第 37 卷 第 7 期
Vol.37 No.7

计 算 机 工 程
Computer Engineering

2011 年 4 月
April 2011

• 人工智能及识别技术 • 文章编号：1000—3428(2011)07—0193—03 文献标识码：A 中图分类号：TP391

基于维基百科的语义相似度计算方法

盛志超，陶晓鹏

(复旦大学计算机科学技术学院，上海 200433)

摘 要：针对目前语义计算准确率低、可理解性差的问题，提出一种基于维基百科的语义相似度计算方法。不同于利用分类信息计算词的语义相似度，该方法利用页面的链接信息，通过模仿人类联想的方式计算不同词之间的相似度，所得到的结果较容易被理解，并结合词语的语义类别提高计算结果的准确率。和现有算法的对比实验证明了该方法的优越性。

关键词：页面网；类别网；维基百科；人脑思维

Semantic Similarity Computing Method Based on Wikipedia

SHENG Zhi-chao, TAO Xiao-peng

(School of Computer Science, Fudan University, Shanghai 200433, China)

【**Abstract**】Aiming at the low accuracy and poor intelligibility of current algorithms for semantic analysis, a semantic similarity computing method based on Wikipedia is proposed. Different from computing word's semantic similarity by category information, this method uses link information to calculate the similarity of different words in a way like human thinking. Result can be easily understood and the accuracy rate can be increased with semantic category. Experiment compared with current algorithms proves its advantage.

【**Key words**】PageNet; CategoryNet; Wikipedia; human thinking

DOI: 10.3969/j.issn.1000-3428.2011.07.065

1 概述

语义相似度的计算在自然语言处理领域有着非常重要的意义，是信息检索、文本分类等相关领域的基础。让计算机能比较语义是一件非常困难的事，需要做大量的工作，比如建立知识库、让计算机获取知识知识、比较语义等。从语义相似度计算的发展来看，这个领域的计算方法大致可以分成 2 类：基于规则的方法和基于统计的方法。考察并总结了已有的方法，总体而言，这些方法的实验结果都不够理想，有许多值得研究和提高的地方。

本文介绍一种不同以往的方法，这种方法能够更好地符合人脑的思维模式。比如，人脑要衡量如下 2 个词语的相似度：“兵”和“将”，可能想到，它们都是军队的一部分，都是象棋中的一个棋子等。人脑会通过已有的知识衡量它们的相似度。本文提出的方法尝试赋予计算机丰富的知识(维基百科)，然后模拟人脑的联想能力衡量词语的相似度。在维基百科中，每个页面都只有一个主题(即页面标题)，页面之间相互连接。WPNRelate 正是利用这些链接信息评估词语之间的相似度。

2 相关工作

语义相似度的计算可分成 2 大类：不利用知识库和利用知识库。前者又可以分成 3 类：(1)通过计算不同词的共现计算不同词之间的相似度；(2)通过网络信息的方法计算不同单词之间的相似度；(3)一些隐含语义的计算方法。这类算法的不足是：它们所用的词库不够强大，不足以在真正的应用中派上用场。另外，虽然 LSA 算法[1]的结果取得了相当程度的提升，但是人们却难以解释其工作的过程，也导致在不同应用环境下出现较大的性能差异。利用知识库的方法是现在比较常用的方法，各种知识库层出不穷。其中一个比较有代表性的就是 WordNet。

WordNet 是一个结构化很好的知识库，它不但包括一般词典的功能，另外还有词的分类信息。目前，基于 WordNet 的方法相对来说已经比较成熟，比如路径方法[2](简称 lch)、基于信息论方法[3](简称 res)等。但是由于 WordNet 词库相对较小，并不能在实际中使用它们。所以引入了维基百科作为背景知识，用来计算语义相关度。维基百科是一个公开的数据库，它的数据量非常大，可以说平时人们用到的词绝大部分都能在其中找到对应的页面，而且这个巨大的数据集一直在不停的添加之中。

维基百科具有很好的结构化信息，可以将维基百科看成 2 个巨大的网络：(1)由页面组成的网络(以下称为页面网)，如图 1 所示。其中，每个点表示一个页面；每根线条表示一个连接。不同的页面之间通过入链和出链相互连接在一起。(2)由类别组成的网络[4](以下称为类别网)，如图 2 所示，其中，每个矩形框代表一个维基百科的一个类。不同的类别通过子类和父类的关系相互连接在一起。

