情 报 学 报 ISSN 1000－0135
第 29 卷第 4 期 701 － 708 ,2010 年 8 月

JOURNAL OF THE CHINA SOCIETY FOR SCIENTIFIC AND TECHNICAL INFORMATION ISSN 1000－0135
Vol .29 No .4 ,701 － 708 August 2010

doi :10 .3772/j .issn .1000-0135 .2010 .04 .016

基于双序列比对的中文术语语义相似度计算的新方法[1)]

徐 硕[1] 朱礼军[1] 乔晓东[1] 薛春香[1,2]
(1. 中国科学技术信息研究所 ,北京 100038; 2. 南京理工大学 ,南京 210094)

摘要 针对中文术语的语义相似度计算问题 ,本文首先用数学语言对其进行了描述 ,然后仔细分析了求解该问题的传统计算方法 ,结果发现传统计算方法大都做了一个隐式假设 :组成两个术语的原子术语的顺序必须大体一致。换句话说 ,传统计算方法并没有考虑原子术语顺序的差异对构建两个术语的原子术语间对应关系质量的影响。为克服这个问题 ,通过类比分析 ,本文认为可将该问题看作一个全局双序列比对问题 ,因而引入生物信息学领域中著名的全局双序列比对算法(NW 算法)。理论及实验研究均表明 ,在绝大多数情况下 ,该方法优于传统方法 ,或至少与传统方法的效果相当。

关键词 语义相似度计算 序列比对 语义知识库

A Novel Approach to Chinese Terms Semantic Similarity Calculation Based on Pairwise Sequence Alignment

Xu Shuo[1] , Zhu Lijun[1] , Qiao Xiaodong[1] and Xue Chunxiang[1,2]
(1. *Institute of Scientific and Technical Information of China* , *Beijing* 100038 ;
2. *Nanjing University of Science and Technology* , *Nanjing* 210094)

Abstract In this study , we first give a problem formulation for Chinese terms semantic similarity calculation . After that , on closer examination , we find that the traditional approach makes an implicit assumption that the order of corresponding primitive terms for two terms must be roughly consistent . In other words , it doesn't consider how the difference in the order affects the quality of correspondence . To overcome this problem , by analogy analysis , we think that this problem can be seen as a global pairwise sequence alignment problem . So a famous global pairwise sequence alignment algorithm , NW algorithm , is introduced from bioinformatics . Finally , an experimental evaluation is conducted , and the result indicates that our approach outperforms or matches at least the traditional one in the majority of cases .

Keywords semantic similarity calculation , sequence alignment , semantic knowledge database

收稿日期 : 2009 年 5 月 13 日

作者简介 : **徐硕** ,男 ,1979 年生 ,博士 ,目前于中国科学技术信息研究所从事博士后科研工作 ,研究方向 :数据挖掘、信息抽取、生物信息等。E-mail : xush@ istic .ac .cn。**朱礼军** ,男 ,1973 年生 ,博士 ,副研究员 ,研究方向 :知识组织、语义检索等。**乔晓东** ,男 ,1965 年生 ,英国谢菲尔德大学硕士 ,研究员 ,研究方向 :信息服务、信息资源管理等。**薛春香** ,女 ,1979 年生 ,博士 ,讲师 ,主要研究方向 :信息智能处理、知识组织系统等。

1) 本研究受“十一五”国家科技支撑计划“知识组织系统的集成及服务研究与实现”(2006BAH03B03)和中国科学技术信息研究所重点工作项目“汉语科技词系统建设与应用工程(新能源汽车领域)”(2008KP01-3-1)资助。

1 引 言

术语语义相似度计算在许多领域都有广泛的应用,例如智能信息检索、信息抽取、文本分类/聚类、词义消歧、基于实例的机器翻译等。针对这一问题,目前已经有许多定量计算方法,这些方法主要分为两类:一类是基于某一语义分类体系来计算[1~8],另一类利用大规模的语料库进行统计[9~12],本文主要考虑第一类计算方法。

