计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2017.20160578
54(11):2576-2585,2017

一种基于大规模知识库的语义相似性计算方法

张立波[1] 孙一涵[2] 罗铁坚[2]
[1](中国科学院软件研究所 北京 100190)
[2](中国科学院大学 北京 101408)
(zsmj@hotmail.com)

Calculate Semantic Similarity Based on Large Scale Knowledge Repository

Zhang Libo[1], Sun Yihan[2], and Luo Tiejian[2]
[1](*Institute of Software, Chinese Academy of Sciences, Beijing* 100190)
[2](*University of Chinese Academy of Sciences, Beijing* 101408)

Abstract With the continuous growth of the total of human knowledge, semantic analysis on the basis of the structured big data generated by human is becoming more and more important in the application of the fields such as recommended system and information retrieval. It is a key problem to calculate semantic similarity in these fields. Previous studies acquired certain breakthrough by applying large scale knowledge repository, which was represented by Wikipedia, but the path in Wikipedia didn't be fully utilized. In this paper, we summarize and analyze the previous algorithms for evaluating semantic similarity based on Wikipedia. On this foundation, a bilateral shortest paths algorithm is provided, which can evaluate the similarity between words and texts on the basis of the way human beings think, so that it can take full advantage of the path information in the knowledge repository. We extract the hyperlink structure among nodes, whose granularity is finer than that of articles form Wikipedia, then verify the universal connectivity among Wikipedia and evaluate the average shortest path between any two articles. Besides, the presented algorithm evaluates word similarity and text similarity based on the public dataset respectively, and the result indicates the great effect obtained from our algorithm. In the end of the paper, the advantages and disadvantages of proposed algorithm are summed up, and the way to improve follow-up study is proposed.

Key words large scale knowledge repository; semantic similarity; Wikipedia; shortest path; connectivity

摘 要 人类知识总量不断增加,依靠人类产生的结构化大数据进行语义分析在推荐系统和信息检索等领域都有着重要的应用.在这些领域中,首要解决的问题是语义相似性计算,之前的研究通过运用以维基百科为代表的大规模知识库取得了一定突破,但是其中的路径并没有被充分利用.研究基于人类思考方式的双向最短路径算法进行单词和文本的相似性评估,以充分利用知识库中的路径信息.提出的算法通过在维基百科中抽取出颗粒度比词条更细密的节点之间的超链接关系,并首次验证了维基百科之间的普遍连通性,并对2个词条之间的平均最短路径长度进行评估.最后,在公开数据集上进行的实验结果显示,算法在单词相似度得分上明显优于现有算法,在文本相似度的得分上趋于先进水平.

收稿日期:2016-08-10;**修回日期**:2017-02-20
基金项目:中国科学院系统优化基金项目(Y42901VED2,Y42901VEB1,Y42901VEB2)
This work was supported by the Foundation of System Optimization in Chinese Academy of Sciences (Y42901VED2, Y42901VEB1, Y42901VEB2).

关键词 大规模知识库;语义相似性;维基百科;最短距离;连通性

中图法分类号 TP391

利用计算机进行语义相似度评估是人工智能领域的一个瓶颈问题,这个问题的核心是如何让计算机模拟人类的思考方式.然而人类思维模式复杂,难以模仿,但一个基本事实是,人类的思考模式是基于自身的经历和经验所形成的知识来进行推理、分析和决断.因此,利用大规模的知识库赋予计算机一种类似人类的知识结构模型,是一个根本的解决途径.人类根据自身掌握的知识进行联想,对2个单词、段落或文本之间的相似性进行判断,本文通过提取维基百科中的双向最短路径来模拟人类的联想机制进行相似性评估.

语义相似度在自然语言处理和信息检索领域中一直都有着重要的应用,例如问答系统、拼写检查、文本翻译等领域[1].传统方法主要基于单词重叠度来评估语义相关性,这在论文查重系统中有着广泛应用,但如果使用不同的单词表示同样的含义时,这类方法明显无法适用,它的缺陷在于简单地将单词作为评估对象,而没有考虑单词之间的内在联系[2].概率潜在语义分析等方法利用了文章中的单词和主题的关系,但是没有考虑到单词在知识结构中的关系[3].近些年来,一些新的算法考虑到可以利用包含人类知识结构的大规模知识库,例如百科全书和语料库进行语义相似性评估[4],具体来说,它们利用百科中2个词条页面中包含的文本和链接捕获真实世界的知识结构,进而评估语义相似度.

在人类编撰形成的知识库中,大英百科全书和维基百科全书最具有代表性,它们经过人工的梳理和编辑,构成以词条为单位的人类知识宝库.相比大英百科,维基百科在保证了同等正确率的同时,包含了更多的词条量.我们将维基百科词条中含有链接的单词称为节点,节点之间的最短路径能够反映出人类联想机制,为了验证这一点,我们提出了一种基于最短路径对语义相关性进行评估的算法.与之前的算法不同,本文第1次对维基百科中词条之间的连通性进行验证,证明了维基百科中的词条具有普遍联系,2个节点之间一定存在一条通路,即从一个概念出发,经过有限次数的跳转链接可以到达另一个任意节点.在相关的研究论文中,研究者假设任意2个词条之间都可以建立路径关系,但是缺乏证明;最新的一些论文使用随机游走算法寻找路径,而我们使用了最短路径.传统的有关最短路径的算法是基于词条所属分类间的最短路径,而我们使用了更细密的节点;大部分对维基进行的研究都将其限定在100万节点左右,而我们使用了1200万节点进行更加细致的探究.本文提出了一种基于维基百科节点的最短路径进行语义相似性分析的算法,并验证了通过最短路径能够模拟人类的联想机制对语义相似性进行评估.