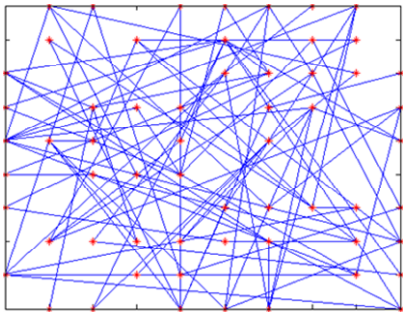

图 1 页面网

作者简介：盛志超(1984－)，男，硕士研究生，主研方向：语义比较，文本分类；陶晓鹏，副教授

收稿日期：2010-10-13 **E-mail**：082024062@fudan.edu.cn

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

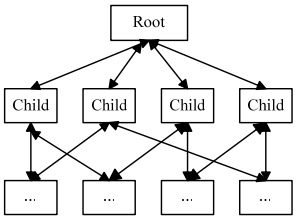

图 2 类别网

在使用维基百科作为知识库的算法中，WikiRelate 算法[5]在计算结果和计算速度上比较均衡。WikiRelate 实际上表示一组算法，它将上述的 WordNet 上具有代表性的方法(lch、res 等)都重新基于维基百科的类别网实现。维基百科类别网的结构化信息和 WordNet 中很相似，有所不同的是：在 WordNet 中，不同词性的词之间的相似程度是等于 0 的，而在维基百科中没有这样的一个限制，所以只要是维基百科上存在的词，即使词性不同，WikiRelate 算法也能比较它们之间的相似度，因此，WikiRelate 方法比基于 WordNet 的方法更贴近现实。

3 基于页面网的语义相似度计算方法

本文提出的方法是也是基于维基百科，但是与已有方法(比如 Wikirelate)不同的是，采用页面网而不是类别网作为背景知识。大体而言，该方法试图模仿人类联想的方式。假设给定一个词：“老虎”，然后问：和它相关度最高的词是什么？可能最先想到的是“狮子”、“豹子”等，因为它们有很多共同特征，这就是说它们可能是同一类动物。另外可能想到“凶猛”、“平原”、“森林”、“迅速”等与“老虎”相关的词。为了使计算机也能“联想”，具体的方法如下所述。

3.1 等权重下页面间最短路径的查找方法

维基百科的页面网可以被视为一个有向图，即每个页面对应一个节点，每个有向边对应一个链接。为了简化问题，将这个有向图变成无向图，即将每个页面上的出链和入链都看作双向链接，并合并相同的链接。然后，假定所有链接的权重相等，求取一个节点到另一个节点的最短路径，这个路径可以用来衡量 2 个节点的相似度。虽然无向图的最短路径的求取算法已经很成熟了，但是考虑到维基百科数据库的特殊性，采用一种更适合于维基百科特点的最短路算法。

维基百科的数据量非常大。通常，一个页面的链接非常多，从几十到几千不等，这就保证所求的路径不会很长(实验表明一般小于 5)。通过两端同时查找向中间靠拢的方式找到最短路径，就是将这 2 个需要求最短路径的页面分别作为根节点，使用宽度优先的方式同搜索这个图，只要这 2 棵树上出现了相同的节点就会形成一条连接这 2 个页面的路径。如图 3 所示，页面 A 和页面 B 同时进行扩展，只要找到 A 到 B 的路就停止搜索。由于每个页面的链接非常多，使得所访问的页面数量大大减少，即访问磁盘的次数大大减少，可以节省很多时间。为了保证算法的速度要求直接把找到的第 1 条路径当作最短路径。

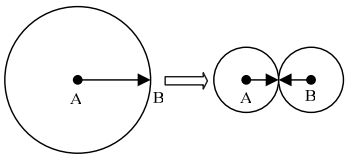

图 3 最短路径的双向搜索

3.2 链接的加权

为了增加问题的复杂性，让假设更接近真实的情况：维基百科的页面之间的链接应该具有不同的权重。比如，与某个页面(记为 pageX)相链接的页面有很多，不能简单地认为这些页面与 pageX 之间的关联程度是相同的。这就要求给这些链接加上权重，采用类似文本处理中的 TFIDF 技术给链接到 pageX 的连接加权。

在文本处理中 TFIDF 的值是用来衡量不同词语和文章主题的相关程度的。在维基百科中，每个页面只阐述一个主题，即题目。可以将这个主题看成一个词语。假如 page2 的题目出现在 page1 的说明文档当中，就可以使用 TFIDF 技术计算出 page2 题目与 page1 之间的关联程度。

在计算 TFIDF 时，2 个必不可少的数据就是 *TF* 和 *IDF*。*TF* 是词语在文档中出现的频率，这个比较好衡量；*IDF* 是反文档频率：

$$IDF(word)=\frac{Set(word)}{ALL} \tag{1}$$

其中，$Set(word)$ 表示含有词语 *word* 的文章总数；*ALL* 表示数据集中的文章总数。但是维基百科的数据量非常大，计算 $Set(word)$ 将会非常耗费时间。为了提高算法的速度，使用入链的数量代替 *IDF*。