语义分类体系通常也称语义知识库(semantic knowledge database, SKD),常用的几种语义知识库包括《同义词词林》[13~14],HowNet[15],WordNet[16]等。然而,任何一部语义知识库都存在一个完备性的问题,而且它的收词粒度通常比较细。也就是说,它不可能收录实际应用中的所有词汇,特别是科技领域中的复合词,从而导致许多术语间的语义相似度无法直接进行计算。在说明如何解决这个问题之前,本文参考文献[7]、[8],首先给出几个定义。称语义知识库中存在的词汇为原子术语(primitive term, PT);语义知识库中不存在,但可由两个或更多原子术语组合而成的词汇称为组合术语(combined term, CT);原子术语与组合术语统称为术语(term)。严格来说,就是给定一部语义知识库 $D=\{PT_1, PT_2, \cdots, PT_K\}$,则 D 中每个元素都是一个原子术语,而符合以下定义的词汇 CT 为组合术语:$CT = PT_{i_1} PT_{i_2} \cdots PT_{i_n}$, $CT \notin D$, $PT_{i_j} \in D$, $j=1, 2, \cdots, n$, $n \leqslant 2$。

对于任意一个组合术语 CT,由于构成它的原子术语的位置是确定的,因此每个组合术语都可以表示为一个有序列表,即

$$CT \equiv \langle PT_{i_1}, PT_{i_2}, \cdots, PT_{i_n} \rangle \quad (1)$$

为一致起见,原子术语 PT 也可类似地表示为 $\langle PT \rangle$。另外,对于术语 $T = \langle PT_{i_1}, PT_{i_2}, \cdots, PT_{i_n} \rangle$ ($n \geqslant 1$),为方便引用相应原子术语的位置信息,定义函数 R 如下:

$$R(T, PT) = \begin{cases} j, \text{if } PT = PT_{i_j} \\ 0, \text{otherwise} \end{cases} \quad (2)$$

现在,术语语义相似度计算问题可正式陈述为:给定一部语义知识库 $D=\{PT_1, PT_2, \cdots, PT_K\}$,对于任意两个术语 $T_1 = < PT_{1,1}, PT_{1,2}, \cdots, PT_{1,m} >$, $T_2 = < PT_{2,1}, PT_{2,2}, \cdots, PT_{2,n} >$, $PT_{1,i} \in D$ ($i=1, 2, \cdots, m$), $PT_{2,j} \in D$ ($j=1, 2, \cdots, n$),计算 T_1 与 T_2 间的语义相似度 $Sim(T_1, T_2)$。如果 T_1 与 T_2 均为原子术语,则 $Sim(T_1, T_2)$可根据前人的研究直接进行计算(见第3节),本文将其称为Ⅰ型问题;否则,通常的做法是首先将 T_1 与 T_2 中的原子术语建立一定的对应关系,然后按照某种准则进行加权求和(见第4节),本文将其称为Ⅱ型问题。

经仔细分析,我们发现解决Ⅱ型问题的传统方法做了一个隐式假设,即假设组成 T_1 与 T_2 的原子术语的顺序大体是一致的。然而,实际应用中许多术语对并不满足这一假设,比如<燃气,汽车>与<汽车,燃气>。而且组合术语的定义并未考虑术语的有效性,这可能导致有效术语与无效术语间的语义相似度很大,从而对某些应用产生不利影响。此处的有效性是指术语是否有明确的意思。如果具有明确的意思,则称为有效术语(如:<汽车,车灯>);否则称为无效术语(如:<车灯,汽车>)。

为解决这些问题,本文提出了一种新的基于双序列比对的新方法。由于本文的重点是Ⅱ型问题的中文术语语义相似度计算,所以为简单起见,本文的语义知识库采用《同义词词林》,不过本文方法同样适用于其他语义知识库。本文其余部分的组织结构如下:第2节简单介绍《同义词词林》;第3节为Ⅰ型问题的语义相似度计算;第4节为Ⅱ型问题的语义相似度计算,包括传统方法以及本文提出的新方法;第5节为实验评价部分;第6节总结全文。

2 《同义词词林》简介

《同义词词林》(简称《词林1》)[13]是梅家驹等人于1983年编纂而成,初衷是希望提供较多的同义词词语,对创作和翻译工作有所帮助。《词林1》共收词53 859条,按照树状的层次结构把所有收录的词条组织到一起,把词汇分成大、中、小三类,大类有12个(用大写英文字母表示),中类有94个(用小写英文字母表示),小类有1428个(用两位十进制整数表示)。每个小类里都有很多词,这些词又根据词义的远近和相关性分成了若干词群(段落)。每个段落中的词语又进一步分成了若干个行,同一行的词语要么词义相同(有的词义十分接近),要么词义有很强的相关性。

由于《词林1》著作时间较为久远,且之后没有更新,所以很多词已经很不常用,成为所谓的罕用

词,而很多新词又没有加入。有鉴于此,哈尔滨工业大学信息检索实验室参照多部电子词典资源,并投入大量的人力和物力,完成了一部《哈工大信息检索研究室同义词词林扩展版》(简称《词林 2》)[14]。它保留了原版中的39 099个高频词,最终的词表包含77 343条词语。《词林 2》保留了《词林 1》的三级分类结构,并且将《词林 1》中小类的段落看作第四级分类(用大写英文字母表示),段落中的行看作第五级分类(用两位十进制整数表示)。这样,《词林 2》就具备了五级分类结构。