1 相关工作

传统语义关系的分析需要依靠人类自身知识和经验的背景进行推理和联想,而这对计算机一直是难以克服的障碍.为了探索2个词条之间的关系,计算机必须获取大量相关领域的背景知识,而之前的相关工作更多地依靠统计而忽略了背景知识;另外一些基于背景知识进行语义相关的研究由于知识总量的限制无法取得很好的效果;一些论文基于网页链接对语义进行分析,取得了一定的突破,但是由于网页中的知识颗粒度较粗,并且知识含量较为稀疏,因此不能够反映出某个知识点的全貌.2005年,Giles[5]指出,维基百科的广度和深度都值得信任,从此之后,开启了基于维基百科进行语义分析的热潮.

本文系统地收集了在维基百科的基础上提出的利用人类知识库进行语义分析的方法,如表1所示.传统的方法主要有2类:1)基于内容的方法;2)基于链接的方法.Gabrilovich等人[6]提出一种名为显示语言分析的方法,这种方法通过维基百科收集概念,将文本中的内容与收集的概念建立矩阵关系,但缺乏对人类知识体系的使用.Zesch等人[7]证明了使用维基百科能够提高之前方法的效果,但性能的提升受到制约.Hassan等人[8]衡量2个概念之间的跨越关系,并通过显示语言分析来建立概念的矢量表述.Syed等人[9]将维基百科词条作为本体,使用3种方式将关键词、主题与文本结合,但是缺乏对词条路径关系的考虑.Coursey等人[10]建立了概念和分类关系图,提出了一个自动化的主题分类方法,但是无法解决分类数量有限和概念从属多种分类的问题.Gabrilovich等人[11]通过建立一个语义转换器对概念建立权重矢量,并通过矢量之间的比较进行语义相关性评估,这种方法仅基于文本内容,在概念之间的关系上只考虑了所属分类之间的关系,而忽视了超链接形成的路径.

Table 1 Comparison of Our Method with Previous Work

表 1 相关工作的总结与比较

Referenec	Purpose	Method	Wikipedia Version	Concepts Number	Dataset	Evaluation
Ref[6]	Word Similarity	Explicit Semantic Analysis	Mar 2006 (2.9 Gb)	1 187 839	WS-353, Lee	Lack of the utilization of the network relationship in Wikipedia.
Ref[7]	Word Similarity	Concept Vectors	Feb 2007	682 982	WS-353, M&C, R&G	Improve the effect of the previous methods by making use of Wikipedia; Limit the improvement of performance.
Ref[8]	Cross-lingual Relatedness	Concept Vectors, ESA	Oct 2008	2 221 980	WS-353, M&C	Measure the cross relationship between two concepts; Establish the vector expression of concepts.
Ref[9]	Document Classification	Cosine Similarity	Nov 2006	2 557 939	Selected Documents	Take the titles in Wikipedia as the ontology; Combine the keywords and themes with the text in three ways; Lack of the consideration of the relationship between the paths of the titles.
Ref[10]	Document Classification	Concept and Category Graph	Jul 2008	2 750 000	150 Articles	Establish the relational graph of concepts and classifications; Propose an automated classification method; Unable to solve the following problems: the number of classification is limited, and one concept may belongs to several classifications.
Ref[11]	Document Clustering	ESA, Cosine Similarity	Nov 2005 (1.8 Gb)	910 989	Short Documents	Only based on the text content; For the relationship between the concepts, only consider the relationship between the classifications they belong to, while ignoring the path of hyperlinks.
Ref[12]	Word Similarity	Shortest Categories Path	Feb 2006	971 518	WS-353, M&C, R&G	Evaluate the correlation of two concepts by the path length of two classifications, while all the concepts mapped to the classification they belong to; The path of the relationship between classifications is just a simple tree structure.
Ref[13]	Word Similarity	Hyperlinks Similarity	Nov 2007	2 000 000	WS-353, M&C, R&G	Evaluate the correlation by calculating the repetitive rate of the links contained in titles; Unable to do precise calculation of the relationship between the concepts.
Ref[14]	Word and Doc Similarity	Personalized PageRank	N/A	1 000 000	WS-353, M&C, Lee	Only based on the relationship of the classifications or the overlaps of the hyperlinks contained in two particular titles.
Ref[4]	Word and Doc Similarity	Visiting Probability	Jun 2006	1 264 611	WS-353, M&C, Lee	Unable to accurately simulate the human thinking pattern because the people who take part in the test are under the time pressure due to the special nature of the game.
Ref[15]	Word Similarity	Random Walk	Feb 2015	N/A	WS-353	Simulate human's behavior of clicking on hyperlinks by applying the random walk algorithm.
Our Method	Word and Doc Similarity	Shortest Link Path	Jun 2016 (53.4 Gb)	12 269 222	WS-353, M&C, Lee	Make full use of the network relationship of the titles in Wikipedia; Introduce the concept named bidirectional distance.

