网络出版时间：2017-10-26 16:49:51
网络出版地址：http://kns.cnki.net/kcms/detail/31.1289.TP.20171026.1649.002.html

结合路径与词林编码的词语相似度计算方法

王松松，高伟勋，徐逸凡
(上海师范大学信息与机电工程学院 上海 200134)

摘 要：该文提出了一种基于路径与同义词词林编码相结合的词语语义相似度计算方法(PathCoding 算法)，该方法通过使用两个词语的词林编码以及它们在词林中的的路径结构来计算两个词语的语义相似度。在计算两个词语之间的相似度时，使用局部敏感哈希算法将两个词语的词林编码转换成两个二进制，使用海明距离来计算两个二进制之间的距离。同时将同义词词林树形结构中不同层次赋予不同的权重，将词林编码与路径结构相结合，来实现两个词语之间的相似度计算。实验表明，该方法在 MC 数据集和 RG 数据集上的词语相似度计算值与人工判定值进行比较，分别获得了 0.8668 和 0.8974 的皮尔逊相关系数，该结果优于目前词语相似度的计算结果。

关键词：同义词词林；路径；编码；词语相似度；局部敏感哈希；语义

WORD SIMILARITY BASED ON PATH AND CILIN CODING

Wang Songsong，Gao Weixun，Xu Yifan
(College of information and Mechatronic Engineering,Shanghai Normal University,Shanghai 200134,China)

【Abstract】In this paper, we propose an approach to measure semantic similarity between words(PathCoding Algorithm). This approach calculates the semantic similarity of two words by using the Cilin coding of two words and their path structure in Cilin. The local sensitive hash algorithm is used to convert the Cilin coding of the two words into two binary vectors and then the Hamming distance is employed to measure the distance between these two binary vectors. The different weights are assigned to different levels of the tree structure of the Tongyici Cilin and the calculation of similarity between the two words is then weighted by the path structure. The experiments show that our approach outperforms the conventional approaches to word similary measure, i.e., the performance in terms of Pearson correlation coefficient between the human judgments and the machine calculation achieves 0.867 and 0.897 on MC dataset and RG dataset, respectively.

【Key words】Tongyici Cilin；path；word code；word similarity；Locality Sensitive Hashing；semantic

0 概述

词语相似度计算是文本数据处理的基础，在信息检索、词义消歧、机器翻译等方面都有广泛的应用。

在词语相似度的研究方法中，主要可以分为基于统计的计算方法和基于语义分析的计算方法。其中，基于统计的方法主要是使用词语间的概率分布，包括信息熵方法、LDA 方法和词语共现方法等。王小林[1]等人在传统词语相似度计算方法的基础上，引入了词表相似度的概念，在词语义原层面上引入信息熵的概念，利用信息熵实现了词语之间的相似度计算。吕亚伟[2]等人将词语的特征向量映射为词语的主题分布来计算词语间的相似度，实现了基于 LDA 的词语相似度计算方法，不仅有效的降低了特征维度，而且具有较好的词语相似度计算效果。Natalia Loukachevitch[3]等人提出了利用相邻句子的词语共现来计算词语相似度，将词语共现与 word2vec 相结合，实现了词语相似度的计算。传统基于统计的方法主要是依赖于语料库的训练，对于未登录词以及词语语义上的问题无法很好的处理，在相似度计算上降低了结果的准确性。

基于语义分析的计算方法主要是通过知识库对词语构建语义关系，包括 HowNet、同义词词林和 WordNet. Mengjia Fan[4]等人对 HowNet 的结构进行分析，分别从义原和概念两方面来计算词语之间的相似度。陈宏朝[5]等人在同义词的基础上，分别从词语的路径深度和路径距离来计算词语之间的相似度。Dean J.Jones[6]等人使用 WordNet 分别从同义词和反义词的角度来计算词语之间的相似度。

作者简介：王松松(1991-)，男，硕士研究生，主要研究方向：自然语言处理、数据挖掘；高伟勋(1973-)，男，博士，高级工程师，IEICE 会员，主要研究方向：计算机网络、智能信息处理；徐逸凡(1993-)，男，硕士研究生，主要研究方向：计算机网络。