在维基百科中，每个页面都有入链，即这个页面的指入型链接，指入型链接越多说明这个页面被维基百科中的其他文章引用的次数就会越多，也就是这个页面题目出现在其他文章中的次数也就越多，那么它的 *IDF* 越大。由于维基百科中有一些入链没有被标注出来，因此入链数量不一定精确地等于 *IDF*，但是可以近似地认为它们相等。即采用 *IDF* 计算式如下：

$$IDF(page(word))=inlinks(page(word))$$

其中，$page(word)$ 表示词语 *word* 对应的维基百科的页面；$inlinks(page(word))$ 表示这个页面的入链数量。这样得到的链接权重公式如下：

$$weight(word1,word2)=\frac{tf_{page(word1)}word2}{IDF(page(word2))} \tag{2}$$

其中，$tf_{page(word1)}word2=\frac{time_{page(word1)}word2}{lengthof(page(word1))}$；$time_{page(word1)}word2$ 表示 *word*2 在 $page(word1)$ 中出现的次数，分母表示 $page(word1)$ 的所有单词数的和。

3.3 加权无向图最短路径的查找方法

经过 3.2 节的处理，维基百科页面网成为带权重的无向图，现在的任务是在这个图上寻找权重最大(即相似度最大)的路径(称为权重最大路径)。简单的做法是，给整个页面网的边计算权重，调用已有的、稍作变通的最短路径算法。但是给维基百科中的所有链接加权的计算量非常大，这里设计了新的高效的近似算法。

(1)采用边的权重相乘的方式得到路径的相似度，即假设存在一条从页面 *p*1 到 *p*4 的路径，<*p*1,*p*2,*p*3,*p*4>，那么这条路径的相似度就是：

$$\prod_{i=1}^{4} weight(p_i,p_{i+1}) \tag{3}$$

通过实验发现 $weight(p_i,p_{i+1})$ 的值一般都很小(一般落在 $(10^{-3},10^{-5})$区间)。计算式(3)的值主要取决于乘数的个数，即路径所含边的个数。

(2)取决于所含边的权重。将权重最大路径的寻找过程分成 2 个独立的步骤：求出节点之间包含边数最少的路径(即等权重无向图中的最短路径)，再给这个路径加权。这样得到的路径是近似带权重最短路径，因为在等权重无向图中可能有多条距离相等的最短路径，而本文只是选了第 1 条作为最短

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

路径，这样做是为了权衡程序的运行速度，所以是一个近似结果。

3.4　类别信息的考量

上述方法还可以结合维基百科类别网提供的类别信息。其基本思想是，如果 2 个词语对应的 2 个页面在维基百科的页面网上是相似的，而且 2 个词语的类别在维基百科的类别网上也是相似的，那么能够更有把握地认为这 2 个词语是相似的。其具体方法如下：采用基于信息量的方法(res)计算出 2 个词语类别的相似度，将这个值乘以 3.3 节得到的 2 个词语页面的相似度。这个乘积就是最终衡量词语相似度的数值。

这个方法在实验中有明显的效果。可以这样理解：比如，计算“老虎”与“草原”、“森林”、“狮子”、“豹子”等词语的相似度。仅根据页面的链接信息发现与“老虎”相似度较高的词语是“草原”、“森林”，而“狮子”、“豹子”与“老虎”的相似度较低。显然，这与常识不太符合，加入类别信息后，“狮子”、“豹子”与“老虎”的相似度有了明显的提高。

4　语义信息数值比较实验

采用 3 个数据集进行实验，它们是 WS-353[1]、M&C 和 R&G。这些数据集都包含多条记录，每条记录包含一对词语和这对词语的语义相关信息，这些语义信息都是通过对人测试的方式得到。实验方法就是把这些人工得到的数值与本文方法获得的数值进行比较。具体而言，M&C 包含 30 个名词对；R&G 包含 65 个同义词对，它包含了 M&C 的全部词语对；WS-353 包括 353 个词语对，分成 2 组，一组是包含 200 个词语对的训练集，另一组是包含 153 个词语对的测试集。特别是 WS-353 的测试集，由于它包含的词语对的语义关系比较复杂，很适合用于算法的测试，将它记为 WS-353Test。

本文采用的维基百科数据集的版本是 enwiki-20070206，其中有近 200 万的页面和 30 多万的类别，页面的链接数为 9 000 多万个。表 1 记录了不同算法在 WS-353Test 上的部分运行结果(前 15 个词语对)。其中，统计结果记录原数据集中人工获得的相似度；Lch 记录文献[5]提出的 WikiRelate 中 lch 算法获得的结果；WPNRelate 记录本文 3.2 节、3.3 节中提出的纯粹基于维基百科页面网的算法获得的结果；加入类别信息记录本文 3.4 节中提出的加入维基百科类别网信息的算法获得的结果。值得一提的是，其中的所有数值都进行了归一化处理，使得它们都落在区间[0,10]上，这样便于比较。