另外一些研究,利用路径信息进行语义相似度评估,例如 Ponzetto 等人[12]将概念映射到所属的分类中,通过 2 个分类之间的路径长度对 2 个概念的相关性进行评估,但是分类之间的路径关系仅仅是简单的树形结构. Milne 等人[13-16]通过计算词条所包含链接的重复率进行相关性评估,这种方法无法对概念之间的关系进行精确计算. 与 Yeh 等人[14]的方法类似,有研究者提出了一种基于维基漫步方法,Yazdani 等人[4]提出一种访问概率的方法来计算节点之间的距离,这种方法仅仅基于概念所属分类的路径关系,或者是特定 2 个词条包含超链接的重叠性. 刘均等人[17]设计了一种利用维基百科链接关系的 13 维特征适量来进行关系抽取,但是缺乏对节点之间通路的考虑. Singer 等人[18]使用了一种维基游戏,通过人们的点击来寻找 2 个词条之间的路径关系,然而由于游戏的特殊性质,会对参与测试的人们造成时间压力,因此无法准确模拟人类的思考模式. Niebler 等人[19]注意到了无约束航行数据,维基百

科点击流数据[20]记录了人们在一个维基词条中如何进行下一次链接点击. Dallmann 等人[15]基于这个数据集,使用随机游走的算法模拟了人类的超链接点击行为,但是随机游走往往无法获取最佳路径,而最短路径才对人类的决策最有意义.

与传统方法中利用词条中的链接关系不同,我们通过寻找 2 个节点之间的链接通路,来描述它们之间的关系. 本文提出一种新的方法,通过结构化的维基百科节点中的最短路径关系,对 2 个词条之间的关系进行刻画.

2 主要工作

2.1 预处理

互联网百科全书是当前能够随意查阅的最大的、最完整的人类知识库,其中维基百科是人类共同参与编撰的规模最大的互联网多语言百科全书,其与专业人士编撰的大英百科全书具有相近的正确率,但由于编撰自由,发展更加迅速. 截至 2016 年 7 月 25 日,英文维基百科总计拥有 5 201 640 篇文章、39 827 021 个页面,覆盖了人类的大部分知识领域,词条为最小知识单元,以链接跳转表现知识之间的联系. 维基百科本身的链接结构是人类经过时间积累的智慧结晶,编撰结构和人类的思考方式高度吻合,因此本文选取维基百科作为基本数据集. 我们使用原始数据集是完整的维基百科版本(2016-06-01),解压后得到一个 53.4 GB 的 XML 文件. 这个 XML 文件包含了所有的词条,而每个词条中都包含了指向其他词条的超链接. 传统的算法主要关注词条之间的关系,而我们希望能从更小的维度进行观察,本文将每个词条中的超链接对应的单词作为研究对象,称之为节点,该数据集包含了所有节点之间的链接关系. 我们处理后所使用的节点数量为 1 200 万,这是传统算法中研究对象数量的 5~10 倍.

本文的预处理工作主要包括 5 个步骤:

1) 在原始 XML 文件中,先抽取词条本身和其文本中超链接对应的单词. 这个数量远大于词条,我们称之为节点.

2) 规整词条格式,在节点中去除不合法的链接和词条,例如词条包含的锚链接等.

3) 删除空指向的词条.

4) 因为文件中有些节点首字母为小写,而在维基百科网页中,全部词条的首字母均为大写,因此将文档中节点的小写首字母转化为大写.

5) 在节点构成的网络中,用 Hadoop 集群将单项链接转化为双向链接,即把链接构成的有向图转化为无向图.

2.2 节点普遍联系的验证

在基于路径的语义分析的研究中,利用超链接寻找从一个词条到另一个词条的通路,都是建立在存在通路的假设前提下. 虽然根据经验可以基本判断维基百科的词条间应该是普遍连通的,但是缺乏有力验证. 我们在预处理的基础上对维基百科中节点的普遍联系进行验证,利用广度优先搜索算法在节点构成的无向图中生成连通分量.

我们获得了节点中连通分量的分布,如表 2 所示,维基百科中节点构成的网络中,一共包含 10 441 个连通分量,这些连通分量一共含有 12 280 497 个节点. 其中最大联通分量中含有 12 269 222 个节点,可以清楚地看出,99.91%的节点都包含在最大联通分量中. 经过观察,其他联通分量包含的节点是一些特殊字符,因此我们将最大联通分量视为研究对象. 这就意味着,如果我们将整个维基百科网络视为无向图,任意 2 个节点之间都存在着通路.

Table 2 Connected Component Number and the Corresponding Node Number

表 2 连通分量及对应节点数量

Connected Component	Number of Connected Component	Number of Nodes
1	12 269 222	12 269 222
12	18	216
13	14	182
2	7	14
14	5	70
9	4	36
23	3	69
124	2	248
10 440	1	10 440

2.3 维基百科节点间的最短路径

维基百科是由词条构成,词条文本中含有节点间的链接关系. 每一个维基百科词条的页面中,都包含了与之相关的其他词条,通过这些词条对其进行解释,因此我们可以直接使用维基百科而无需对文本做深层次的语义理解. 图 1 显示了维基百科词条间的链接关系,圆点代表词条,颜色代表词条距离选定词条的不同层级,虚线圆代表处于同一层级的词条,红色箭头代表词条间通过超链接的跳转. 具体

来说,红色圆圈代表选定的初始节点;绿色圆圈代表与初始节点存在直接链接关系的词条,也就是从初始节点出发,可以通过一次链接到达绿色节点,我们把这些绿色节点定义为第 1 层节点.

