

文章编号:1674-2974(2016)02-0135-06

中文短文本语法语义相似度算法^{*}

廖志芳^{1†}, 周国恩¹, 李俊锋², 刘飞¹, 蔡飞¹

(1. 中南大学 软件学院, 湖南 长沙 410075; 2. 中南大学 信息科学与工程学院, 湖南 长沙 410075)

摘要:通过分析中文短文本的特征,提出了一种基于语法语义的短文本相似度算法.该算法结合中文语句语义的相似性以及语句语法的相似性,即计算具有相同句法结构的短文本的相似度以及考虑语句词组顺序对相似度的贡献,对中文短文本相似度进行计算.实验表明,本文提出的算法在中文短文本相似度计算结果上更加接近人们的主观判断并且拥有比较好的精确率与召回率.

关键词:语法语义相似度;语句相似性计算;HowNet;语料库;语法分析;语义分析;相似度计算

中图分类号:TP391.1

文献标识码:A

A Chinese Short Text Similarity Algorithm Based on Semantic and Syntax

LIAO Zhi-fang^{1†}, ZHOU Guo-en¹, LI Jun-feng², LIU Fei¹, CAI Fei¹

(1. School of Software, Central South Univ, Changsha, Hunan 410075, China;

2. School of Information Science and Engineering, Central South Univ, Changsha, Hunan 410075, China)

Abstract: A short text algorithm based on semantic and syntax by analyzing the characteristics of Chinese short text was proposed. The algorithm combines semantic similarity with the similarity of sentence syntax in Chinese short text, that is to say, to calculate the similarity of short text with the same syntax structure and consider the contribution with sentence words order to similarity. The experiments show that the proposed algorithm is closer to people's judgment and gets a better precision rate and recall rate in the aspect of Chinese short text similarity calculation results.

Key words: semantic and syntax similarity; sentence similarity calculating; HowNet; corpus; semantic analysis; syntax analysis; similarity measure

短文本相似度计算在文章查重、信息检索、图像检索、智能机器问答、词义消歧和搜索引擎等多个领域有着非常广泛的应用^[1],并在英文处理方面取得了许多实质性的成果,如机器人语音对话系统等.但是在中文信息处理中存在一些困难,例如未登录词识别问题、语法结构复杂和一词多义等^[2].为此本文主要研究中文短文本(语句)相似度计算方法,并提

高计算结果的精确度.

短文本相似度表示的是多个短文本(语句)之间的相似程度,属于一种度量参数,相似度值越高,则表明文本间越相似,反之越不相似^[3].文本相似度包括语义和语法等方面,但是在实际中主要考虑语义层次的相似性,往往忽略了文本的语法结构对文本相似度的重要影响.大量的研究证明短文本的语法

* 收稿日期:2015-01-18

基金项目:国家自然科学基金青年基金资助项目(61202095), National Natural Science Foundation of China(61202095); 湖南省自然科学基金资助项目(12JJ3074)

作者简介:廖志芳(1968-),女,湖南长沙人,中南大学副教授,博士

† 通讯联系人, E-mail: zfliao@csu.edu.cn

结构对相似性的影响是非常重要的^[4],但是当前基于 HowNet^[5]的中文短文本相似度计算大都是分析文本的语义层次的相似性,这样就会导致计算结果的精确度和召回率都不高,为了提高相似度计算的精确度和召回率,本文以 HowNet 为语料库和 Stanford^[6]为语法解析工具,在中文短文本的语义信息基础上加入了文本的语法结构信息来研究短文本相似度计算。

1 语句相似度计算

HowNet 主要揭示了概念间关系及概念的属性间的关系,有关 HowNet 详细介绍可见参考文献[2],在此不赘述。

义原距离定义为两个义原间最短路径上边的数目^[7]。义原距离反映的是两个义原的相对关系,距离越大,则表明两者相似性越低,反之则越高。义原深度定义为两个义原的最近公共父节点到根节点的层次数。义原深度反映的是两个义原在义原层次结构中的绝对关系,深度越小,表明两者越不相似,反之则越相似。

