 所示。经过反向匹配调整后的结果如下图所示。

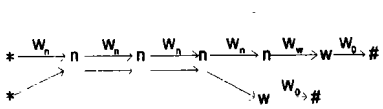

图 7　经过预匹配后的有向图形式

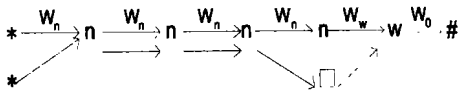

图 8　经过匹配调整后的有向图形式

从图 6 可知, $C=4$, $D_1=D_2=D_3=0$, $D_4=1$, $E=9$. 在实验中, 我们分配权值 3 给名词词类, 分配权值 5 给动词词类, 所有其它的词类, 我们都分配权值 1。根据结构相似度值的计算公式, 以上两个句子的结构相似度值为 0. 804。

3　实验结果及分析

该实验中, 我们采用了由清华大学周强博士提供的部分句子库。该句子库共有 3548 个汉语句子, 并已经完成了词语切分和标注。下面列出部分句子实例。

1025　[zj [dj 我/r　[dj 心里/s　[ap 很/d　踏实/a　]]] 。/w　]

1036　[zj [dj 我/r　[vp 有/v　[np 个/q　弟弟/n　] [vp [pp 在/p　大学/n] 读书/v　]]] 。/w　]

1047　[zj [dj 我/r　[vp 曾/d　[vp 送给/v　他/r　[np [mp 一/m　两/m　次/q　] 东西/n　]]]] 。/w　]

1058　[dj 我/r　[vp 昨天/t　[vp [vp 给/v　她/r　] [vp [vbar 写/v　了/u　] [np [mp 一/m　封/q　] 信/n　]]]]]

1069　[dj [tp 下班/v　后/f　] [dj [np 我/r　和/c　妻子/n　] [vp 去/v　商店/n　]]]

在上面的句子中, 句子前面的数字是句子的序号, 而其它符号, 像 vp, dj, v 等是词类标记或者是短语类型标记。符号‘ [’和‘] ’是边界标记。

在句子库中, 共有 31 类词语。在目前的实验中, 根据我们对各词类重要性的判断, 分配权值 3 给名词词类, 分配权值 5 给动词词类, 分配权值 1 给所有其它的词类。

在目前的实验中, 我们从句子集中选取了一些句子作为输入句子(源句子)。分别在句子库中查找与之结构相似的句子, 并且按照相似度从大到小排序。由于篇幅限制, 表 1 列出了部分计算结果。

表 1　汉语句子结构相似性计算的部分结果

实验用句子		结构相似度值	词串匹配相似度值
源句子 1	鲁迅浙江绍兴人。		
相似的句子	鲁迅浙江绍兴人。	0.960	0.888
	鲁迅浙江人。	0.818	0.675
	江水很凉。	0.600	0.000
	机器人的控制是很难的。	0.375	0.041
	公司在计算机的价格上进行了调整。	0.342	0.000
源句子 2	我喜欢农村的生活。		
相似的句子	我喜欢农村的生活。	1.000	1.000
	我特别喜欢农村的生活。	0.777	0.909
	他从事新技术的推广工作。	0.697	0.012
	我要刷牙。	0.613	0.018
	我会画图。	0.613	0.018
源句子 3	姐姐去买水果。		
相似的句子	弟弟去买冰激凌。	1.000	0.247
	姐姐去买水果。	1.000	1.000
	计算机能模仿人的思维。	0.864	0.000
	弟弟会读书。	0.854	0.015
	学校决定录用他。	0.838	0.000
源句子 4	校长同意派人出国。		
相似的句子	校长同意派人访问。	1.000	0.492
	校长同意派人出国。	1.000	1.000
	弟弟摘下一串葡萄来。	0.869	0.006
	妈妈要料理家务。	0.829	0.010
	大嫂气得抱着孩子发抖。	0.782	0.006