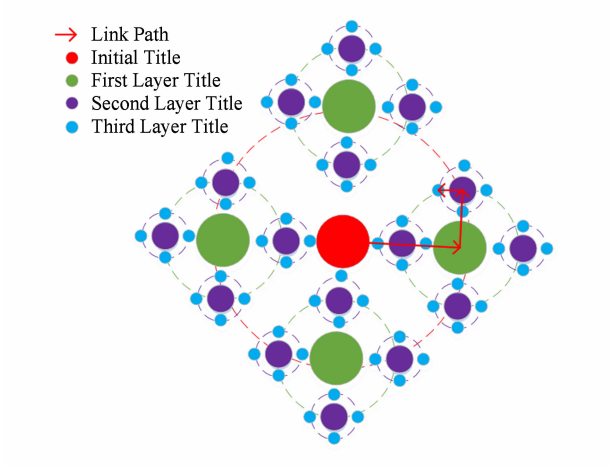

Fig. 1 The hyperlink relationship in Wikipedia nodes

图 1 维基百科中节点的链接关系

类似地,从绿色节点词条经过一次跳转,可以到达紫色节点,我们把紫色节点定义为初始节点的第 2 层节点.以此类推,我们把蓝色节点定义为初始节点的第 3 层节点.需要额外说明的是,在计算第 n 层节点的词条时,先排除处于 n 层中但与 n 层之前重复的部分词条.我们认为 2 个节点之间的最短路径越短,表明他们之间的关系越密切;2 个节点之间的路径越长,则表明他们之间的关系越遥远.这个观点也与人类的思维模式相同,面对一个具体问题,人类首先会联想到与这个问题最相关的因素,之后再深入到每个因素之中.举个例子,当提到"Milk"时,人们首先会想到有关的产品,例如"Cheese"等;通过"Cheese"会深入联想到制作所需的材料,例如"Pepper"等.而这和维基百科的构建方式刚好吻合,在维基百科中,词条"Cheese"和"Pepper",分别是词条"Milk"的第 1 层和第 2 层上的节点.

2.4 基于双向最短路径的语义相似性算法

在使用维基百科评估语义相似度的方法中,大多都利用了 2 个单词之间超链接所构建的路径[18].传统算法将单词 A 到单词 B 的路径长度和单词 B 到单词 A 的路径长度看作是等同的,并随机地从单词 A 和单词 B 中选取一个单词作为起始点,然后在这种模式下探寻到达另一个单词的超链接路径.但是这种路径方式有很大的局限性,因为大部分情况下,2 种方向的路径长度并不相同,可以通过一个显而易见的例子进行说明:当提到单词"成都"时,人们会自然地想到"中国";但是当提到"中国"时,却很难想到成都.从这个例子可以明显看出,从"成都"到"中国"和从"中国"到"成都"的距离(联想难度)明显不同.我们进一步通过维基百科的链接结构进行解释,如图 2 所示为传统算法考虑路径时链接路径的模式,给定的 2 个单词"Night"和"System",可以通过一个"Energy"页面进行中转,于是这 2 个单词的最短距离是 2.传统算法的思考模式将整个维基的链接网络看成一个无向图,任选一个单词作为起点并不会影响最短路径的长度.本文算法的灵感来自于对人类思维模式的思考:当人类考虑单词 A 和单词 B 相似性时,不会采用单方向的思考方式,而是采用双向的思考模式,正向距离和反向距离都会影响思考过程.我们将整个维基链接结构视为有向图,在有向图中寻找最短路径.

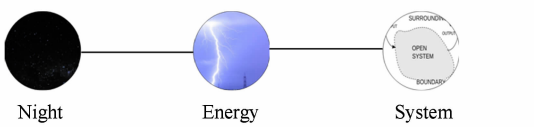

Fig. 2 The representation of traditional shortest link path

图 2 传统链接最短路径表示

如图 3 所示,人类联想 2 个单词相关性的最优解法是通过双向最短路径(我们称之为前向路径和反向路径),这是因为维基百科中 2 个单词会因为起点和终点的转换引起最短路径的长度不同.具体来说,前向路径中从"Night"出发,经过"Energy"1 次跳转后到达"System";反向路径中从"System"出发,依次经过"Interaction","Action","Time"3 次跳转后到达"Night".由此看出,"Night"和"System"中的前向最短路径长度为 2,反向最短距离为 4.

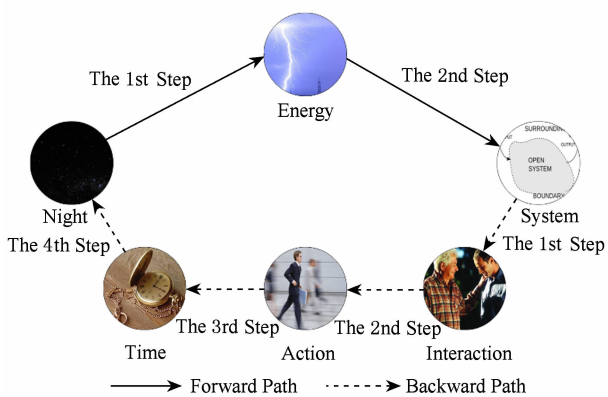

Fig. 3 The representation of bidirectional shortest link path

图 3 双向链接最短路径表示

我们使用维基百科有 2 个原因:

1) 维基百科包含多种语言,其中英文版本约有 500 万的词条和1 000 万的标题,是现今人类在网络中最大的知识仓库,它由人类经过结构化编撰而成,正确率和权威性已经得到了普遍认可;

2) 维基百科具有良好的成长性,在每个知识的宽度和深度上,随着时间稳步增长.