两个义原间的相似度用 $s(p_1, p_2)$ 如文献[2]中的表示方法,即:

$$s(p_1, p_2) = f_1(l) \cdot f_2(h). \quad (1)$$

通过义原间的相似度来计算得到概念间的相似度,概念的相似度实现过程及计算公式的分析如文献[2]所述,在此仅列出其计算公式,用 $s(S_1, S_2)$ 来表示, S_1, S_2 代表两个概念,则:

$$s(S_1, S_2) = \sum_{k=1}^4 \beta_k \prod_{i=1}^{t=k} s_i(S_1, S_2). \quad (2)$$

HowNet 中的词语一般由若干个概念来进行描述,所以一般可以通过概念相似度来计算词语的相似度^[2]。词语相似度可以定义如下:

$$s(W_1, W_2) = \max_{i=1, \dots, m; j=1, \dots, n} s(S_{1i}, S_{2j}). \quad (3)$$

公式(3)的详细分析过程见文献[2],此处不再重复描述。

可根据词语相似度计算来获得语句间的相似度,其详细分析过程详见文献[2],此处仅列出计算公式,如下所示:

$$s(\text{Sen}_1, \text{Sen}_2) =$$

$$\frac{\sum_{i=1}^m \max_{j=1, \dots, n} s(W_{1i}, W_{2j})}{m} + \frac{\sum_{j=1}^n \max_{i=1, \dots, m} s(W_{1i}, W_{2j})}{n} \quad (4)$$

2 基于语法语义的相似度计算

目前基于中文短文本的相似度算法中,大多数研究者着眼于语句的语义层次,然而组成一个语句的不单单是词语本身的语义含义,还包括语法结构,例如一个句子包含不同的句法和词语顺序^[8]等。

最近大量实验证明语句的语法结构对相似度有着重要的影响,不能忽视^[4]。例如,“今天我追赶了一只狗。”和“一只狗今天追赶了我。”这两个语句根据人的主观判断,是两种相反的意思,可是目前绝大多数相似度算法中这两个语句的相似度值是极其高的,甚至是 1,即完全一样,这显然不合适。忽视了语句的语法信息,其计算结果往往缺乏合理性。

本文对中文语句的相似度算法进行改进,在计算相似度的时候,不但考虑语义信息,还将语句的语法结构加进来。本文涉及的语法结构主要从 3 方面来进行考虑:

- 1) 具有相同语法结构的词语集合间的相似度;
- 2) 两个语句中所包含的语法结构的相似性;
- 3) 经过分词后的词组在语句中出现的位置的相似性。

为了解决目前中文领域大多数短文本相似度计算仅仅涉及语句语义特性的缺陷,本文结合语句的语义特性和语法特性,进行短文本的相似度计算。

2.1 语法结构的提取

要对语句的语法结构进行提取与分析,首先需要对语句进行分词。本文使用 Ictclas4j 分词工具,该工具是 Sinboy 在中科院张华平和刘群研制的 Free-ICTCLAS 的基础上完成的一个开源中文分词项目。同时使用 Stanford^[6]工具进行语法解析,该工具是由 Stanford NLP Group 开发的开源工具。

对于某一中文语句,经过语法工具分析后,能够得到某个词语的具体词性,斜杠后面表示的就是该词语的具体词性,如图 1 所示。

[对于/P,今天/NT,下午/NT,的/DEG,考试/NN,,/PU,我/PN,非常/AD,有/VE,信心/NN,,/PU]

图 1 Stanford 分析结果
Fig. 1 Stanford analytical result

但是,仅仅知道词语的具体词性是不够的,具体的词性代表的是词语的特性,并不表示一个语句的结构组成,我们需要从中提取出语句的语法结构,例如简单从句、名词性从句和动词短语等,然后把相应

的词语或短语归类到语法结构集合中。

本文通过分析汉语语法结构的多样性及词语的具体词性,提出一种新的语句语法结构的提取方法,提取过程如下:

1)使用 Ictclas4j 与 Stanford 对语句进行分词处理以及语法解析,得到最基本的词语或短语的词性标注。

2)进一步分析以上获得的结果,通过逐层解析以及句法树分析的方式把已经标注好语法词性的词语或短语进行抽取与归纳,抽取出语句的组成结构类型,例如简单从句类型和动词短语类型等。

3)最后把具有相同语法结构类型的词语或短句存放在相应结构类型的集合中,形成不同的语法结构集合,例如简单从句类型集合、名词性从句类型集合和动词短语类型集合等,实现对语句的语法组成结构的分析。

通过使用上述语句语法结构提取方法进行语句语法处理,提取出语句的语法结构以及相应所包含的词组,如图2所示。

VP:非常	有信心
PP:对于	今天下午的考试
PU: ,	。
NP:我	

图2 本文提取出的语句语法结构

Fig. 2 The grammatical structure of sentence

与图1相比较,我们发现一个中文语句经过语法结构提取后,可以得到该语句的结构组成,例如动词性短语和介词性短语等,相比于图1中单个词语的词性标注,能够得到语句的语法结构组成,更加有利于对中文语句语法语义相似度的研究。因为研究语句的语法结构时,我们更加倾向于研究其整体的结构类型组成,而不是单单某个词语。

中文语句句法结构类型部分见表1。

表1 中文语句句法结构类型分类(节选)
Tab. 1 Classification of syntax structure type

句法结构	含义
ROOT	要处理的文本语句
IP	简单从句
NP	名词性短语
VP	动词性短语
PP	介词性短语
ADV P	副词短语
ADJP	形容词短语

2.2 语句句法结构类型相似度计算

经过语句语法结构提取方法分析后,每个语句都被切分成若干个结构类型,例如有的包含名词性短语和动词性短语等,有的却包括简单从句、名词性短语和副词短语等。

语句语法结构类型相似度计算思想就是计算两个语句中含有相同结构类型的个数与所有结构类型的个数的比值。该比值反映了两个语句在句法结构上的相似性。

定义1 假设语句 Sen1 包含 m 个不同的句法结构类型,分别为 $ST11, ST12, \dots, ST1m$, Sen2 包含 n 个不同的句法结构类型,分别为 $ST21, ST22, \dots, ST2n$, 则句法结构上的相似度 s_{st} 计算公式如下:

$$s_{st} = \frac{\text{sameCount}}{(m+n-\text{sameCount})}. \quad (5)$$

其中 sameCount 表示两个语句中含有相同句法结构类型的个数。

2.3 语句词组位置相似度计算

中文语句中词组的位置对短文本间的相似性有着重要影响,所以需要计算词组位置的相似度^[8]。

本文中采用的方法不是以单个汉字为基本单位,而是以经过分词后的词组为单位,因为单个汉字包含的信息太少,词组能够反映更多的信息,所以以词组为单位计算词组位置的相似度更加合理。

本文计算语句词组位置相似度的方法如下:

1)对语句 Sen1 和 Sen2 分别进行分词,分别得到若干个词组或短语,记为向量 T_1 和 T_2 , 分别含有 s 和 t 个词组,每个位置上的词组分别为 $T11, T12, \dots, T1s$ 和 $T21, T22, \dots, T2t$ 。

2)通过 T_1 和 T_2 , 得到它们的并集 T , 且包含 k 个词组。

3)对于 T 中的每个词组 T_i , 查找在 T_1 中与之相同或含义最相近的词组,预先设定的相似度阈值为 thresholdVale, 记下该词组在 T_1 中的下标 j , 然后构建词组位置向量 R_1 , 令 $R_{1i} = j$; 同理构建向量 R_2 。

4)通过 R_1 和 R_2 计算语句词组位置相似度 s_p 。

定义2 假设语句 Sen1 和 Sen2 经过分词后分别得到若干个词组或短语向量,记为 T_1 和 T_2 , 向量 T 为 T_1 和 T_2 的并集, T_1 和 T_2 对应于 T 的词组位置向量分别为 R_1 和 R_2 。则语句词组位置相似度计算公式见式(6)。

$$s_p = 1 - \sqrt{\frac{\sum_{i=1}^k (R_{1i} - R_{2i})^2}{\sum_{i=1}^k (R_{1i} + R_{2i})^2}} \quad (6)$$