对于第五级的分类结果,由于有的行是同义词,有的行是相关词,有的行只有一个词,为了加以区分,《词林 2》增加了第八位标记,分别是"="、"#"、"@"。"="代表"相等"、"同义";"#"代表"不等"、"同类",属于相关词语;"@"代表"自我封闭"、"独立",它在词典中既没有同义词,也没有相关词。这样,《词林 2》中的每个原子术语都可以用一个八位的编码来表示(表 1),当然原子术语与编码之间并不是一一对应的。例如,编码"Aa01A01="与集合{人,士,人物,人士,人氏,人选}中的所有元素对应;而原子术语"人"对应于集合{Aa01A01=,Ab02B01=,Dd17A02=,De01B02=,Dn03A04=}中的所有编码。

表 1 《词林 2》的词语编码表

编码位	1	2	3	4	5	6	7	8
符号举例	B	o	2	1	A	2	6	# \= \@
符号性质	大类	中类	小类		词群	原子词群		
级别	第 1 级	第 2 级	第 3 级		第 4 级	第 5 级		

为方便起见,令函数 $Code(PT)$ 表示原子术语 PT 的所有编码组成的集合,例如,$Code$(人)={Aa01A01=,Ab02B01=,Dd17A02=,De01B02=,Dn03A04=}。并且令小写字母 c 表示这个集合中的元素,即 $c\in Code(PT)$。

3 Ⅰ型问题的语义相似度计算

从信息论的角度来说,两个事物的相似度不仅与其个性有关,而且与其共性有关[17]。基于此,编码 c_1 与 c_2 间的词义相似度可定义为[7,8]:

$$Sim(c_1,c_2)=\frac{2\times Spd(c_1,c_2)}{Dsd(c_1,c_2)+2\times Spd(c_1,c_2)} \tag{3}$$

其中,$Spd(c_1,c_2)$和 $Dsd(c_1,c_2)$分别表示 c_1 与 c_2 的重合度(superposed degree)和相异度(dissimilitude degree)。对于像《词林 2》这样的语义分类树来说,$Spd(c_1,c_2)$表示 c_1 与 c_2 所代表的叶节点共享的路径长度,$Dsd(c_1,c_2)$表示 c_1 与 c_2 所代表叶节点间的最短路径长度。对于《词林 2》,容易验证式(3)可简化为[18]

$$Sim(c_1,c_2)=\frac{Spd(c_1,c_2)}{5} \tag{4}$$

此时,原子术语 PT_1 与 PT_2 间的语义相似度可定义为[6~8]:

$$Sim(PT_1,PT_2)=\max_{c_1\in Code(PT_1)}\max_{c_2\in Code(PT_2)}Sim(c_1,c_2) \tag{5}$$

4 Ⅱ型问题的语义相似度计算

本节考虑Ⅱ型问题的语义相似度计算问题,即给定两个术语 $T_1=<PT_{1,1},PT_{1,2},\cdots,PT_{1,m}>$,$T_2=<PT_{2,1},PT_{2,2},\cdots,PT_{2,n}>$,计算 T_1 与 T_2 间的语义相似度,不失一般性,可令 $m\leqslant n\leqslant 2$。4.1 节对传统方法进行了仔细分析,并指出其潜在的问题,然后 4.2 节提出了一种基于双序列比对的新方法。

4.1 传统方法

通常的作法[6~8]是首先建立组成 T_1 与 T_2 的原子术语间的对应关系,即构建式(6)所示的对应关系集合:

$$CS=\{PT_{1,1}\leftrightarrow PT_{2,j_1},PT_{1,2}\leftrightarrow PT_{2,j_2},\cdots,PT_{1,m}\leftrightarrow PT_{2,j_m}\} \tag{6}$$

构建该集合的伪代码为:

Algorithm 1:构建对应关系集合

输入:两个术语 $T_1=<PT_{1,1},PT_{1,2},\cdots,PT_{1,m}>$,$T_2=<PT_{2,1},PT_{2,2},\cdots,PT_{2,n}>$ 以及一部语义知识库 D。