我们定义 2 个单词之间的最短路径长度:

$$L=\frac{L_F+L_B}{2}, \tag{1}$$

其中,L_F 为前向最短路径长度,L_B 为反向最短路径长度.在通过双向最短路径进行语义相似性评估时,有一个需要解决的核心问题:如何设定最短路径的步长上限.因为尽管在我们验证了节点的普遍联通性,但是寻找 2 个节点间的最短路径时,仍会遇到路径过长的问题.为了解决这个问题,我们设计了一个实验:随机选取 300 000 个单词对,用广度优先搜索算法找到每对单词的前向最短路径长度,最后求得 2 个节点间的平均最短路径长度值 $E(L)=3.84$,标准偏差 $\sigma=1.73$.根据切比雪夫不等式:

$$P\{|L-E(L)|\geqslant k\times\sigma\}\leqslant\frac{1}{k^2}, \tag{2}$$

令 $k=3$ 可得平均路径长度 $L\geqslant 9.03$ 的概率小于 1/9,因此我们将最短路径的步长上限设置为 8,即超过 8 步就认为 2 个单词不存在相似性.

在运用双向最短路径衡量 2 个单词相似性的基础上,我们提出了一个新的评测文本相似性的方法.

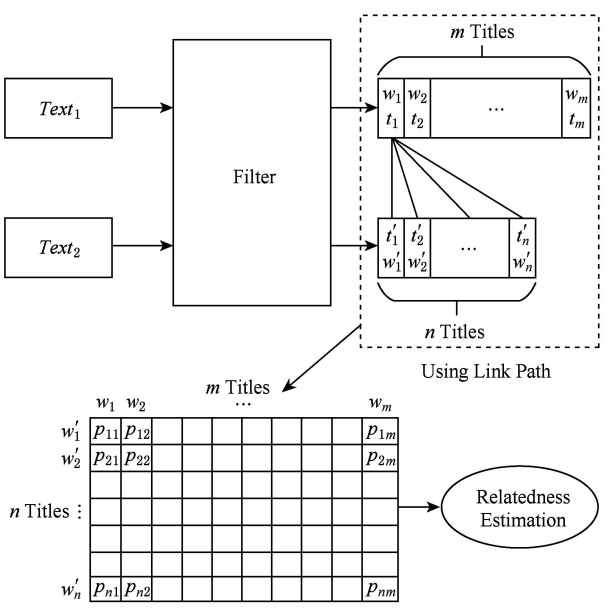

Fig. 4 The similarity evaluation of texts based on bidirectional shortest path

图 4 基于双向最短路径的文本相似性算法

具体算法图 4 所示,$Text_1$ 和 $Text_2$ 为需要进行相似性评估的文本,我们将其输入一个滤波器,这个滤波器会对文本进行预处理工作,包括去除停用词、词形归并等;之后通过将文本中的单词与维基词条比对后进行分词.将 $Text_1$ 划分成由 m 个词条构成的集合 $T_1=\{w_1,w_2,\cdots,w_n\}$,其中 w 代表每个词条,t 代表对应词条在 $Text_1$ 中出现的次数.同样地,将 $Text_2$ 划分成由 n 个词条构成的集合 $T_2=\{w_1',w_2',\cdots,w_n'\}$,其中 w'代表每个词条,t'代表对应词条在 $Text_2$ 中出现的次数.通过 $Text_1$ 中 m 个词条和 $Text_2$ 中的 n 个词条构成一个 $m\times n$ 的矩阵(假设 $m>n$),矩阵第 i 行第 j 列的值 m_{ij} 为单词 w_i 与单词 w_j' 的平均最短路径长度,即代表了 2 个单词的相似性;m 行分别包含 $Text_1$ 文本中可用单词的元组 $A=(w_1,w_2,\cdots,w_m)$;n 列分别包含 $Text_2$ 文本中可用单词的元组 $B=(w_1',w_2',\cdots,w_n')$.

下面我们通过矩阵分块来计算 $Text_1$ 文本和 $Text_2$ 文本的相关性:

1) 我们对矩阵进行重排,假设 $|A\cap B|=p$,重排元组 A 和 B 使得:

$$\forall\, 1\leqslant k\leqslant p,\ w_k=w_k', \tag{3}$$

这是将 A 和 B 中共有的单词排列于各自元组的前端,并保证下标相同.

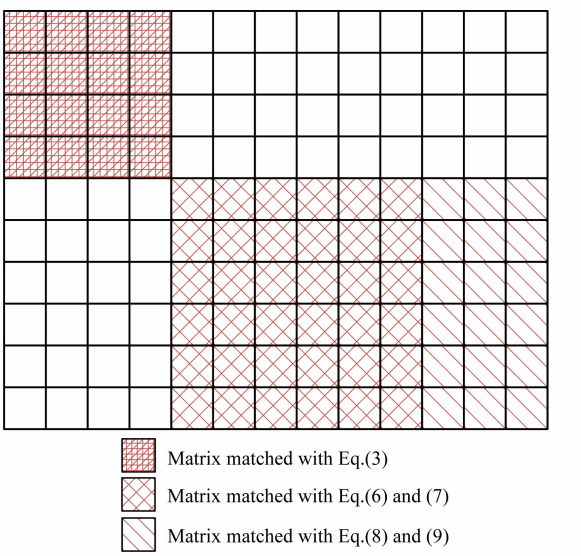

Fig. 5 Matrix partitioning

图 5 矩阵分块

2) 从 A 和 B 中删除共有的元素得到 $\overline{A}$ 和 $\overline{B}$,即令:

$$\overline{A}=(w_{p+1},w_{p+2},\cdots,w_m), \tag{4}$$

$$\overline{B}=(w'_{p+1},w'_{p+2},\cdots,w'_n), \tag{5}$$

其中，$w_{p+1}, w_{p+2}, \cdots, w_m, w'_{p+1}, w'_{p+2}, \cdots, w'_n$ 两两不同，这样我们得到了一个$(m-p)\times(n-p)$的子矩阵.