收稿日期： **修回日期：** **E-mail：kevinelstri@foxmail.com**

本文对比分析了不同的相似度计算方法，在词林上大多是基于路径的研究方法。本文将词林编码引入到词语相似度计算中，使用局部敏感哈希算法对词林编码进行处理，同时使用词林的路径结构进行加权，来计算词语之间的相似度。

1 相关工作介绍

在词语相似度计算方法研究中，主要包括英文词语相似度计算和中文词语相似度计算，英文词语相似度的研究主要是通过 WordNet[7]、FrameNet[8]和 MindNet[9]等语义词典，中文词语相似度的研究主要是通过《同义词词林》[10]和《知网》[11]等语义词典，随着 WordNet 的发展，也逐渐产生了中文版 WordNet，包括北京大学 WordNet(BOW)[12]、东北大学 WordNet[13]、台湾大学 WordNet(CWN)[14]和南洋理工大学 WordNet(COW)[15]。

在英文词语相似度计算中，主要是基于 WordNet 进行研究，Abhijit Adhikari[16]等人以信息内容（IC）为基础，对比分析了已有的研究成果，在基于信息内容相似度距离计算的基础上提出了新的信息理论模型，实验对比分析不同的方法结论，实现了词语相似度计算方法的优化。Mohamed Ben Aouicha[17]等人结合 WordNet 和维基词典两个词汇表来对名词词语的语义相似度进行分析，使用信息内容来计算名词的语义因素，并对重复词语进行处理，实现了词语语义相似度算法在不同语料库上的优化。Chongchong Zhao[18]等人针对本体异构性问题，提出了本体概念相似度计算，从本体的特性出发，提出了基于本体语法的编辑距离计算方法，从信息内容的角度，提出了基于信息内容改进的语义相似度算法，对编辑距离和语义算法的加权算法研究，提高了计算精度。

在同义词词林上进行相似度研究最早是田久乐[19]，主要是对同义词的树形结构进行研究，构造了词语相似度公式，实现了同义词词林下的词语相似度计算。但是该方法仅仅从树形结构的分支距离来计算相似度，使得分支较大的两个词的相似度较小，带来了词语相似度的不准确性。Peiying Zhan[20]等人将词林层次结构和知网树形结构进行结合，使用线性加权求和方法来计算词语相似度。Tao Chi[21]提出了一种基于中文词语特性、知网和同义词词林三种算法相结合的词语相似度计算，同时在文本相似度的计算中使用本体模型来表示。线性加权的方法带来了词语相似度计算的准确度的提高，同时也带来了复杂度的提升。LiHong Xu[22]对传统文本特征提取方法进行分析，提出了使用语义向量空间模型来计算词语相似度的方法，使用词林中的相似度为 1 的词语来替换特征词，达到降维的目的。词语相似度算法的研究主要是从同义词词林树形结构进行研究，本文在词林树形结构的基础上提出新的词语相似度算法,在词语相似度计算的准确性上得到很大的提高，编码替换也提高了相似度计算的效率。

通过对已有的研究进行分析，提出了基于路径和语义编码的词语相似度计算方法，使用局部敏感哈希算法对词语编码进行计算，不仅实现了词语相似度的计算，而且降低了词语计算的复杂度。

1.1 局部敏感哈希

局部敏感哈希[23]，主要用于高维数据的近似最近邻快速查找技术，将高维数据映射到一个新的数据空间中，使得具有相同特征的两个原始数据在新的数据空间中，仍然具有相同的特征，而具有不同特征的两个原始数据，映射的结果也是不同的。局部敏感哈希就是在 hash 处理前后其特征保持不变，在进行词语相似度计算的过程中就可以利用局部敏感哈希对词语进行 hash 处理，不仅能够保持词语的特征不变，还能够对词语进行降维，提高相似度计算精度和效率。

1.2 同义词词林

《同义词词林》[10]（简称《词林》）是一部中文词语分类词典，每一个同义词条目都使用同一个编码来表示。本文所使用的是哈尔滨工业大学在《词林》的基础上进一步研究而得到的《同义词词林（扩展版）》。《词林（扩展版）》中包括 17817 个同义词条目，每一个条目都有一个语义编码，由 77492 个词语组成，其中一词多义的词语有 8860 个。