输出:组成 T_1 与 T_2 的原子术语间的对应关系集合 CS。

1. $CS\leftarrow\phi$;
2. FOR $i=m$ TO 1, STEP=−1

//考虑到中文词汇构成具有"重心后移"的特点,通常按照从后向前的顺序计算。

j$\leftarrow \underset{PT_{2,j}\in T_2}{arg\,max}$Sim(PT$_{1,i}$,PT$_{2,j}$)

2.1;

2.2 $CS\leftarrow CS\cup\{PT_{1,i}\leftrightarrow PT_{2,j}\}$;

2.3 $T_2\leftarrow T_2\leftrightarrow PT_{2,j}$;

END FOR

不同文献在构建对应关系时采用的策略略有不同,本处采用参考文献[8]中的策略。需要注意的是,本处采用了与集合有关的符号表示法,如 $PT_{2,j} \in T_2$ 表示 T_2 包含原子术语 $PT_{2,j}$,$T_2 - PT_{2,j}$ 表示从 T_2 中删除原子术语 $PT_{2,j}$。容易看出,该算法的时间复杂度为 $O(m \times n)$,空间复杂度为 $O(m+n)$。

对应关系集合 CS 构建完成之后,就可按式(7)计算 T_1 与 T_2 间的语义相似度,其中 α 的典型值为0.3,$Sim(PT_1, PT_2)$为原子术语 PT_1 与 PT_2 间的语义相似度。

$$Sim(T_1,T_2) = \alpha \times \left(\frac{1}{n}+\frac{1}{n}\right) \times \sum_{i=1}^{m} Sim(PT_{1,i}, PT_{2,j_i}) + (0.5-\alpha) \times \frac{m}{n} \times \sum_{i=1}^{m} \left\{ \left[\frac{R(T_1, PT_{1,i})}{\sum_{i'=1}^{m} i'} + \frac{R(T_2, PT_{2,j_i})}{\sum_{j'=1}^{n} j'} \right] \times Sim(PT_{1,i}, PT_{2,j_i}) \right\} \quad (7)$$

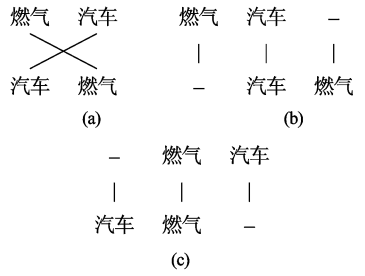

图1 组成 T_1 与 T_2 的原子术语间的对应关系

Example 1:令 T_1=<燃气,汽车>,T_2=<汽车,燃气>。根据 Algorithm1,组成 T_1 与 T_2 的原子术语间的对应关系见图1(a),则 T_1 与 T_2 间的语义相似度为

$$Sim(T_1,T_2) = 0.3 \times \left(\frac{1}{2}+\frac{1}{2}\right) \times 2 + 0.2 + \frac{2}{2} \times \left[\left(\frac{1}{1+2}+\frac{2}{1+2}\right) \times 1 + \left(\frac{2}{1+2}+\frac{1}{1+2}\right) \times 1 \right] = 1.0$$

显然,这是非常不合理的。参考文献[18]也注意到这种现象,并提出了一种基于多层特征的字符串相似度计算模型。经仔细分析,本文发现该方法做了一个隐式假设:组成 T_1 与 T_2 的原子术语的顺序大体一致,换句话说,它并没有考虑顺序的差异对构建对应关系质量的影响。然而,顺序对中文术语是非常重要的,因为中文词汇构成具有"重心后移"的特点,即表达某一具体专指概念的词汇,其中心词往往在词的后半部分。对于 Example1 来说,Fig.1(b)或(c)所示的对应关系应该更合理,其中"–"表示间隔(gap,详见下文),下一小节将考虑如何来构建这样的对应关系。

4.2 基于双序列比对的新方法

在生物信息学中,双序列比对是指将两条DNA、RNA或蛋白质序列排列在一起,标明其相似之处,序列中可以插入间隔,对应的相同或相似的符号排列在同一列上。通过比较两个序列之间的相似区域和保守性位点,寻找二者可能存在的分子进化关系。依据参与比对的是整个序列还是序列片断,可将双序列比对分为全局的和局部的,二者均可通过动态规划(dynamic programming,DP)[19]技术进行求解,本文主要考虑全局双序列比对算法。如需了解有关生物信息更详细的知识,可参考文献[20]。值得注意的是,胡熠等人在自动构建面向信息检索的概念关系时,已经将双序列比对算法成功用于生成相似上下文的模板[21]。