2.4 基于语法结构的语句语义相似度计算

目前绝大多数的中文语句相似度计算方法都是根据第一个语句中的每个词语分别和第二个语句中的每对词语计算相似度,取最大值作为第一个语句中的那个词语对第二个语句的相似度,然后第一个语句中每个词语都这样计算,最后取所有相似度值的均值作为第一个语句对第二个语句的相似度。

上述的方法完全依靠语义信息,而忽视了语法结构信息,计算结果都不太理想,因此本文把语法结构信息加入计算中,即计算具有相同语法结构类型的词组间的相似度,综合了语法和语义两大方面。

定义3 假设两个语句 Sen1 和 Sen2, Sen1 包含 m 个不同的语法结构类型,分别为 ST11, ST12, ..., ST1 m , Sen2 包含 n 个不同的语法结构类型,分别为 ST21, ST22, ..., ST2 n , 其中具有相同语法结构类型的重新记为 SST1, SST2, ..., SST p , 共有 p 个,且 $p \leq \min(m, n)$, 则令第 i ($i \leq p$) 个相同类型结构在第一个语句 Sen1 中表示为 SST1 i , 在第二个语句中 Sen2 表示为 SST2 i , 则对应结构类型的词组集合在两个语句中分别为 WL1 i 和 WL2 i 。

令 $\text{count}(\text{WL1}i)$ 表示 Sen1 中语法结构类型为 SST i 对应的词组集合 WL1 i 中词组的个数;同理 $\text{count}(\text{WL2}i)$ 表示 Sen2 中语法结构类型为 SST i 对应的词组集合 WL2 i 中词组的个数; WL1 iq 和 WL2 iq 分别表示词组集合 WL1 i 和 WL2 i 中第 q 个词组。则 Sen1 和 Sen2 的基于语法结构的语义层次的相似度公式为:

$$s_s = \frac{\sum_{i=1}^p \text{ratio}_i \times S(\text{WL}_{1i}, \text{WL}_{2i})}{\sum_{i=1}^p \text{ratio}_i \times \text{noSameCount}} \quad (7)$$

其中

$$S(\text{WL}_{1i}, \text{WL}_{2i}) = \frac{\left[\sum_{j=1}^{\text{count}(\text{WL}_{1i})} \max_{k=1, \dots, \text{count}(\text{WL}_{2i})} s(\text{WL}_{1ij}, \text{WL}_{2ik}) \right] / \text{count}(\text{WL}_{1i}) + \left[\sum_{j=1}^{\text{count}(\text{WL}_{2i})} \max_{k=1, \dots, \text{count}(\text{WL}_{1i})} s(\text{WL}_{1ik}, \text{WL}_{2ij}) \right] / \text{count}(\text{WL}_{2i})}{2}$$

且 ratio_i 表示的是结构类型 SST i 所对应的权重值,

不同的语法结构对整个语句的含意的贡献度不一样,最终的 ratio_i 权重值也就不一样; noSameCount 表示的是上述两条语句都不具有的句法结构类型的个数,即 $\text{noSameCount} = m + n - p$; pf 表示的是句法结构差异的调节因子。

2.5 基于语法语义的语句相似度计算

基于语法语义的语句相似度计算方法综合考虑了语法和语义两方面特性,主要包括基于语法结构的语句语义相似度计算、语句语法结构类型相似度计算和语句词组位置相似度计算。

通过公式(5),(6)和(7)可以得到语句间的最终计算公式,见式(8)。

$$s(\text{Sen}_1, \text{Sen}_2) = a \times s_s + (1 - a) \times (b \times s_{st} + (1 - b) \times s_p) \quad (8)$$

式中: a 和 b 都是权重因子。

公式(8)表示,中文语句间的相似度最终由语句的语法和语义的相似性共同构成。在计算短文本(语句)间的相似度时,改变了以往仅仅考虑语句语义相似性的思路,本文不但考虑语义相似性,还考虑了语句语法结构对相似性的影响。