《词林（扩展版）》中使用 6 个类别对中文词语进行分类，分别对应大类、中类、小类、词群、原子词群和词语标记，每一个词语都有一个 8 位语义编码对应，最后一位词语标记有 3 种，分别是“=”“#”“@”，其中“=”表示“相等”“同义”，属于相同意义的词语；“#”表示“不等”“同类”，属于相关词语；“@”表示“自我封闭”“独立”，表示在词典中没有同义词和相关词，图 1 表示同义词词林的树形结构。

在《词林（扩展版）》中，本文主要使用的是词林编码，在词林的编排中，词义相同的词其所具有的编码也是相同的，词义相关的词其所具有的编码也是相关的，在词林的分类中，编码也具有一定的规律，12 个大类使用 A 到 L 来表示，每个大类中的中类又是由 abc 等小写

字母表示，小类编号由十进制来表示，词群、原子词群、词语标记依次使用相应的英文字母和数字来表示。

在语义编码的使用中，使用完整的八位编码作为词语的编码。例如：词语“立即”的语义编码为 “Ka08A01=”，在语义编码“Ka08A01=”下的同义词分别为“立刻 即刻 立时 即时 立地 马上 随即 当即 迅即 当下 登时 顿时 顿然 及时 眼看 当时 应时 应声 旋踵 旋即 二话没说 立马”，这些词都具有相同的语义特征，在文本中使用相同的语义编码替换就能够提高文本之间的相似度。

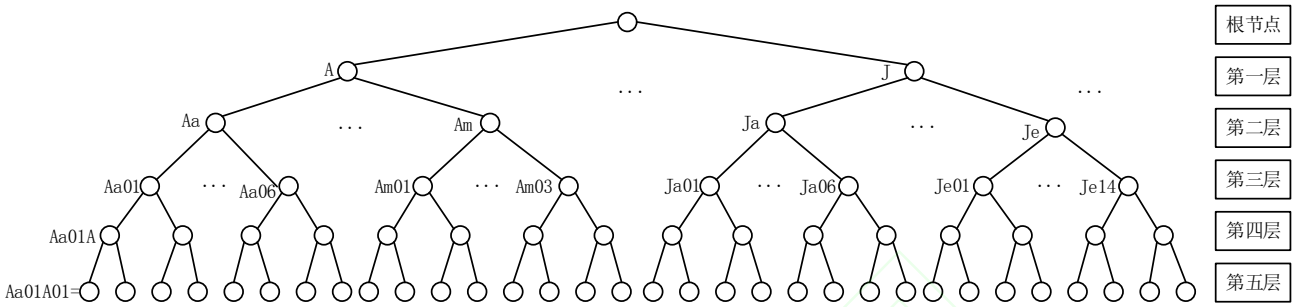

图 1 同义词词林的树形结构

1.3 基于路径的计算方法介绍

Z.Wu[24]等人将 WordNet 词典应用到基于知识的机器翻译领域中，提出了基于 WordNet 结构特点的词语语义相似度计算方法，对于两个词语 w_1 和 w_2，其相似度计算公式为式（1）：

$$ComSim(w_1,w_2)=\frac{2N_3}{N_1+N_2+2N_3} \quad (1)$$

其中，N_1 和 N_2 表示两个词语到最近公共父节点之间的距离，N_3 表示最近公共父节点到根节点之间的距离。

D.Hao[25]等人在使用 WordNet 计算词语相似度的问题上，将两个词语之间的最短距离和最近公共父节点到根节点的距离作为衡量两个词语之间相似度的标准，对于两个词语 w_1 和 w_2，其相似度计算公式为式（2）：

$$Sim(w_1,w_2)=(1-\frac{Dist}{Dist+Dep+\beta})\times(\frac{Dep}{Dist+Dep/2+\alpha}) \quad (2)$$

其中，Dist 表示两个词语之间的最短距离，Dep 表示最近公共父节点到根节点之间的距离。

X.Liu[26]等人在 WordNet 的基础上，使用边权重的方法来对两个词语的相似度进行计算，提出了使用线性函数和指数函数两种方式来计算词语之间的相似度，对于两个词语 w_1 和 w_2，其相似度计算公式为式（3）（4）：

$$S_1(w_1,w_2)=\frac{\alpha d}{\alpha d+\beta l}(0<\alpha,\beta\le 1) \quad (3)$$

$$S_2(w_1,w_2)=\frac{e^{\alpha d}-1}{e^{\alpha d}+e^{\beta l}-2}(0<\alpha,\beta\le 1) \quad (4)$$