让我们来类比一下,如果把每个原子术语看作一个核苷酸或氨基酸残基,那么每个术语就可以看作一条序列,从而不难发现构建类似于图1(b)或(c)所示的对应关系可以看作寻找两个序列的全局比对,这就是本文方法的主要思想。考虑到中文词汇的构成特点,本文从后向前建立比对,这刚好与生物信息学中的比对过程相反。目前最著名的全局序列比对算法是 Needleman-Wunsch 算法(简称为 NW 算法)[20~22],下面对其做一下简单介绍。

双序列比对通常用打分矩阵 F 进行描述,两条序列分别作为矩阵的两维,它的第 i 行第 j 列元素记为 $F_{i,j}$。对于 T_1 中的每个原子术语,F 中都有一行与其对应;同样对于 T_2 中的每个原子术语,F 中都有一列与其对应。随着算法的运行,$F_{i,j}$ 将被赋值为 T_1 中最后 i 个原子术语与 T_2 中最后 j 个原子术语间最优比对分数。因此,全局序列比对问题就是在矩阵 F 中寻找最佳比对路径。

该算法的主要过程如下:

初始化:

$F_{i,n+1} \leftarrow d((m-i+1)$,$F_{m+1,j} \leftarrow d((n-j+1)$;$i=1,2,\cdots,m+1$;$j=1,2,\cdots,n+1$。

递推公式:

$F_{i,j} \leftarrow \max(F_{i+1,j+1} + Sim(PT_{1,i}, PT_{2,j}), F_{i,j+1} +$

$d, F_{i+1,j}+d)$;

$i=m, m-1, \cdots, 1$; $j=n, n-1, \cdots, 1$。

其中 d 为空位罚分,是为了惩罚一个原子术语与一个间隔比对对比对分数的影响,本文令 $d=-0.05$。一旦矩阵 F 中所有的元素都赋予了值,则 F 中最左上角那个元素($F_{1,1}$)就是最优比对分数。为了揭示最优的比对结果,只需从 $F_{1,1}$ 开始按照如下方式进行比较即可:

Case 1:

IF $F_{i,j}=F_{i+1,j+1}+Sim(PT_{1,i}, PT_{2,j})$,THEN $PT_{1,i}$ 与 $PT_{2,j}$ 比对;

Case 2:

IF $F_{i,j}=F_{i,j+1}+d$,THEN $PT_{2,j}$ 与一个间隔比对;

Case 3:

IF $F_{i,j}=F_{i+1,j}+d$,THEN $PT_{1,i}$ 与一个间隔比对。

当然,如果与 $F_{i,j}$ 相等的情形不只一个,本文按如下优先级顺序进行处理:Case 1>Case 2>Case 3。

实际上,并不需要显式地计算最优比对结果,因为 T_1 与 T_2 间的语义相似度可以在这个过程中进行计算。

Example 2:令 T_1=<可变,气门,正,时,调控,系统>,T_2=<智能,可变,气门,正,时,系统>,T_3=<反射,式,光电,传感器>,T_4=<投射,式,光电,转速,传感器>。图 2 给出了对应的打分矩阵 F,图中的箭头表示每个元素的来源(即 Case 1,Case 2 还是 Case 3),黑色箭头表示最优的比对结果。为清晰起见,比对结果也在图 3 中给出,图 3 也同时给出了 Algorithm1 得到的对应关系。

通过对图 3 中的(a)与(c)以及(b)与(d)的比较,不难发现如果组成两个术语的原子术语的顺序大体一致,则两种算法得到的对应关系相同;否则本文提出的方法更优,这一点在本文的实验部分得到了进一步的验证。