本文在计算语义相似度时,加入语法结构信息,即计算具有相同语法结构的词组集合间的相似度,另外充分考虑了句法结构类型和词组位置相似性对整体语句相似度的贡献与影响。最终短文本(语句)间相似度计算更加符合中文语句的特点,计算结果更加合理,与人们的主观判断更为接近。

3 实验及分析

当前基于语法的相似度计算方法多用于英文短文本处理,为进行中文短文本相似度计算,本文采用了以下3种方法来对相似度算法进行分析。

方法一:刘群、李素建等^[9]提出的一种相似度方法,在计算时仅仅考虑义原距离,未考虑深度。

方法二:一种既考虑义原距离和深度,又在计算过程中加入词语词频作为权重的方法^[2],具体参数设置详见参考文献^[2]。

方法三:本文中介绍的基于语法语义的短文本相似度计算方法。

利用上述3种方法分别测试50对中文语句,计算语句(短文本)间的相似度值,部分计算结果见表2。

表 2 三种方法短文本相似度实验对比(节选)
Tab.2 Similarity results of three methods

序号	测试语句对	方法 1	方法 2	方法 3
1	今天我追赶了一只狗. 今天一只狗追赶了我.	1.000	0.889	0.387
2	他为人忠厚老实. 那家伙非常狡猾.	0.740	0.586	0.284
3	不学习怎会有知识? 不好好学习,怎能取得好成绩呢?	0.797	0.761	0.455
4	今天天气非常好. 今天阳光明媚.	0.711	0.641	0.679
5	她穿上这身衣服,显得越发标致. 她长成了一位漂亮的女人.	0.713	0.684	0.765
6	这种药品出自于哪个生产商? 这类药品定价是多少?	0.519	0.467	0.400

3.1 权重因子实验与分析

利用上述实验数据,对方法三的短文本相似度计算公式中的权重因子 a 和 b 进行实验分析,获得能够使得计算公式的效果最佳的权重因子组合.

采用控制变量法对 $a = 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.90, 0.95$ 和 $b = 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$ 进行实验分析,且设置相似度阈值为 0.6, 根据获得的结果进而计算得到不同的精确度和召回率. 不管权重因子 a 和 b 取上述的哪个值,本文提出的相似度计算方法的召回率基本上都是差不多的,稳定在 0.882 上下,这说明了本文方法的稳定性. 根据实验结果绘制本文方法在权重因子 a 和 b 不同取值时的精确度的折线图,如图 3 所示,其中横坐标表示 a 的取值,纵坐标表示精确度, b 的不同取值采用不同的线条表示.

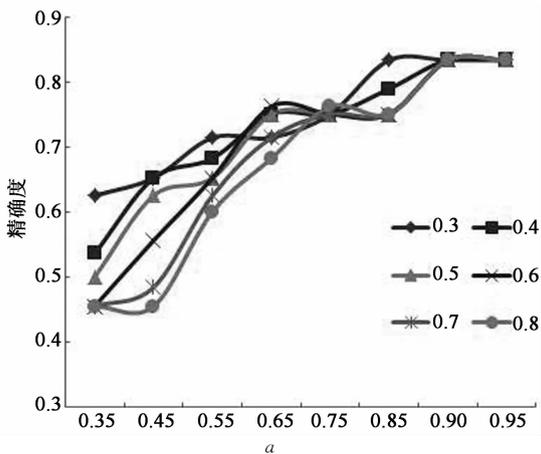


图 3 不同权重因子的精确度比较
Fig.3 The precision rate comparison of different weighting factors

根据图 3 分析可知,当权重因子 a 的值大于 0.6, b

的取值在 $[0.4, 0.6]$ 时,本文方法的精确度基本上都在 0.75 以上,尤其当 a 值在 $(0.85, 0.95]$ 区间时,本方法的精确度可达到最大值 0.833,且比较稳定.

综上所述,可以得到本文最终计算公式中的权重因子 a 和 b 的取值范围. 当 a 值在 $(0.85, 0.95]$ 区间,且 b 的取值在 $[0.4, 0.6]$ 时,本文方法的召回率以及精确度能够同时达到比较高的水平,分别为 0.882 和 0.833.