3) 对子矩阵进行重排，使得：

$$\forall (p+1)\leqslant r\leqslant s\leqslant m, t_r\geqslant t_s, \tag{6}$$

$$\forall (p+1)\leqslant r'\leqslant s'\leqslant n, t'_r\geqslant t'_s, \tag{7}$$

其中 t_r, t_s 分别为 w_r, w_s 在 $Text_1$ 文本中的权重；t'_r, t'_s 分别为 w'_r, w'_s 在 $Text_2$ 文本中的权重. 即行和列都按照各单词的权重降序排列.

4) 令 $q=\min(m,n)$，取：

$$\widetilde{A}=(w_{p+1}, w_{p+2}, \cdots, w_q), \tag{8}$$

$$\widetilde{B}=(w'_{p+1}, w'_{p+2}, \cdots, w'_q), \tag{9}$$

我们进一步得到一个 $q\times q$ 的方阵.

由于对于任意 $w\in\widetilde{A}$ 和 $w'\in\widetilde{B}$，相关性与 w 和 w'的顺序无关，正向和反向应用 Gale-Shapley 算法的结果相同，因此在 $q\times q$ 的方阵中应用 Gale-Shapley 算法.

5) 对 $q\times q$ 方阵进行重排，重排元组 $\widetilde{A}$ 和 $\widetilde{B}$ 使得：

$$\forall (p+1)\leqslant k\leqslant q, w_k \text{ 与 } w'_k \text{ 匹配}, \tag{10}$$

即相互匹配的元素下标相同.

6) 将最终获得的矩阵记为 $\boldsymbol{M}'$，其中第 i 行第 j 列的元素记为 m'_{ij}. 计算文本 A 和文本 B 的相关性：

$$corr(A,B)=\frac{\sum_{k=1}^{q} w_k\times m'_{kk}}{\sum_{k=1}^{q} w_k}, \tag{11}$$

$$w_k=\frac{1}{2}(t_k+t'_k). \tag{12}$$

3 实验评估

3.1 WS-353Ex 数据集

在语义相关性评估中，研究者通常会使用标准的黄金数据集 WS-353[21] (WordSimilarity-353). WS-353 中包含了 353 对英语单词，每一对都用一个范围在 0～10 的分数来表示它们的相关程度. 每个分数都是由 16 个人分别打出，最后取平均值而得到. WS-353 是根据人类的联想和判断，确定对语义相似度的评估标准. 因此，我们可以通过与其结果的对比，判断本文提出的算法能否与人类的联想机制契合.

我们对 WS-353 进行了改进，称之为 WS-353Ex (WS-353 extend version)，这是因为其中一些单词无法映射到维基百科的词条中. 之前的研究中一些研究者也发现了这一问题，但是一直没有得到有效的解决. Milne 等人[13]选择直接删除 60 个不合适的单词. Singer 等人[18]也发现了这个问题，并给出了改进版的数据集 WikipediaSimilarity-353，将单词对删减到 308 个，但是 Singer 等人忽视了引起这个问题的真正原因. 我们发现，自从 2002 年 WS-353 提出之后，维基百科经过 14 年的发展和变化，在数量级上发生了数十倍的增长，导致部分单词无法映射到维基百科中进行路径探索的原因是 WS-353 中的一些单词出现在重定向页面和消歧义页面. 为了解决这个问题，我们做了 2 项工作：

1) 将重定向的单词进行替换，例如 “Diego Maradona”更换为“Maradona”，“theater”更换为“theatre”；

2) 将维基百科中无法查到的单词删除，例如“defeating”.

我们对 2002 年的数据集 WS-353 进行了改进和修正，形成 WS-353Ex，最后保留了 344 对单词. 如表 3 所示，WS-353 在最初的 3 个维度(Word1, Word2, human mean)的基础上，增加了 3 个新的维度：前向最短路径长度 f、反向最短路径长度 b 和平均最短路径长度 a. 以“tiger”和“cat”举例，人类对 2 个单词相似性的打分是 7.35，从“tiger”出发经过 2 步能够链接到“cat”，从“cat”出发经过 3 步能够链接到“tiger”，“tiger”与“cat”的平均最短路径为 2.5.

Table 3 Demonstration of WS-353Ex Dataset

表 3 WS-353Ex 数据集样例

Word 1	Word 2	Similarity	f	b	a
love	sex	6.77	2	4	3
tiger	cat	7.35	2	3	2.5
tiger	tiger	10	0	0	0
computer	keyboard	7.62	2	2	2

3.2 语义相似性评估

在本节中，我们分别使用 WS-353Ex 和 Lee[22] 数据集来验证算法在单词和文本相似度评估的有效性. 首先使用 WS-353Ex 数据集进行单词相似度评估，具体步骤如下：

1) 基于在 3.1 节中相关方法所计算出的 WS-353Ex 中每对单词之间的 f 和 b，求出：

$$a=\frac{f+b}{2}. \tag{13}$$

2) 采用岭回归方法,分别使用单词对的 f,b,a 与对应的人类打分进行拟合:

$$\hat{h}(w,p)=w'_0+w_1p_1+\cdots+w_mp_m, \quad (14)$$

其中,h 代表人类的打分,$w=(w_1,w_2,\cdots,w_p)$ 代表系数,w_0 代表截断我们使用岭回求解:

$$\min_{\omega}\|P\omega-h\|_2^2+\alpha\|\omega\|_2^2, \quad (15)$$

其中,α 是复杂性系数,α 的值越大,收缩性越好,也就使得系数拥有更强的鲁棒性.