另外,该算法的时间及空间复杂度均为 $O(m\times$

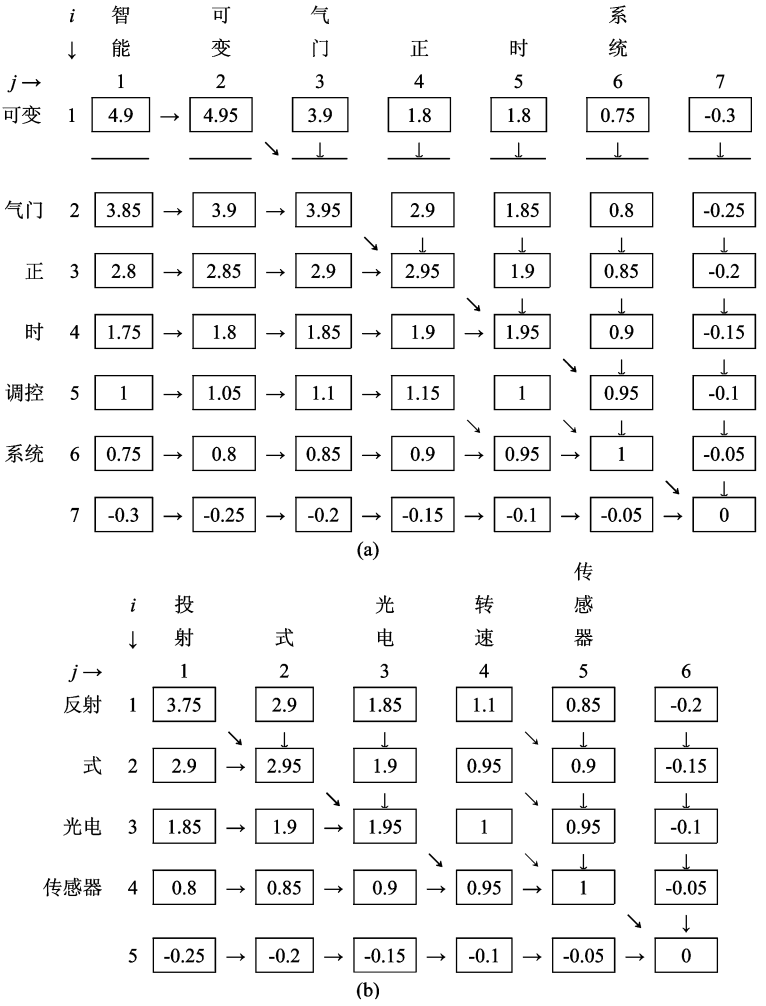

图 2　计算 T_1 与 T_2 间(a)以及 T_3 与 T_4 间(b)最优比对的打分矩阵 F

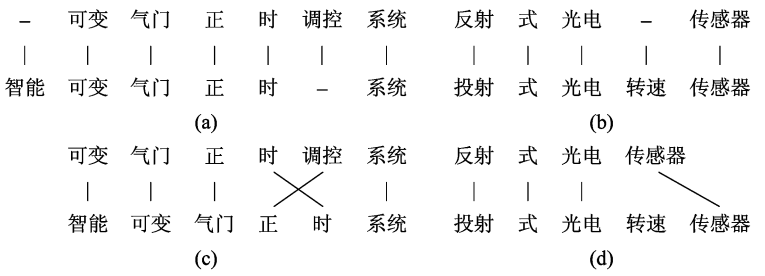

图3 采用NW算法得到的 T_1 与 T_2 间(a)以及 T_3 与 T_4 间(b)的对应关系；以及采用Algorithm1得到的 T_1 与 T_2 间(c)以及 T_3 与 T_4 间(d)的对应关系。

n),也就是说该算法具有二次复杂度。不过可以将空间复杂度从二次改进为线性的,所付出的代价只是稍微增加一点处理时间,大约是原来的两倍,但近似时间复杂度仍为 $O(m\times n)$[22]。然而,本文并未做这种改进,主要是因为组成一个术语的原子术语的个数通常较少,而且时间是本文非常关心的一个因素。

最后,为了计算术语 T_1 与 T_2 间的语义相似度,本文仍然采用式(7),不过与间隔比对的原子术语并不参与计算。

Example 3:现在重新考虑Example1中的术语,即 T_1=<燃气,汽车>,T_2=<汽车,燃气>,并采用图1(c)所示的比对,则 T_1 与 T_2 间的语义相似度为

$$Sim(T_1,T_2)=0.3\times\left(\frac{1}{2}+\frac{1}{2}\right)\times 1+0.2\times\frac{2}{2}\times\left(\frac{1}{1+2}+\frac{2}{1+2}\right)\times 1=0.5$$

这样,如果相似度阈值高于0.5,则不可能得出 T_1 与 T_2 语义相似的结论,然而,对于传统方法,无论阈值设为多少,T_1 与 T_2 语义相似这样的假阳性结论都是不可避免的。

5 实验结果及讨论

目前还没有一个评价术语(尤其是组合术语)语义相似度计算性能的公认标准,本文认为原因主要有两点:①语义相似度是一个相当主观的概念,它不仅因人而异,而且因应用而异;②据我们所知,目前还没有一个公开的与中文术语语义相似度计算有关的标准数据集。因此,表2只列出了我们应用中一些术语间的语义相似度,主要是为下一步研究提供参考及启发。