其中，S_1 为线性函数计算方法，S_2 为指数函数计算方法，α和β为平滑参数，l 是两个词语之间的最短距离，d 是最近公共父节点到根节点之间的距离。

田久乐[19]对同义词词林进行语义分析，从词林的结构特点出发，结合义项的相似性和相关性来计算词语之间的相似度，对于两个词语 w_1 和 w_2，其相似度计算公式为式（5）：

$$Sim(w_1,w_2)=weight(w_1,w_2)\times\cos(n\times\frac{\pi}{180})(\frac{n-k+1}{n}) \quad (5)$$

其中，weight(w_1,w_2)是相似度计算的权重，表示两个词语 w_1 和 w_2 的最近公共父节点的位置权重，从上到下第 1、2、3、4、5 层的权重依次为 0.1，0.65，0.8，0.9，0.96。$\cos(n\times\frac{\pi}{180})(\frac{n-k+1}{n})$ 是相似度调节参数，n 是分支层的节点总数，k 是两个分支间的距离。

张沪寅[27]在使用知网计算词语相似度时，使用第一基本义原替换概念义项表达式中出现的具体词，使用义原之间的距离来减小义原深度的影响，对于两个词语，其相似度计算公式为式（6）（7）：

$$Sim(S_1,S_2)=\sum_{i=1}^{n}\beta_i\prod_{j=1}^{i}Sim_i(S_1,S_2) \quad (6)$$

$$Sim(w_1,w_2)=\max_{1\le i\le n,1\le j\le m}Sim(S_{1i},S_{2j}) \quad (7)$$

其中，词语 w_1,w_2 分别有 n 和 m 个不同概念;S_{1i} 为 w_1 的第 i 个概念;S_{2j} 为 w_2 的第 j 个概念。

陈宏朝[5]结合词语义项之间的最短路径和最近公共父节点在树结构的深度来计算两个词语之间的相似度，对于词语 w_1 和 w_2，其相似度公式为式（8）（9）（10）：

$$Path(w_1,w_2)=2\times\sum_{i=1}^{n}Weight(i) \quad (8)$$

$$Depth(LCP(w_1,w_2))=\sum_{i=k+1}^{k+m}Weight(i) \quad (9)$$

$$Sim(w_1,w_2)=\frac{Depth(LCP(w_1,w_2))+\alpha}{Depth(LCP(w_1,w_2))+\alpha+Path(w_1,w_2)+\beta} \quad (10)$$

其中，Weight(i)表示在树形结构中，不同层次上的权重，从上到下依次设定为 Weight(i)(1≤i≤5),且满足：0≤Weight(5)≤Weight(4)≤Weight(3)≤Weight(2)≤Weight(1)≤10，Path(w_1,w_2)表示 w_1 和 w_2 到最近公共父节点经过 n 条边，两个词语之间的最短路径距离，Depth(LCP(w_1,w_2))表示 w_1 和 w_2 到第 k 层的最近公共父节点经过

m 条边，两个词之间的深度距离，α 为深度调节参数，β 为路径调节参数。

2 基于 PathCoding 的词语相似度计算方法

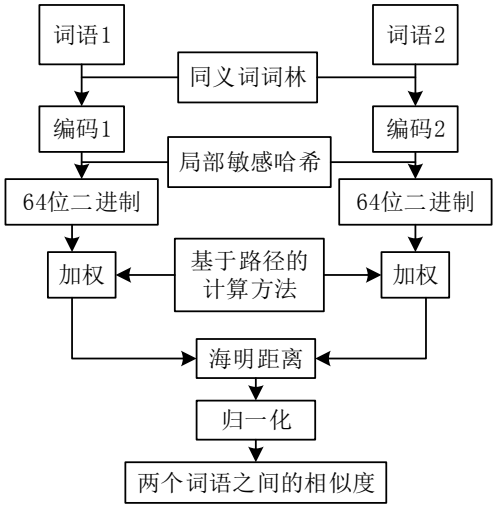

图 2 基于 PathCoding 的词语相似度计算流程

本文提出了使用局部敏感哈希算法对词语编码进行处理，使用词语编码的树形结构特点来计算词语之间的相似度，同时通过对田久乐[19]和陈宏朝[5]等人的方法研究，将词林路径权重表示方法与基于词语编码的计算方法相结合，实现了基于路径与编码相结合的词语相似度计算方法(PathCoding 算法)，图 2 是基于 PathCoding 算法的实现流程。