3.2 短文本相似度实验与分析

方法三中的权重因子 a 和 b 分别取为 0.88 和 0.5, 然后对实验数据进行统计分析,计算 3 种不同计算方法的计算结果的精确率和召回率,且设置相似度阈值为 0.6, 结果如图 4 所示.

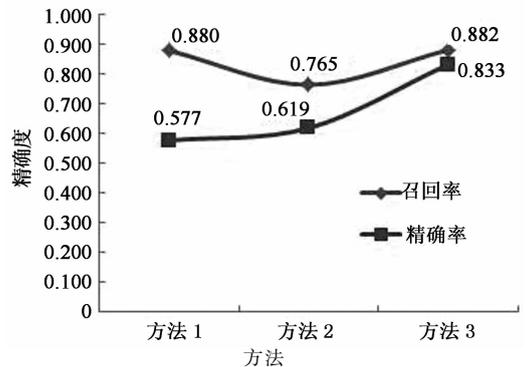


图 4 三种方法计算语句相似度结果评价
Fig.4 Similarity result evaluation of three methods

根据图 4 可知,在召回率基本上比较高的情况下,本文提出的方法的精确率为 0.833,比方法一的 0.577 和方法二的 0.619 都要高,这说明本文方法非常明显地提高了查准率,有效地减少了噪音数据,计算结果更加能够被人们接受.

把实验数据分为近义的语句对集合和反义的语句对集合,分别进一步分析.

对于近义的语句对集合,将相似度区间分为 3 个,图 5 描述了不同方法在每个区间内近义语句对占有所有近义语句对的比例. 方法一中接近一半的语句对的相似度值在 $(0.8, 1]$ 间,其在 $(0.5, 0.8]$ 间的语句对还不到一半,因为语句对没有完全一样的,所以计算结果偏高,且还有一些语句对的相似度低于 0.5, 显然其计算结果不合理. 方法二中大概 82.00% 的语句对的相似度值都在 $(0.5, 0.8]$ 之间,比较符合实际,但是其计算结果中仍然有低于 0.5 的,结果也不太理想. 而方法三中在 $(0.8, 1]$ 之间的不到 25.00%, 绝大部分都是在 $(0.5, 0.8]$ 之间,没有低于 0.5 的,这样的计算结果显然更加合理,更接近人们的主观判断.

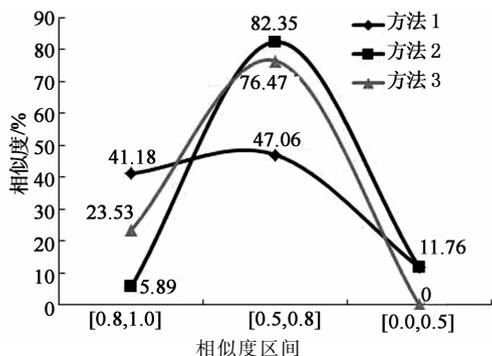


图5 近义语句对间的实验数据对比

Fig. 5 The comparison of similar sentences

对于反义语句对集合,把相似度区间分为4个,图6描述了不同方法分别在每个区间中的测试语句对所占的比例.由图可知,方法一和方法二在相似度大于0.5时大概都有75.00%的反义语句对,而低于0.5的却只有25.00%,显然它们的计算结果都非常不理想,计算粗糙.而方法三在(0.5, 1]区间中只有29.00%左右的反义语句对,且约71.00%的语句相似度都是低于0.5.显然方法三计算结果更加合理和精确.另外由表2可知,有些语句意思明明是相反的,可方法一和方法二计算结果都非常高,甚至是1,而方法三却能够得到非常合理的结果.

