一种基于 WordNet 的短文本语义相似性算法

翟延冬[1],王康平[1,2],张东娜[1],黄 岚[1,2],周春光[1,2]
(1.吉林大学计算机科学与技术学院,吉林长春 130012; 2.吉林大学符号计算与知识工程教育部重点实验室,吉林长春 130012)

摘 要: 短文本语义相似性计算在文献检索、信息抽取、文本挖掘等方面应用日益广泛.本文提出了一种短文本语义相似性计算算法 ST-CW.此算法使用 WordNet 和 Brown 文集来计算文本中的概念相似性,在此基础上提出了一个新的方法综合考虑概念、句法等信息来计算短文本的语义相似性.在 R&B 及 Miller 数据集上进行实验,实验结果验证了算法的有效性.

关键词: 短文本语义相似性; WordNet; 基于文集的方法
中图分类号: TP16 **文献标识码:** A **文章编号:** 0372-2112 (2012)03-0617-04
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.03.35

An Algorithm for Semantic Similarity of Short Text Based on WordNet

ZHAI Yan-dong[1], WANG Kang-ping[1,2], ZHANG Dong-na[1], HUNAG Lan[1,2], ZHOU Chun-guang[1,2]
(1. *College of Computer Science and Technology, Jilin University, Changchun, Jilin* 130012, *China*;
2. *Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin*130012, *China*)

Abstract: The algorithm for semantic similarity of short text is used widely in document retrieval, information extraction and text mining. An algorithm for semantic similarity of short text named ST-CW is presented. The algorithm calculates semantic similarity of concept based on WordNet and The Brown Corpus, and then a formula is presented which refers to both concept similarity and syntactic information in short text. The evaluations are conducted on R&B and Miller dataset.

Key words: semantic similarity of short text; WordNet; corpus-based method

1 引言

随着网络的飞速发展,信息以及数据大量的增加,人类对信息处理的依赖性进一步增强,要求信息处理从词形处理、词义处理向智能化语义处理发展.短文本语义相似性计算是语义应用的基础,可以有效的提高信息搜索效率和准确性,在文本相关性判断、网页检索及分类、信息挖掘、QA 问题研究以及摘要提取等应用领域扮演着越来越重要的角色,同时在短信、微博信息处理上有很好的应用前景[1].在相似性计算方面,目前已有一些针对长文本以及大文档的相似性计算方法[2~5],但是这些算法计算开销较大,对短文本的表达形式不完善,并且其中一些算法需要一定的背景知识和人工输入信息的辅助.

在文献[6]提出的概念相似性计算方法的基础上,本文提出了一种新的计算短文本语义相似性的算法(Short Text based on Corpus and WordNet,简称 ST-CW),该算法不需要任何领域知识.在 R&B 以及 Miller 数据集上进行了实验,实验结果验证了算法的有效性.

2 WordNet 及常见的文本语义相似性计算方法

2.1 知识库 WordNet

WordNet 本体库是 Princeton 大学研发的在线的词汇参照系统.本文采用的是 WordNet V3.0,共有 155287 个独立的词形,117659 个同义词集合,206941 个词形一词义对[7,8].

2.2 常见的语义相似性算法

从上世纪 90 年代开始,基于 WordNet 的语义相似性计算兴盛起来,涌现出大量的语义相似性计算算法,这些研究成果主要针对长文本计算,短文本语义相似性计算的研究相对较少[9].已有的相似计算主要有以下几

收稿日期:2011-04-19; 修回日期:2011-10-12
国家自然科学基金项目资助(No.61175023,No.60903097)

个缺点:第一,文本被表示成的高维空间,维度可能达到几百甚至是上千维[7,10],文本向量构成稀疏向量矩阵,计算效率较低,高维或是高度稀疏的向量会导致语义相似性计算的不可接受;第二,有的方法要求对相似性计算的句子进行复杂的人工预处理;第三,有些方法是领域相关的,局限性比较大,不能用于其他领域.为了克服上述缺点,本文提出了一种新的语义相似性计算方法 ST-CW,算法只关注句子中的相似性高的概念从而降低维数,同时不需要对句子进行人工的预处理,不使用领域相关知识,实现了语义相似性计算的领域无关性.