2.1 基于 PathCoding 的海明距离计算公式

在词林的编排过程中，将词语划分为大类、中类、小类、词群和原子词群五大类，对应到同义词词林的树形结构中，在根节点下就具有了五层结构。在词林中，相同意义的词语使用同一个编码来表示，词林中编码的构建是按照词林树形结构来编排的，因此对于词林编码，同样可以对应到词林五层树形结构中，具体的编码在树形结构上的编排在 1.2 节图 1 所示。其中，词林编码中的第一位对应到大类，第二位对应到中类，第三位和第四位对应到小类，第五位对应到词群，其余对应到原子词群。

本文在词语相似度计算过程中使用局部敏感哈希算法将词林编码转换成固定长度的二进制，然后使用海明距离[28]来计算两个词语之间的相似度。在局部敏感哈希算法的计算中，将词林编码中的字符使用 unicode 编码进行替换，然后按位进行哈希运算，并转换成固定长度的二进制。在图 1 的词林编码树形结构中，同一个父节点下的子节点的编码都是按照顺序进行排列的，其词林编码中的字符 unicode 编码也是相邻的，因此通过对词林编码进行局部敏感哈希运算可以实现词语之间的相似度计算。

在词林相似度研究中，大多数研究者都是从树形结构出发，将词语之间的距离以及到根节点之间的距离作为研究目标。在田久乐的研究中，使用最近公共父节点的位置权重作为计算方式，公式（5）对词林结构进行加权运算，分别赋予 0.1，0.65，0.8，0.9，0.96 的权值。在陈宏朝的研究中，公式（9）对词林结构进行加权，使权重满足：0≤Weight(5)≤Weight(4)≤Weight(3)≤Weight(2)≤Weight(1)≤10，分别赋予 8，6，4，1.5，0.5 的权值。

在本文提出的算法中，将词语编码与词林路径结构相结合，每一层对应不同的类别，每个类别对应到词林编码的不同位，结合局部敏感哈希算法的特点，相似的词语，其哈希值也是相似度的，哈希运算前后不改变词语的特性。在同义词词林树形结构中，从上到下依次为第 1、2、3、4、5 层，分别对应 8 位编码的第 1 位，第 2 位，第 3、4 位，第 5 位和第 6、7、8 位，设置 5 层树形结构边权重依次为α、β、γ、δ、ε，对应到 8 位编码中，weight(1)=α,weight(2)=β,weight(3)=weight(4)=γ,weight(5)=δ,weight(6)=weight(7)=weight(8)=ε。

对陈宏朝提出的公式（9）进行改进，对二进制进行按位加权，同时将海明距离代入公式（9）中，实现基于 PathCoding 的海明距离计算公式（11）：

$$PathHamming(x,y)=\sum_{i=1}^{n}\left[weight\left(\left[\frac{i}{8}\right]+1\right)\times(x[i]\oplus y[i])\right] \quad 0\le i\le 63 \qquad (11)$$

其中，x[i]和 y[i]是二进制编码的第 i 位，⊕表示异或运算，n=64 表示两个哈希结果为固定长度 64 位二进制，$weight\left(\left[\frac{i}{8}\right]+1\right)$ 表示基于路径的权重。

2.2 海明距离归一化

同义词词林中词语的数量是庞大的，在计算海明距离的时候，距离较大的两个词语的海明距离也是很大的，对整个同义词词林进行分析，所有词语两两之间同时进行相似度计算，将得到的海明距离进行归一化，将结果映射到[0-1]之间，归一化公式为式（12）：

$$Norm(x)=\frac{x-minPathHamming}{maxPathHamming-minPathHamming} \qquad (12)$$

其中，minPathHamming=0 表示两个语义相同的语义编码的海明距离，maxPathHamming=165 表示两个语义差距最大的语义编码的海明距离，均为实验验证的结果。