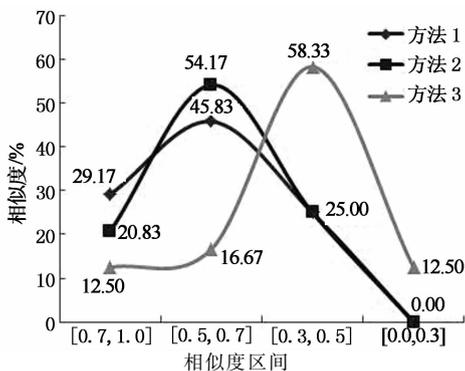


图6 反义语句对间的实验数据对比

Fig. 6 The comparison of antonymous sentences

由上述实验结果可知,本文提出的方法相比于方法一和方法二,具有比较好的查全率,并且其精确率更高,能够非常有效地减少噪音数据的产生,更加接近人们的主观判断.

4 结论

本文以HowNet为词典库,以Stanford为语法解析工具,并在此基础上研究了本文提出的基于语法的中文短文本的相似度计算.

在本文描述的方法中,我们结合语句的语法结构和语义信息计算整个语句的相似度,即计算具有

相同语法结构的词组间的语义相似度以及考虑语法结构类型间相似性和词组位置相似性对整个语句相似度的影响.即使两个语句完全一样,但是语句结构不同或词组位置不同,也会导致意思完全不一样,这样的计算方式与人们的主观判断更加接近,也符合中文语句(短文本)的复杂性特点.

本文研究的方法在一定程度上解决了目前中文领域基于HowNet进行短文本相似度计算的方法中存在的结果不合理现象.通过实验对3种方法进行对比分析,证明了本文描述的中文短文本相似度方法更合理,具有比较好的召回率和精确率.

参考文献

- [1] 蒋溢,丁优,熊安萍,等.一种基于知网的词汇语义相似度改进计算方法[J].重庆邮电大学学报:自然科学版,2009,21(4):533-537. JIANG Yi, DING You, XIONG An-ping, et al. An improved computation method of word's semantic similarity based on HowNet[J]. Journal of Chongqing University of Posts and Telecommunications: Natural Science, 2009,21(4): 533-537. (In Chinese)
- [2] 廖志芳,邱丽霞,谢岳山,等.一种频率增强的语句语义相似度计算[J].湖南大学学报:自然科学版,2013,40(2):82-88. LIAO Zhi-fang, QIU Li-xia, XIE Yue-shan, et al. A frequency enhanced algorithm of sentence semantic similarity[J]. Journal of Hunan University: Natural Sciences, 2013,40(2): 82-88. (In Chinese)
- [3] 李连,朱爱红,苏涛.一种改进的基于向量空间文本相似度算法的研究与实现[J].计算机应用与软件,2012,29(2):282-284. LI Lian, ZHU Ai-hong, SU Tao. Research and implementation of an improved VSM-based text similarity algorithm[J]. Computer Applications and Software, 2012,29(2):282-284. (In Chinese)
- [4] OLIVA J, SERRANO J I, CASTILLO M D, et al. SymSS: a syntax-based measure for short-text semantic similarity[J]. Data & Knowledge Engineering, 2011,70(4):390-405.
- [5] 董振东,董强.关于知网一中文信息结构库[EB/OL]//http://www.keenage.com/html/c_index.html. DONG Zhen-dong, DONG Qiang. Chinese information database based on CNKI[EB/OL]//www.keenage.com/html/C.cndex.html. (In Chinese)
- [6] The Stanford Natural Language Processing Group. The Stanford NLP[EB/OL]//http://nlp.stanford.edu/software/lex-parser.shtml.
- [7] ISLAM A, INKPEN D. Semantic text similarity using corpus-based word similarity and string similarity[R]. Ottawa:University of Ottawa,2008.
- [8] LI Y H, MCLEAN D, BANDAR Z A, et al. Sentence similarity based on semantic nets and corpus statistics[J]. IEEE Transactions on Knowledge and Data Engineering,2006,18(8):1138-1150.
- [9] 刘群,李素建.基于《知网》的词汇语义相似度计算[C]//第三届语义学研讨会论文集.台北:台北中央研究院,2002:149-163. LIU Qun, LI Su-jian. How net-based lexical semantic similarity calculation[C]//Third Semantics Workshop Proceedings. Taipei: Academia Sinica, 2002: 149-16. (In Chinese)