3 ST-CW 算法

3.1 概念相似性的计算

概念相似性的计算是短文本语义相似性计算的基础,现有的大部分概念相似性算法都以 WordNet 为基础[15].文献[6]提出 SS-CW 算法使用 WordNet 为基础,综合 Brown 文集来计算概念的相似性,取得了较好的效果.SS-CW 算法在使用主要考虑 WordNet 中概念之间的层次关系,并使用 Brown 文集中概念出现的频率来分配权重,实验结果证明了算法的有效性.因此本文采用此算法来计算短文本中概念的相似性.

3.2 相似概念词序相似性计算

句法信息(Syntactic Information)即概念的顺序相似性对短文本的语义相似性有一定贡献,但比重不应该很大[11],一般要小于 0.5.

一个具有 m 个概念的句子 $\boldsymbol{P}$ 与一个具有 n 个概念的句子 $\boldsymbol{R}$,经过预处理去掉停用词与标点符号后,表示为向量形式 $\boldsymbol{P}=\{p_1,p_2,\cdots p_m\}$,$\boldsymbol{R}=\{r_1,r_2,\cdots r_n\}$.利用 SS-CW[6]方法计算概念之间的相似性.概念之间的相似性匹配的阈值为 0.85,如果两个句子中的某两个概念的相似性超过这个值,则认为这两个概念匹配,则将其从 $\boldsymbol{P}$ 和 $\boldsymbol{R}$ 中删除,同时把这两个概念分别加入列表 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$,直到 $\boldsymbol{P}$、$\boldsymbol{R}$ 中所有概念的相似性都小于指定阈值.将列表 $\boldsymbol{X}$、$\boldsymbol{Y}$ 中概念的个数记为 δ,将 $\boldsymbol{X}$、$\boldsymbol{Y}$ 中的元素按照它们在 $\boldsymbol{P}$、$\boldsymbol{R}$ 中出现的顺序分别排列.使用公式(1)计算顺序相似性.其中 x_i,y_i 表示相似概念在 $\boldsymbol{X},\boldsymbol{Y}$ 中的位置,n 表示任意正整数.

$$S=\begin{cases}0 & \delta=0\\ 1-\dfrac{2\times\sum_{i=1}^{\delta}|x_i-y_i|}{\delta^2}\delta=2n\\ 1-\dfrac{2\times\sum_{i=1}^{\delta}|x_i-y_i|}{\delta^2-1}\delta=2n+1\end{cases}\quad(1)$$

如果 $\boldsymbol{X}$、$\boldsymbol{Y}$ 中对应的概念位置相同,那么 S 等于 1.从公式中容易得出,$\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 中的概念的顺序越相似,S 的值越大.下面举一个例子说明计算的过程:

短文本如下所示:

P:Many consider Malin as the best player in Ping-Pong history.

R:Malin is one of the best Ping-Pong players.

经过相似性计算后得到的相似概念序列为 $\boldsymbol{X}=\{$Malin,best,player,Ping-Pong$\}$,$\boldsymbol{Y}=\{$Malin,best,Ping-Pong,player$\}$,代入公式中可得 $S=1-\dfrac{2\times(|4-3|+|3-4|)}{16}$,顺序相似性的值为 0.75.

3.3 ST-CW 算法流程

在计算短文本相似性时,两个短文本中的概念被分为 3 组,分别是匹配的概念、类似的概念和无关的概念.匹配的概念是指含义基本一致的概念,这些概念的相似性一般都大于 0.85.类似概念的相似程度要小于匹配的概念.无关的概念是文本中除去匹配概念和类似概念后的概念.它们的相似程度最低.两个短文本的相似性由它们的匹配概念和类似概念决定.在短文本相似性的计算中,匹配概念的重要程度最高,通过这种划分,可以对不同种类的概念对使用不同的方法,减少参与结算的概念,从而提交算法效率.