在表 1 中，我们列出了其中 4 个输入句子的实验结果。对于每个输入句子，我们从实验结果中选出 5 个最相似的句子。在表格中同时列出结构相似度值和基于词串方法的相似度值。相似句子按照结构相似度值从大到小排列。从实验结果可知，本文提出的方法，可以把结构相似的句子从数据库中提取出来。例如，当输入源句子 3（姐姐去买水果。）时，用我们的方法可以把与输入句子结构完全相同的句子（弟弟去买冰激凌。）搜索出来，其结构相似度值为 1，如表中所示，而它的基于词串的相似度值很小（0.247）。同时，对于词串相似性值为 0 的句子，本文的方法能够给出一个结构相似性的值，表示在两个句子之间的结构具有一定的相似性，尽管其值很小。

我们在小句子集上进行了实验。从实验结果可知，系统能把结构一样和非常相似的句子从数据库中找出来。其它部分相似或有点相似的句子，系统能给出一个从 0 至 1 的值。因为句子相似性的判断，并没有绝对的标准，只是一个模糊的概念。所以，我们并不能非常准确地用一个确定的数字来表示它们的相似性，只能把上述相似度值，看作是一个相对的概念，反应相似的趋势。

4　结论

汉语句子相似度的计算在基于实例的汉英机器翻译中，有着举足轻重的地位。本文提出了一种汉语句子结构相似度计算的方法。从实验结果可知，利用词类串信息进行句子结构相

似性计算是可行的。因为结构相似的句子,其词类串信息必然相似。当该方法应用于比较复杂的中文句子时,由于复杂句子的修饰成分比较多,会有匹配错误现象。目前的汉语句子分析技术,包括相似度计算技术,都集中在单句和简单句的处理。对于一些现有的技术,在进行试验时,作者需要专门选择简单句作为数据库。当单句分析处理技术进一步成熟以后,才可能对长句和复句有较好的处理效果。

在实际的基于实例机器翻译系统中,我们可以结合结构相似性的计算方法与基于词串的相似性计算方法。在它们之间选取一个平衡点,设置不同的权值,最后的计算结果作为句子相似性的值,以更加合理地衡量两个汉语句子的相似性。

在目前完全句法分析技术还不成熟的情况下,本文提出的方法对汉语句子结构相似性的计算是有效的。在以后的工作中,可以对句子先进行一些预处理,如去掉助词等一些不重要的词语。再应用本文所提出的方法进行结构相似性的计算。也需要对权值的分配进行研究,希望对词类分配的权值更符合句子使用实际。

参 考 文 献:

[1] M. Carl. Recent Research in the Field of Example-Based Machine Translation[A]. CICLing 2001, LNCS 2004.

[2] W. John Hutchins. Machine Translation: a brief history. Concise history of the language sciences: from the Sumerians to the cognitivists[M]. Oxford: Pergamon Press, 1995.

[3] Sumita, E. and H. Iida. Experiments and Prospects of Example-Based Machine Translation[A]. Proceedings of 29th ACL Meeting[C]. Berkeley, 1991, 185—192.

[4] 赵铁军. 机器翻译原理[M]. 哈尔滨: 哈尔滨工业大学出版社, 2000.

[5] K. Chidananda Gowda and E. Diday. Symbolic Clustering Using a New Similarity Measure [J]. IEEE Transactions on Systems, Man, and Cybernetic, 1992, 22(2).

[6] 李素建. 基于语义计算的语句相关度研究[J]. 计算机工程与应用, 2002, 38(7): 75—76.

[7] Federica Mandreoli, Riccardo Martoglia, and Paolo Tiberio. Searching Similar(Sub)Sentences for Example-Based Machine Translation[A]. In: Atti del Decimo Convegno Nazionale su Sistemi Evoluti per Basi di Dati(SEBD 2002), Isola d'Elba, Italy, 2002.

[8] 胡俊峰, 俞士汶. 唐宋诗中词汇语义相似度的统计分析及应用[J]. 中文信息学报, 2002, 16(4): 39—44.

[9] 李红莲, 何伟, 袁保宗. 一种文本相似度及其的语音识别中的应用[J]. 中文信息学报, 2003, 17(1): 60—64.

[10] 穗志方, 俞士汶. 基于骨架依存树的语句相似度计算模型[A]. 中文信息处理国际会议论文集(ICCIP'98)[C]. 北京: 清华大学出版社, 1998, 458—465.