3) 我们采用斯皮尔曼等级相关系数、皮尔逊相关系数和谷本系数[1,4]来评价预测拟合模型的正确性,评估的分数区间为 0~1,越趋近于 1 说明该方法的准确度越高.实验结果验证了通过最短路径算法进行相似性评估的准确性.

本文设计了 3 组实验,分别使用 f,b,a 与人类打分值进行组合预测:实验随机选取了测试集和验证集(分别采用 3:7,5:5,7:3 的比例),重复 500 次取平均值.实验结果如表 4 所示,单独使用 f 和 b 最高只能达到 0.51 和 0.53 的分数,而使用 a 能够将分数稳定在 0.84.为了进一步的评估,我们将本文的算法与之前的一些重要相关研究结果作了比较,如表 5 显示,Milne 等人[13]所提出的一种基于超链接相似度的方法可以将分数提高到 0.78.在本文基本方法的基础上,我们观察到在 WS-353Ex 数据集中有一些明显错误样本数据,人工删除这些误差样本后重新进行评估,能够将准确率提高到 0.90.从表 5 中可以看出,通过本文提出的算法与 8 种传统算法的比较,验证了所述算法在单词相似度计算的有效性上明显超越现有算法.

Table 4 Comparison of Word Similarity Using Three Kinds of Shortest Path

表 4 基于 3 种最短路径评估单词相似性的分数对比

Path	Train Set: Test Set		
	3:7	5:5	7:3
f	0.46	0.51	0.48
b	0.49	0.53	0.52
a	0.77	**0.84**	0.79

为了探究在评估文本相似度的效果,我们基于双向最短路径算法在 Lee 数据集[26]上进行了验证,这个数据集包含了 50 个文本,以及每一对文本对应的人类打出的相似性分数.实验中随机抽取 1 000 对文本进行实验,得到分数的平均值.如表 6 所示,Hassan 等人[8]提出基于小规模语料库进行训练的 LSA 算法,能够得到 0.69 的分数.Gabrilovich 等人[11]提出了 ESA 算法,并得到 0.75 的分数.本文通过双向最短路径算法进行文本相似性分析,能够得到 0.77 的分数.基于双向最短路径的算法在单词相似性的评估上取得了很好的效果,但是在文本相似度上效果与之前的算法效果相当,我们认为这主要因为训练集较小,仅基于数据集 WS-353Ex,只包含 344 对单词;另一个影响因素是我们仅通过文本在维基百科网络中的链接关系进行评估,没有结合其他更多的属性.

Table 5 Comparison of Word Similarity Scores with 8 Methods

表 5 本文算法与 8 种算法基于单词相似度分数的比较

Method	Scores		
	Spearman Correlation	Pearson Correlation	Tanimoto Coefficient
Ref[21]	0.56	0.53	0.52
Ref[23]	0.55	0.61	0.49
Ref[12]	0.48	0.44	0.50
Ref[24]	0.55	0.47	0.57
Ref[11]	0.75	0.67	0.76
Ref[25]	0.78	0.85	0.86
Ref[4]	0.70	0.76	0.71
Ref[13]	0.79	0.74	0.77
Our Basic Method	0.84	0.83	0.82
Our Advance Method	**0.89**	**0.90**	**0.89**

Table 6 Comparison of Text Similarity Scores with 3 Methods

表 6 本文算法与 2 种算法基于文本相似度分数的比较

Method	Scores		
	Spearman Correlation	Pearson Correlation	Tanimoto Coefficient
Ref[8]	0.64	0.69	0.66
Ref[11]	0.75	0.72	0.71
Our Basic Method	0.75	0.70	0.73
Our Advance Method	**0.77**	**0.76**	**0.73**

4 总 结

本文提出了一种基于维基百科中超链接结构进行语义相似性评估的方法.这是首次在论文中考虑

到2个单词之间双向路径的不同,并证明了维基百科中词条的普遍连通性,发现了它们之间的平均最短路径长度约为3.84.考虑到人类双向思维的方式,我们应用双向最短路径长度设计了一种新的算法,该算法能够充分利用维基百科结构中的潜在人类特性.本文使用提出的算法分析单词相似性,取得了很好的效果,在此基础上提出一种使用矩阵分解进行文本相似性分析的方法,取得了良好的效果.另外,我们对传统的WS-353进行了修正和升级.本文最大的贡献在于:在仅使用维基百科中链接关系和数量较少的训练数据的情况下,取得了出色的评估效果.在未来的工作中,我们希望能够结合其他文本特征,并增加更多训练数据来改进评估模型.