2.3 基于 PathCoding 的词语相似度计算公式

$$Hamming(w_1,w_2)=LSH(code(w_1))[i]\oplus LSH(code(w_2))[i] \quad (13)$$

$$PathHamming(w_1,w_2)=\sum_{i=1}^{n}\left[weight\left(\left[\frac{i}{8}\right]+1\right)\times(Hamming(w_1,w_2))\right] \quad (14)$$

$$Sim(w_1,w_2)=Norm\left(PathHamming(w_1,w_2)\right) \quad (15)$$

式（13）（14）（15）为词语相似度计算公式，其中 LSH(code(w_1))和 LSH(code(w_2))表示词语编码的局部敏感哈希处理，$LSH(code(w_1))[i]\oplus LSH(code(w_2))[i]$ 表示按位计算海明距离，$weight\left(\left[\frac{i}{8}\right]+1\right)$ 表示按位增加路径权重，Norm 表示对海明距离计算结果进行归一化处理，Sim(w_1, w_2)表示词语 w_1 和 w_2 两个词语之间的相似度。

另外，在同义词词林中存在许多一词多义的词语，一词多义词语在进行词语相似度计算时，使用最大相似度算法进行计算，其计算公式为式（16）：

$$w_1:\{code_{11},code_{12},code_{13}...\}$$

$$w_2:\{code_{21},code_{22},code_{23}...\}$$

$$Similarity=\max_{1\le i,1\le j}\{sim(code_{1i},code_{2j})\} \quad (16)$$

其中，w_1 和 w_2 为两个多义词语，code 为词语在同义词词林中的编码，$code_{ij}$ 为一词多义的多个编码。

算法描述：

输入：任意的两个词语 w_1 和 w_2

输出：两个词语的相似度 Sim(w_1,w_2)

1 将同义词词林加载到字典中，转换为同义词字典进行存储：{word:code}，word 为单个词语，code 为词语的编码，将一词多义词典合并，实现多值字典：{word($code_1$,$code_2$…)}

2 计算两个词语 w_1 和 w_2 的语义编码相似度，将两个词语的语义编码进行排列组合构造编码对，即：($code_{1i}$,$code_{2j}$)

3 计算词语编码的 LSH 值，得到长度为 64 位二进制哈希值，即：LSH($code_{1i}$),LSH($code_{2j}$)

4 将权重与二进制位数相对应，实现基于路径权重的海明距离计算，即：PathHamming(LSH_{1i},LSH_{2j})

5 对海明距离进行归一化,使得相似度结果落在[0-1]之间，结果为：Norm(PathHamming(LSH_{1i},LSH_{2j}))

6 使用最大相似度算法对归一化结果进行选择，即：

$$Similarity=\max_{1\le i,1\le j}\{Norm(PathHamming(LSH_{1i},LSH_{2j}))\}$$

3 实验结果及分析

在词语语义相似度计算中，数据集采用国际上普遍采用的 Miller&Charles(MC)[29]发布的 30 对数据集，这个数据集来源于 Rubenstein&Goodenough(RG)[30]发布的 65 对数据集。本文使用两个数据集，一个是 MC 和 RG 共有的 30 对词，定义为集合 M_2，一个是 RG 中单独的 35 对词，定义为集合 M_1.

3.1 边权重的确定

在使用词语编码的局部敏感哈希来计算词语的相似度时，编码位置距离越近的词语，其相似度越高，而编码距离越远的词语，其相似度越低。编码的距离是在局部敏感哈希算法的基础上使用海明距离来计算的，在海明距离的计算中，差异越大的编码，海明距离越大，而差异越小的编码，海明距离越小。

在词林的树形结构中，词语的位置也是决定两个词语相似性的重要因素。例如：一对词语“人类”和“爷们”的编码分别为“Aa01A02=”和“Ab01A02=”，其海明距离为 1，另一对词语“人”和“人类”的编码分别为“Aa01A01=”和“Aa01A02=”，其海明距离也为 1，但两对词语实际的相似度并不相同，后一对词语的相似度明显比前一对相似度要大。对比两对词语编码可以发现，区别主要存在于编码差异的位置不同。前一对词语编码的不同在第二个位置上，后一对词语编码的不同在第七个位置上，对应到词林的树形结构分别在第二层和第五层上，因此为了提高不同词语之间相似性计算的准确性，引入了边权重信息，将词语编码与词林树形结构相结合，实现词语之间的相似性准确性的提高。

使用 M_1 数据集的 35 对词语对本文提出的公式进行训练，在训练的过程中不断调整边权重的取值，使用词语的相似度与人工判定值的皮尔逊相关系数作为比较值，当皮尔逊相关系数最大时，所使用的边权重值就是所确定的值。