考虑到实际应用中大部分术语是组合术语,因此事先需要对其切分。为了充分利用《词林2》中的知识,本文以《词林2》作为分词词典。对于分词算法,本文同时采用了前向最大匹配(Forward Maximum Match,FMM)[23]、后向最大匹配(Backward Maximum Match,BMM)[23]以及人工纠正的方式,具体来说就是,如果FMM与BMM的分词结果不一致,从二者之中选择一个更合理的。当然,这仍然会存在一些切分错误,本文对此不作讨论。

从表2中结果(ID=1,2,…,9)不难看出,本文提出的方法可以很好地避免上文提到的问题。对于ID=10的术语对,<车灯,汽车>尽管是一个无效术语,但传统方法却给出了一个极端不合理的相似度,这点与本文引言中的分析也是一致的。然而,由于中文语言现象的复杂性,总是存在一些例外的情形,如表2中ID=11,12,13对应的术语对。这些术语对所表达的意思完全相同,但本文提出的方法并没有给出1.0或接近1.0的相似度。幸运的是,经初步统计,这种情形相对来说比较少;而且注意到这些术语的中心词的位置都是不变的,不同的只是表达非中心意思的原子术语的顺序差异,正因如此,本文提出的方法得到相似度才没有到不可接收的地步。

如果组成术语 T_1 与 T_2 的原子术语的顺序大体一致,则这两种方法几乎给出完全相同的结果,如表2中ID=15,16,…,20,这再次与前文的分析一致。另外,需要说明的是,相似度1.0并不总是意味着两个术语等价,如 T_1=<柴油,发动机>,T_2=<汽油,引擎>(ID=14),主要原因是:在计算原子术语间的语义相似度时,并未考虑相应编码的第八位。

注意到表2中还有一个有趣的现象,如果将阈值设为0.6,可以从中抽取一些具有上下位关系的术语对,如 T_1=<阀>与 T_2=<释放,阀>(ID

表 2　我们应用中一些术语间的语义相似度

ID	T_1	T_2	传统方法	本文方法
1	＜燃气，汽车＞	＜汽车，燃气＞	1.0	0.5
2	＜前轮，驱动＞	＜驱动，轮＞	0.9	0.5
3	＜电阻，式，传感器＞	＜陶瓷，电容器＞	0.65	0.3611
4	＜可变，气门，正，时，调控，系统＞	＜智能，可变，气门，正，时，系统＞	0.4686	0.8429
5	＜直流，电动机，驱动＞	＜驱动，电动机＞	0.7444	0.3833
6	＜点火，控制，计算机＞	＜微机，控制，点火，系统＞	0.765	0.255
7	＜驱动，电动机＞	＜四，轮，驱动＞	0.5144	0.3611
8	＜点火，提前，作用，板＞	＜离心，点火，提前，机构＞	0.038	0.518
9	＜多，片，离合器＞	＜离合器，压，板＞	0.82	0.4867
10	＜汽车，车灯＞	＜车灯，汽车＞	1.0	0.5
11	＜防，抱，死，制动，系统＞	＜制动，防，抱，死，系统＞	1.0	0.8133
12	＜动力，制动，系＞	＜制动，力，系统＞	0.94	0.7
13	＜辅助，制动，系＞	＜制动，辅助，系统＞	1.0	0.7
14	＜柴油，发动机＞	＜汽油，引擎＞	1.0	1.0
15	＜反射，式，光电，传感器＞	＜投射，式，光电，转速，传感器＞	0.785	0.785
16	＜磁，粉，式，安全，联，轴，器＞	＜销钉，式，安全，联，轴，器＞	0.8033	0.8020
17	＜阀＞	＜释放，阀＞	0.6167	0.6167
18	＜释放，阀＞	＜机油，压力，释放，阀＞	0.62	0.62
19	＜阀＞	＜机油，压力，释放，阀＞	0.445	0.445
20	＜多晶硅，薄膜，太阳能，电池＞	＜非，晶，硅，薄膜，太阳能，电池＞	0.7476	0.7476

＝17)，T_2 与 T_3＝＜机油，压力，释放，阀＞(ID＝18)，然而其他一些具有上下位关系的术语对却抽取不出来，如 T_1 与 T_3(ID＝19)。这主要是由式(7)中的权重引起的，这些权重与相应原子术语的个数有关。不过 T_1 与 T_3 间的上下位关系可以由 T_1 与 T_2 以及 T_2 与 T_3 的关系推导出来。

最后，《词林 2》的不完备性也很容易从 Table2 中观察到，如 T_1＝＜多晶硅，薄膜，太阳能，电池＞，T_2＝＜非，晶，硅，薄膜，太阳能，电池＞(ID＝20)。如果《词林 2》中收录了原子术语＜非晶硅＞，则可以想象 T_1 与 T_2 间的语义相似度可能会更高、更合理一些。