参 考 文 献

[1] Zhang Libo, Zhang Fei, Luo Tiejian. Knowledge structure evolution and evaluation method in computer science [J]. Journal of University of Chinese Academy of Sciences, 2016, 33(6): 844-850 (in Chinese)
(张立波, 张飞, 罗铁坚. 计算机科学知识体系演化与评估方法[J]. 中国科学院大学学报, 2016, 33(6): 844-850)

[2] Xin Yu, Yang Jing, Tang Chuheng, et al. An overlapping semantic community detection algorithm based on local semantic cluster [J]. Journal of Computer Research and Development, 2015, 52(7): 1510-1521 (in Chinese)
(辛宇, 杨静, 汤楚蘅, 等. 基于局部语义聚类的语义重叠社区发现算法[J]. 计算机研究与发展, 2015, 52(7): 1510-1521)

[3] Wang Junhua, Zuo Wanli, Yan Zhao. Word semantic similarity measurement based on Naive Bayes model [J]. Journal of Computer Research and Development, 2015, 52(7): 1499-1509 (in Chinese)
(王俊华, 左万利, 闫昭. 基于朴素贝叶斯模型的单词语义相似度度量[J]. 计算机研究与发展, 2015, 52(7): 1499-1509)

[4] Yazdani M, Popescu-Belis A. Computing text semantic relatedness using the contents and links of a hypertext encyclopedia: Extended abstract [J]. Artificial Intelligence, 2013, 194: 176-202

[5] Giles J. Internet encyclopaedias go head to head [J]. Nature, 2005, 438: 900-901

[6] Gabrilovich E, Markovitch S. Computing semantic relatedness using Wikipedia-based explicit semantic analysis [C] //Proc of the 20th Int Joint Conf on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 2007: 1606-1611

[7] Zesch T, Müller C, Gurevych I. Using Wiktionary for computing semantic relatedness [C] //Proc of AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2008: 861-866

[8] Hassan S, Mihalcea R. Cross-lingual semantic relatedness using encyclopedic knowledge [C] //Proc of Conf on Empirical Methods in Natural Language Processing. New York: ACM, 2009: 1192-1201

[9] Syed Z S, Finin T, Joshi A. Wikipedia as an ontology for describing documents [C] //Proc of Int Conf on Weblogs and Social Media. Menlo Park, CA: AAAI, 2008

[10] Coursey K, Mihalcea R. Topic identification using Wikipedia graph centrality [C] //Proc of Conf on the North American Chapter of the Association of Computational Linguistics. New York: ACM, 2009: 117-120

[11] Gabrilovich E, Markovitch S. Wikipedia-based semantic interpretation for natural language processing [J]. Journal of Artificial Intelligence Research, 2014, 34(4): 443-498

[12] Strube M, Ponzetto S P. WikiRelate! computing semantic relatedness using Wikipedia [C] //Proc of the 19th National Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2006: 1419-1424

[13] Milne D, Witten I H. An open-source toolkit for mining Wikipedia [J]. Artificial Intelligence, 2013, 194: 222-239

[14] Yeh E, Ramage D, Manning C D, et al. WikiWalk: Random walks on Wikipedia for semantic relatedness [C] //Proc of Workshop on Graph-Based Methods for Natural Language Processing. New York: ACM, 2009: 41-49

[15] Dallmann A, Niebler T, Lemmerich F, et al. Extracting semantics from random walks on Wikipedia: Comparing learning and counting methods [C] //Proc of the 10th Int AAAI Conf on Web and Social Media. Menlo Park, CA: AAAI, 2016: 33-40

[16] Witten I, Milne D. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links [C] //Proc of AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy. Menlo Park, CA: AAAI, 2008: 25-30

[17] Wei Bifan, Liu Jun, Ma Jian, et al. Motif-based hyponym relation extraction from Wikipedia hyperlinks [J]. IEEE Trans on Knowledge and Data Engineering, 2014, 26(10): 2507-2519

[18] Singer P, Niebler T, Strohmaier M, et al. Computing semantic relatedness from human navigational paths: A case study on Wikipedia [J]. International Journal on Semantic Web & Information Systems, 2013, 9(4): 171-172

[19] Niebler T, Schlör D, Becker M, et al. Extracting semantics from unconstrained navigation on Wikipedia [J]. Künstliche Intelligenz, 2016, 30(2): 1-6

[20] Ellery W, Dario T. Wikipedia Clickstream, v21.0 [OL]. [2016-09-24]. https://meta.wikimedia.org/wiki/Research:Wikipedia_clickstream

[21] Finkelstein L, Gabrilovich E, Matias Y, et al. Placing search in context: The concept revisited [C] //Proc of the 10th Int Conf on World Wide Web. New York: ACM, 2001: 406-414

[22] Lee M D. An empirical evaluation of models of text document similarity [J/OL]. Cognitive Science, 2005 [2016-09-24]. https://core.ac.uk/download/pdf/22875578.pdf

[23] Jarmasz M. Roget's thesaurus as a lexical resource for natural language processing [D]. Ottowa, Canada: University of Ottowa, 2012

[24] Hughes T, Ramage D. Lexical semantic relatedness with random graph walks [C] //Proc of the 45th Conf on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Stroudsburg, PA: ACL, 2007: 581-589

[25] Agirre E, Alfonseca E, Hall K, et al. A study on similarity and relatedness using distributional and WordNet-based approaches [C] //Proc of Conf on the North Americ Human Language Technologies. New York: ACM, 2013: 19-27

[26] Hoerl A E, Kennard R W. Ridge regression: Biased estimation for nonorthogonal problems [J]. Technometrics, 1970, 12(1): 55-67

Zhang Libo, born in 1989. Received his PhD degree in computer software and theory from the University of Chinese Academy of Sciences. Assistant research professor in the Institute of Software, Chinese Academy of Sciences. His main research interests include image processing, pattern recognition, knowledge graph and deep learning.

Sun Yihan, born in 1994. PhD candidate. Her main research interests include recommendation system, information retrieval, machine learning and distributed storage system.

Luo Tiejian, born in 1962. PhD, professor. Director of the Information Dynamic and Engineering Applications Laboratory. His main research interests include Web mining, large scale Web performance optimization and distributed storage systems.