具体的步骤如下:

Step1 短文本预处理.首先对字符串进行分词,然后去除停用词和特殊符号,并对单词进行词根还原,使用 WordNet 中 Morph 接口来实现.经过预处理的短文本用向量 $\boldsymbol{P}=\{p_1,p_2,\cdots p_m\}$,$\boldsymbol{R}=\{r_1,r_2,\cdots r_n\}$ 表示,其中 $m\geqslant n$,如果不是,则交换 $\boldsymbol{P}$ 和 $\boldsymbol{R}$.

Step2 匹配概念相似性计算.利用 SS-CW 相似性计算方法计算概念之间的相似性,设置概念之间的相似性匹配的最小值阈值为 0.85,将概念匹配的个数记为 δ,去掉所有匹配过的单词之后将向量 $\boldsymbol{P}$ 和 $\boldsymbol{R}$ 记做 $\boldsymbol{P}'=\{p_1,p_2,\cdots,p_{m-\delta}\}$,$\boldsymbol{R}'=\{r_1,r_2,\cdots,r_{n-\delta}\}$.使用 3.2 中的方法计算匹配概念之间的顺序相似性 S_0.如果所有的概念均匹配,即 $m=n=\delta$,则转到 Step 4.

Step3 类似概念相似性计算.采用概念相似性计算方法 SS-CW,计算除去已匹配概念之后的 $\boldsymbol{P}'$、$\boldsymbol{R}'$ 之间的相似性,相似性矩阵 $\boldsymbol{M}$ 如下所示.

$$\boldsymbol{M}=\begin{pmatrix}\gamma_{11} & \gamma_{12} & \cdots & \gamma_{1j} & \cdots & \gamma_{1(n-\delta)}\\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2j} & \cdots & \gamma_{2(n-\delta)}\\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots\\ \gamma_{i1} & \gamma_{i2} & \cdots & \gamma_{ij} & \cdots & \gamma_{i(n-\delta)}\\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots\\ \gamma_{(m-\delta)1} & \gamma_{(m-\delta)2} & \cdots & \gamma_{(m-\delta)j} & \cdots & \gamma_{(m-\delta)(n-\delta)}\end{pmatrix}$$

选出矩阵中最大的元素 γ_{ij},将其对应的概念分别加入到向量 $\boldsymbol{X}'$、$\boldsymbol{Y}'$,将 i 行和 j 列从 $\boldsymbol{M}$ 中删除,重复这一过

程直到 $\boldsymbol{M}$ 中所有元素均小于一个指定的阈值 0.14,或者 $\boldsymbol{M}$ 为空.利用 3.2 中的方法计算 $\boldsymbol{X}'$、$\boldsymbol{Y}'$ 的顺序相似性 S_1.

Step4　采用式(2)计算两个短文本之间的相似性

$$S(P,R)=\frac{(m+n)}{2mn}\times(\delta(1-w_f(1-S_0))+\sum_{i\in X',j\in Y'}\gamma_{ij}(1-w_f(1-S_1))) \quad (2)$$

其中,w_f 代表的是词序相似性对语义相似性的重要程度,它的值一般在 0.5 以下.γ 表示类似概念对的概念相似性,从 $\boldsymbol{M}$ 中获得.公式分为两个部分,公式的前一部分体现了两个短文本长度因素,第二部分体现了概念相似性和词序相似性的影响,其中分别计算了匹配概念和类似概念的贡献.如果两个短文本完全相同,则相似性为 1.

4　实验分析

本文使用常用的数据集 Rubenstein and Goodenough (R&G)[12].在本文中使用其中 30 对,这 30 对数据分布较为平均,有很好的代表性.在实验时比较的部分结果由文献[12]给出,同时还和 PMI-IR[13]算法进行了比较.