6　结束语

本文主要考虑了中文术语语义相似度计算中的 II 型问题，即参与计算的两个术语并不全是原子术语。在对问题进行正式描述之后，仔细分析了传统方法，结果发现它并未考虑组成术语的原子术语的顺序差导对构建对应关系质量的影响。通过类比分析，我们认为构建对应关系的问题可以看作全局双序列比对的问题，从而提出了一种基于双序列比对的新方法，克服了传统方法的缺陷。

参　考　文　献

[1] Agirre E, Rigau G. A Proposal for Word Sense Disambiguation using Conceptual Distance [C] // Current Issues in Linguistic Theory, Proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP), Tzigov Chark, Bulgaria. Amsterdam: John Benjamins Publishing Company. 1995: 258-264.

[2] 刘群，李素建. 基于《知网》的词汇语义相似度计算 [J]. Computational Linguistics and Chinese Language Processing. 2002, 7(2): 59-76.

[3] Chen K-J, You J-M. A Study on Word Similarity using Context Vector Models[J]. Computational Linguistics and Chinese Language Processing, 2002, 7(2): 37-58.

[4] Tran H-M, Dan S. Word Similarity in WordNet[C]// Modeling, Simulation and Optimization of Complex Processes, Proceedings of the 13th International Conference on High Performance Scientific Computing, Hanoi, Vietnam. Berlin: Springer, 2006: 293-302.

[5] Liu X Y, Zhou Y M, Zheng R S. Measuring Semantic Similarity in WordNet[C]//Proceedings of the 6th International Conference on Machine Learning and Cybernetics, Hong Kong, China. Washington: IEEE Computer Society Press, 2007: 3431-3435.

[6] 章成志. 一种基于语义体系的同义词识别研究[J]. 淮阴工学院学报, 2004, 13(1): 59-62, 67.

[7] 夏天. 汉语词语语义相似度计算研究[J]. 计算机工程, 2007, 33(6): 191-194.

[8] 王文荣. 词汇知识系统动态构建方法研究与工具实现[D]. 中国科学技术信息研究所, 2008: 58-75.

[9] 李涓子. 汉语词义排歧方法研究[D]. 清华大学, 1999.

[10] 鲁松. 自然语言中词相关性知识无导获取和均衡分类器的构建[D]. 中国科学院计算技术研究所, 2001.

[11] Dagan I, Marcus S, Markovitch S. Contextual Word Similarity and Estimation from Sparse Data[C]//Proceedings of the Annual Meeting the Association for Computational Linguistics (ACL). NY: Association for Computational Linguistics, 1993: 164-171.

[12] Dagan I, Lee L, Pereira F C N. Similarity-based Models of Word Cooccurrence Probabilities. Machine Learning[J]. Special Issue on Machine Learning and Natural Language, 1999, 34(1-3): 43-69.

[13] 梅家驹, 竺一鸣, 高蕴琦, 等. 同义词词林[M]. 上海: 上海辞书出版社, 1983.

[14] 哈工大信息检索研究室. 哈工大信息检索研究室同义词词林扩展版》[OL]. [2009-04-11]. http://www.ir-lab.org/.

[15] 董振东, 董强. HowNet[OL]. [2009-03-12]. http://www.keenage.com/html/e_index.html.

[16] Miller G A. WordNet[OL]. [2009-04-01]. http://wordnet.princeton.edu/.

[17] Lin D. An Information-Theoretic Definition of Similarity[C]// Proceedings of the 15th International Conference on Machine Learning, CA: Morgan Kaufmann Publishers Inc., 1998: 296-304.

[18] 章成志. 基于多层特征的字符串相似度计算模型[J]. 情报学报, 2005, 24(6): 696-701.

[19] Eddy S R. What is Dynamic Programming? [J] Nature Biotechnology, 2004, 22(7): 909-910.

[20] Setubal J C, Meidanis J. Introduction to Computational Molecular Biology. MA: PWS Publishing, 1997: 47-101.

[21] 胡熠, 陆汝占, 刘慧. 面向信息检索的概念关系自动构建[J]. 中文信息学报, 2007, 21(5): 46-50.

[22] Needleman S B, Wunsch C D. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins[J]. Journal of Molecular Biology, 1970, 48: 443-453.

[23] 陈小荷. 现代汉语自动分析——Visual C++实现[M]. 北京: 北京语言文化大学出版社, 2000: 90-103.

(责任编辑 马 兰)