【之前一直听说根据维基百科中的信息计算相似度，但是一直不知道是如何计算的，今天看了这篇文章，大概了解了，同时也知道了一些关于 WordNet 的东西。】

【题目】基于维基百科的语义相似度计算方法
【作者】盛志超，陶晓鹏
【单位】复旦大学计算机科学技术学院
【期刊】计算机工程
【时间】2011 年 4 月

【方法描述】WPNRelate，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。

【相关工作】

语义相似度计算的计算方法大致可以分成两类：基于规则的方法和基于统计的方法；

语义相似度的计算可分成两大类：不利用知识库和利用知识库。

不利用知识库可以分为三类：1、通过计算不同词的共现计算不同词的相似度；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。

LSA 算法（潜在语义分析）的结果取得了相当程度的提升，但是人们难以解释其工作过程，导致在不同应用环境下有较大的性能差异。

利用知识库的方法是目前比较常用的方法，比较有代表性的就是 WordNet。由于 WordNet 词库相对较小，并不能在实际中使用它们。

【维基百科】

维基百科具有很好的结构化信息，可以将维基百科看作两个巨大的网络：

（1）页面网：每个点表示一个页面；每根线表示一个链接，不同的页面之间通过入链和出链相互连接在一起；

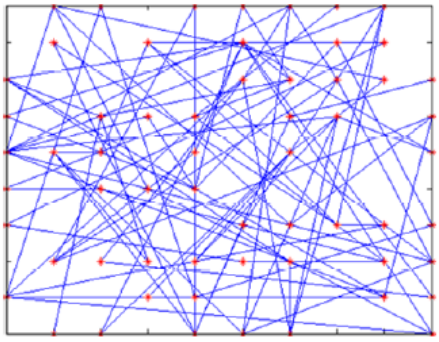

图 1　页面网

（2）类别网：每个矩形框代表维基百科中的一个类，不同的类别通过子类和父类的关系相互连接在一起。

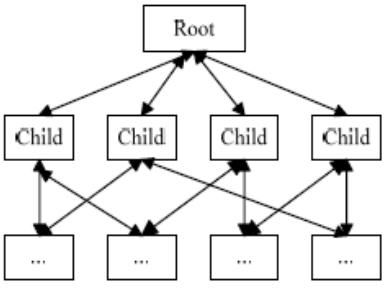

图 2　类别网

使用维基百科作为知识库的算法中，WikiRelate 算法在计算结果和计算速度上比较均衡。WikiRelate 实际上是一组算法，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。维基百科的类别网的结构化信息和 WordNet 中很相似，不同的是：在 WordNet 中，不同词性的词之间的相似度为 0，而在维基百科中没有这种限制，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。

【基于页面网的语义相似度计算方法】

【等权重下页面间最短路径的查找方法】

也就是把维基百科的页面网看作无向图，考虑到维基百科数据量非常大，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，就是将这两个待求最短路径的页面分别作为根节点，使用 BFS 方式同时搜索，只要这两棵树上出现了相同的节点就会形成一条连接 2 个页面的路径。

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。

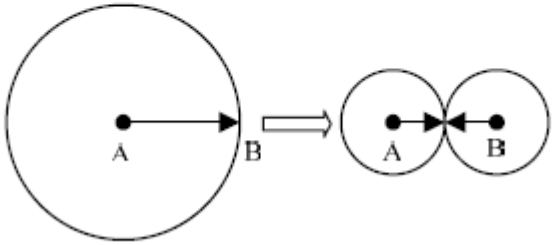

图 3　最短路径的双向搜索

【链接的加权】

为了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。

在文本处理中的 TFIDF 的值是用来衡量不同词语和文章主体的相关程度的。在维基百科中每个页面只阐述一个主题，即题目。可以将这个主题看作一个词语。例如 page2 的题目在 page1 的说明文档中出现，就可以使用 TFIDF 技术计算出 page2 题目与 page1 之间的关联程度。

TF：词语在文档中出现的频率；

IDF：在文本处理中的计算方法如（1）所示。

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

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，也就是这个页面的题目出现在其他文章中的次数也就越多，那么它的 IDF 越大。（由于维基百科中的一些入链没有被标注出来，因此入链数量不一定精确地等雨 IDF，但是可以近似的认为它们相等。）权重计算方法如下：

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

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：

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

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

【加权无向图最短路径的查找方法】

(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 $p1$ 到 $p4$ 的路径，$<p1,p2,p3,p4>$，那么这条路径的相似度就是：

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

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。

【类别信息的考量】

基本思想：如果两个词语对应的两个页面在维基百科的页面网上是相似的，而且两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两个词语是相似的。

具体方法：采用基于信息量的方法（res）计算出 2 个词语类别的相似度，将这个值与这个 2 个词语的页面相似度相乘。

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。

【实验结果】

数据集：WS-353，M&C，R&G。

M&C 包含 30 个名词对；

R&G 包含 65 个同义词对；

WS-353 包含 353 个词语对，分成两组，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。

维基百科数据集：enwiki-20070206，其中近 200 万页面和 30 多万类别，页面的链接数为 9000 多万个。

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。

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