表 1　ST-CW 与其他语义相似性算法的比较

R&G	R&G 概念对	人工判断	Li 方法结果	ST 方法结果	ST-CW 结果
1	Cord Smile	0.01	0.33	0.06	0.04
5	Autograph Shore	0.01	0.29	0.11	0.08
9	Asylum Fruit	0.01	0.29	0.11	0.12
13	Boy Rooster	0.11	0.53	0.16	0.12
17	Coast Forest	0.13	0.36	0.26	0.20
21	Boy Sage	0.04	0.51	0.16	0.18
25	Forest Graveyard	0.07	0.55	0.33	0.24
29	Bird Woodland	0.01	0.33	0.12	0.11
33	Hill Woodland	0.15	0.59	0.29	0.19
37	Magician Oracle	0.13	0.44	0.20	0.16
41	Oracle Sage	0.28	0.43	0.09	0.24
47	Furnace Stove	0.35	0.72	0.30	0.39
48	Magician Wizard	0.36	0.65	0.34	0.40
49	Hill Mound	0.29	0.74	0.15	0.16
50	Cord String	0.47	0.68	0.49	0.52
51	Glass Tumbler	0.14	0.65	0.34	0.23
52	Grin Smile	0.49	0.49	0.32	0.38
53	Serf Slave	0.48	0.39	0.44	0.43
54	Journey Voyage	0.36	0.52	0.41	0.43
55	Autograph Signature	0.41	0.55	0.19	0.18
56	Coast Shore	0.59	0.76	0.47	0.53
57	Forest Woodland	0.63	0.70	0.26	0.64
58	Implement Tool	0.59	0.75	0.51	0.56
59	Cock Rooster	0.86	1	0.94	0.92
60	Boy Lad	0.58	0.66	0.60	0.62
61	Cushion Pillow	0.52	0.66	0.60	0.59
62	Cemetery Graveyard	0.77	0.73	0.51	0.73
63	Automobile Car	0.56	0.64	0.52	0.50
64	Midday Noon	0.96	1	0.93	1
65	Gem Jewel	0.65	0.83	0.65	0.64

在计算中,参数设定都是通过一定的实验得出来的结果.反映词汇次序权重的 w_f 取 0.3.前文中提到,此值应该小于 0.5,经过多次实验后,我们发现此值在 0.3 左右时算法有较高的性能.划分匹配概念和类似概念的阈值分别是 0.85 和 0.14,也就是当概念之间的相似性大于 0.85 时,两个概念可以定义为匹配概念,当两个概念之间的相似性小于 0.85 大于 0.14 时被认为是类似概念.这两个阈值的选择也是通过多次实验后选取的经验值.实验结果如表 1 所示.

根据上面的结果可以看出,在 Miller 数据集上本文提出的算法更加符合人的判断.在概念相似性较小时,与 ST 算法基本持平,但是对于概念之间相似性较大的概念对,新的方法具有更加有效的判断能力.

进一步在全部 R&B 数据集测试了本文提出的算法,为了比较两种方法的优劣,参照文献[16],用公式(3)计算计算实验结果和人工判断的相关性:

$$\rho_{xy}=\frac{\mathrm{Cov}(x,y)}{\sqrt{D(x)D(y)}} \quad (3)$$

其中 $\mathrm{Cov}(x,y)=E((x-E(x))(y-E(y))$.符号 $D(x)$,$D(y)$表示方差,ρ_{xy}越大,表示相关性越高.

图 1 中纵坐标表示算法结果和人工结果的相关度.可以清晰的看到在两个数据集上 ST-CW 均有较高的相关系数,也就是与人工判断更加接近.结果如图 1 所示.

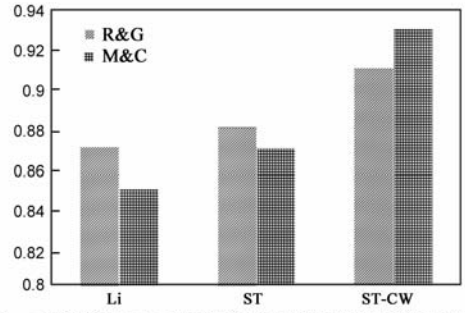

图1　三种算法与人工测得语义相似性值的相关系数比较

5　结论

本文提出了一种新的短文本间的语义相似性算法 ST-CW,这个算法综合考虑了概念的相似性信息和主要概念在句子中的顺序信息.但是,此方法没有考虑到自然语言中的复杂的语境环境、上下文等很重要的信息,在某些场景可能不够准确.同时参数的选择基本依赖经验,需要进一步研究理论方法来指导参数的选择.短文本的相似性计算在不同的应用场景有不同的要求,本文提出的算法提供了一个通用的快速有效的实现途径,具有一定的应用价值和学术价值.