在词林树形结构中，从第一层到第五层依次权重为α、β、γ、δ、ε，并满足:0＜ε≤δ≤γ≤β≤α＜10，在海明距离相似度计算方式下，边权重只选择整数。在图 1 中，第一层之间的差异最大，而第五层之间的差异最

小，因此在边权重整数的情况下，第一层权重为α=9，第五层权重为ε=1，图 3 是取不同权值时，所对应的皮尔逊相关系数。

图 3 中纵坐标是皮尔逊相关系数值，横坐标是部分权重取值，并且满足：ε=1，ε≤δ≤γ≤β≤α以及α=9 三个条件。在 165 组实验的对比中，当β=9，γ=4，δ=3 时，皮尔逊相关系数获得最大值 0.8974，而在仅仅使用局部敏感哈希算法计算的皮尔逊相关系数仅有 0.8425.

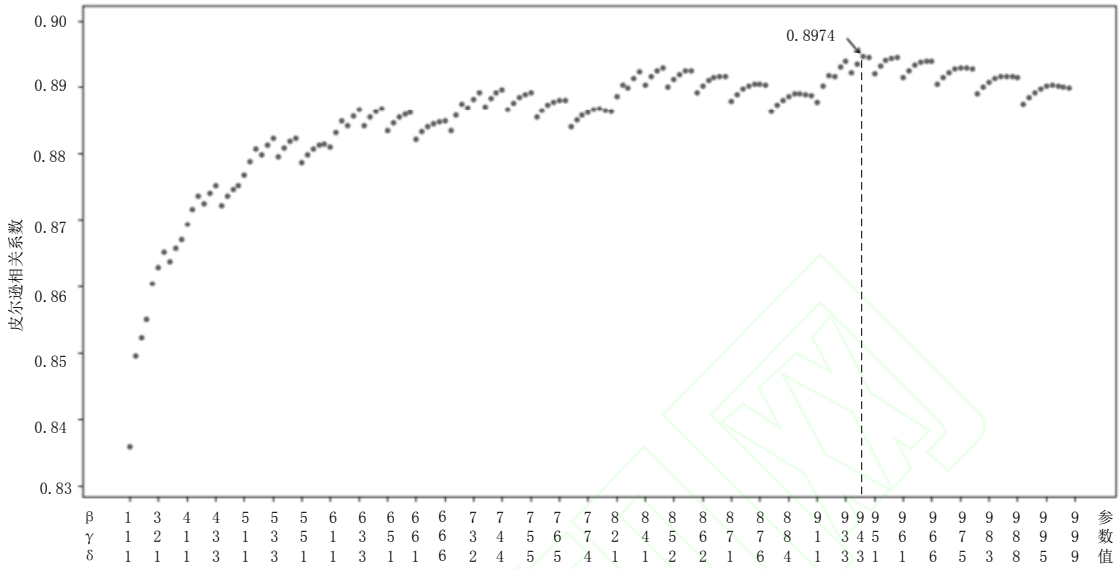

图 3 不同权重对结果的影响

表 1 是在α=9，β=9，γ=4，δ=3，ε=1 时，数据集 M_1 在不同方法下的词语相似度计算结果比较。

表 1 不同方法在 M_1 数据集上的词语相似度比较

编号	词语对	陈宏朝[5]方法	本文方法	RG 人工值
1	水果-火炉	0.2526	0.0286	0.0125
2	署名-海滨	0.0200	0.0114	0.015
3	汽车-巫师	0.0217	0.2229	0.0275
4	高地-火炉	0.0213	0.2629	0.035
5	大笑-器械	0.0198	0.3543	0.045
6	庇护所-水果	0.0217	0.1257	0.0475
7	庇护所-和尚	0.0213	0.2686	0.0975
8	墓地-精神病院	0.0217	0.2171	0.105
9	男孩子-公鸡	0.0217	0.0629	0.11
10	垫子-宝物	0.2348	0.3371	0.1125
11	庇护所-墓地	0.5277	0.2114	0.1975
12	大笑-小伙子	0.0195	0.2229	0.22
13	男孩子-圣人	0.2421	0.2514	0.24
14	汽车-垫子	0.2678	0.2457	0.2425
15	护堤-海滨	0.2479	0.0286	0.2425
16	海滨-航行	0.0200	0.2686	0.305
17	鸟-树林	0.2678	0.1314	0.31
18	火炉-器械	0.5341	0.2286	0.3425
19	鹤-公鸡	0.5503	0.28	0.3525
20	山岗-树林	0.2625	0.26	0.37
21	墓地-坟堆	0.0217	0.2857	0.4225
22	玻璃-珠宝	0.2625	0.3086	0.445
23	魔术师-圣贤	0.2509	0.2857	0.455
24	圣人-巫师	0.2509	0.3257	0.615
25	圣贤-圣人	1.0	1.0	0.6525
26	山岗-斜坡	0.7992	0.6286	0.8225
27	绳索-绳子	1.0	1.0	0.8525
28	玻璃-杯子	0.2625	0.6057	0.8625
29	大笑-微笑	0.9517	0.9943	0.865
30	农奴-奴隶	1.0	1.0	0.865
31	署名-签名	1.0	1.0	0.8975
32	森林-树林	1.0	1.0	0.9125
33	雄鸡-公鸡	1.0	1.0	0.92
34	靠枕-枕头	0.9423	0.9943	0.96
35	墓地-墓园	1.0	1.0	0.97