表 1　WS-353Test 前 15 个词比较结果

Word1	Word2	统计结果	Lch	WPNRelate	加入类别信息
Love	Sex	6.77	7.35	8.15	6.58
Tiger	Cat	7.25	4.14	5.69	5.99
Book	Paper	7.46	9.13	8.44	9.30
Computer	Keyboard	7.62	8.57	5.53	5.83
Computer	Internet	7.58	6.38	8.20	7.51
Plane	Car	5.77	7.02	5.30	5.27
Training	Car	6.31	9.22	4.66	4.90
Telephone	Communication	7.50	7.01	5.87	5.83
Television	Radio	6.77	7.67	5.70	5.21
Media	Radio	7.42	8.43	5.81	5.77
Drug	Abuse	6.85	6.71	6.57	5.30
Bread	Butter	6.19	7.40	4.87	4.83
Cucumber	Potato	5.92	4.73	5.34	5.62
Doctor	Nurse	7.00	8.57	5.63	5.92
Professor	Doctor	6.62	8.57	6.84	7.54

实验的下一步是将各个算法的结果与人工的结果进行比较，本文采用 Spearman 相关系数来衡量不同数据列表的相似度，比较结果如表 2 所示。

表 2　Spearman 系数比较结果

数据集	WordNet	WikiRelate	WPNRelate
M&G	0.37~0.82	0.23~0.46	0.49
R&G	0.34~0.86	0.31~0.53	0.54
WS-353	0.21~0.34	0.19~0.48	0.52
WS-353Test	0.21~0.35	0.22~0.55	0.58

在表 2 中，WordNet 表示基于 WordNet 的算法获得的结果[3]；WikiRelate 表示基于 WikiRelate 的算法获得的结果，类似 WordNet，这里也是一组算法，因此，给出了一个结果范围；WPNRelate 表示本文 3.2 节、3.3 节中提出的算法(称为 WPNRelate)获得的结果。表 2 对比显示，本文提出的算法在所有测试数据集上都明显优于基于 WikiRelate 的算法，在 WS-353 的 2 个数据集(全集和测试集)上都优于基于 WordNet 的算法，在 M&G 和 R&G 的 2 个数据集上，也优于大多数基于 WordNet 的算法。总体而言，本文提出的方法优于现有的方法。

最后用实验检验了引入维基百科类别网信息的有效性，主要是比较未加入类信息的 WPNRelate 方法和加入类信息的 WPNRelate 方法的实验数据，结果如表 3 所示。在 WS-353 的 2 个数据集上，加入类信息的 WPNRelate 方法的结果都要优于未加入类信息的方法。

表 3　加入类信息的 WPNRelate

数据集	WPNRelate	CWPNRelate
WS-353	0.521 3	0.534 2
WS-353Test	0.584 9	0.601 6

5　结束语

本文提出了一种全新的词语语义相似度的计算方法，它基于维基百科的页面网络的链接信息，并结合了改进的 TFIDF 技术，计算词语之间的语义相似度。从实验结果上看，这个新方法产生结果的准确度优于已有的方法。如果允许更充分的计算时间，还提出了一种算法，通过模仿人类思维的方式，加入词语的类别信息，获得了更好的结果。

下一步的研究方向为：(1)由于算法复杂度的问题，现在采用的最短路径算法是一个近似算法，将来可以改进这个算法。(2)对于词语而言，维基百科的页面信息和类别信息是 2 种很不一样的知识源，结合这 2 种知识的方法太过简单化，缺少理论依据，将来可以研究更好的结合方式。

参考文献

[1] Jurafsky D, Martin J H. 自然语言处理综论[M]. 冯志伟, 孙　乐, 译. 北京: 电子工业出版社, 2005.

[2] Leacock C, Chodorow M. Combining Local Context and WordNet Similarity for Word Sense Identification[EB/OL]. (1998-05-18). http://www.bibsonomy.org/bibtex/2087c974e471792dd1fa536aa6a75e0bc/asalber.

[3] Resnik P. Using Information Content to Evaluate Semantic Similarity in a Taxonomy[C]//Proc. of the 14th International Joint Conference on Artificial Intelligence. [S. l.]: Springer, 1995: 448-453.

[4] 史天艺, 李明禄. 基于维基百科的自动词义消歧方法[J]. 计算机工程, 2009, 35(18): 62-65.

[5] Struve M, Ponzetto S P. WikiRelate! Computing Semantic Relatedness Using Wikipedia[C]//Proc. of Association for the Advancement of Artificial Intelligence. Boston, USA: IEEE Press, 2006: 1419-1424.

编辑　顾逸斐

© 1994-2012 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net