参考文献

[1] 杨震,范科峰,雷建军,郭军 . 基于语义的文本流形研究[J].电子学报,2009,37(3):557 - 561.
YANG Zhen, FAN Ke-feng, LEI Jian-jun, GUO Jun. Text man-

ifold based on semantic analysis[J]. Acta Electronica Sinica, 2009,37(3):557-561.(in Chinese)
[2] 周子力,基于 WordNet 的本体构建及其在安全领域应用关键技术研究[D].上海:华东师范大学,2009.
[3] T K Landauer, D Laham, B Rehder, M E Schreiner. How well can passage meaning be derived without using word order? A comparison of latent semantic analysis and humans[A]. Proc 19th Ann Meeting of the Cognitive Science Soc[C]. Mawhwah, NJ: Lawrence Erlbaum, 1997.412 – 417.
[4] Jiang, Jay J, David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy[A]. Proceedings of International Conference on Research in Computational Linguistics[C]. Taiwan: IEEE, 1997.19 – 33.
[5] C Burgess, K Livesay, K Lund. Explorations in context space: words, sentences, discourse[J]. Discourse Processes, 1998, 25(2 – 3):211 – 257.
[6] 张东娜,周春光,刘彦斌,郭东伟 一种基于 WordNet 和 Corpus Statistics 的语义相似性计算方法[J].吉林大学学报(理学版),2010,48(05):811 – 816.
ZHANG Dong-na, ZHOU Chun-guang, LIU Yan-bin, GUO Dong-wei. A semantic similarity computing approach based on WordNet and Corpus statistics[J]. Journal of Jilin University (Science Edition), 2010, 48(05):811 – 816.(in Chinese)
[7] E K Park, D Y Ra, M G Jang. Techniques for improving web retrieval effectiveness[J]. Information Processing and Management, 2005, 41(5):1207 – 1223.
[8] WordNet Documentation[EB/OL]. http://wordnet.princeton.edu/wordnet/documentation/, October 27, 2010.
[9] Li Y, Mclean D, Bandar Z, O'Shea J, Crockett K. Sentence similarity based on semantic nets and corpus statistics[J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(8):1138 – 1149.
[10] Madhavan J, Bernstein P, Doan A, Halevy A. Corpus-based schema matching[A]. Proceedings of the International Conference on Data Engineering[C]. Tokyo: IEEE Computer Society, 2005.57 – 68.
[11] G A Miller. WordNet: a lexical database for english [J]. Comm ACM, 1995, 38(11):39 – 41.
[12] G Salton, C S Yang. A veetor space model for antomatic indexing[J]. Communications of the ACM, 1975, 18: 613 – 620.
[13] Alexander Budanitsky, Graeme Hirst. Evaluating WordNet-based measures of lexical semantic relatedness[J]. Computational Linguistics, 2006, 32(1):13 – 47.
[14] Turney P. Mining the web for synonyms: PMI-IR versus LSA on TOEFL[A]. Proceedings of the Twelfth European Conference on Machine Learning[C]. Berlin: Springer-Verlag, 2001. 491 – 502.
[15] Budanitsky A, Hirst G. Evaluating wordnet-based measures of lexical semantic relatedness[J]. Computational Linguistics, 2006, 32(1):13 – 47.
[16] Nuno Seco, Tony Veale, Jer Hayes. An intrinsic information content metric for semantic similarity in WordNet[A]. ECAI [C]. Valencia, Spain: Elsevier, 2004.1089 – 1090.

作者简介

翟延冬 男,1968 年 10 月出生于吉林省长春市.现为吉林大学计算机科学与技术学院博士生,从事计算智能和语义网的研究.

王康平(通信作者) 男,1980 年 10 月出生于湖南省邵阳市.现为吉林大学计算机科学与技术学院教师,主要从事进化计算和语义网研究.
E-mail: wangkp@jlu.edu.cn