皮尔逊相关系数		0.8615	0.8974	

3.2 对比实验结果分析

使用数据集 M_2 对实验进行进一步的比较分析，采用不同的公式对数据集计算相似度，获取不同公式下的相似度与 MC 人工值的皮尔逊相关系数。表 2 和表 3 分别是不同的方法在数据集上的实验结果。

表 2 不同方法在 M_2 数据集上的词语相似度比较

编号	词语对	Z．Wu[24]方法	D．Hao[25]方法	X．Liu[26]方法	陈宏朝[5]方法	本文方法	MC 人工值
1	轿车-汽车	0.8	0.9091	0.6452	0.9361	0.9394	0.98

2	宝石-宝物	0.8	0.9091	0.6452	0.9376	0.9091	0.96
3	旅行-远行	0.8	0.9091	0.6452	0.9483	0.9939	0.96
4	男孩子-小伙子	0.8	0.9091	0.6452	0.9346	0.9879	0.94
5	海岸-海滨	0.8	0.9091	0.6452	0.9477	0.9818	0.925
6	庇护所-精神病院	0.8	0.9091	0.6452	0.9501	0.9818	0.9025
7	魔术师-巫师	1.0	0.5098	0.4054	0.8120	0.7697	0.875
8	中午-正午	1.0	1.0	1.0	1.0	1.0	0.855
9	火炉-炉灶	0.8	0.9091	0.6452	0.9454	0.7909	0.7775
10	食物-水果	0.2	0.1176	0.1020	0.3091	0.5152	0.77
11	鸟-公鸡	0.4	0.2737	0.2326	0.7041	0.7879	0.7625
12	鸟-鹤	0.4	0.2737	0.2326	0.7364	0.7636	0.7425
13	工具-器械	0.4	0.2737	0.2326	0.1717	0.2242	0.7375
14	兄弟-和尚	0.2	0.1176	0.1020	0.4505	0.3273	0.705
15	起重机-器械	0.4	0.2737	0.2326	0.1717	0.3212	0.42
16	小伙子-兄弟	0.2	0.1176	0.1020	0.6307	0.3394	0.415
17	旅行-轿车	0.0	0.0087	0.0	0.1	0.303	0.29
18	和尚-圣贤	0.2	0.1176	0.1020	0.5856	0.2364	0.275
19	墓地-林地	0.4	0.2737	0.2326	0.4619	0.0848	0.2375
20	食物-公鸡	0.2	0.1176	0.1020	0.3434	0.1515	0.2225
21	海岸-丘陵	0.4	0.2737	0.2326	0.7922	0.2182	0.2175
22	森林-墓地	0.0	0.0087	0.0	0.1	0.0	0.21
23	岸边-林地	0.2	0.1176	0.1020	0.3434	0.3455	0.1575
24	和尚-奴隶	0.2	0.1176	0.1020	0.3604	0.2545	0.1375
25	海岸-森林	0.2	0.1176	0.1020	0.5495	0.2061	0.105
26	小伙子-巫师	0.2	0.1176	0.1020	0.5406	0.1758	0.105
27	琴弦-微笑	0.0	0.0087	0.0	0.1	0.2424	0.0325
28	玻璃-魔术师	0.0	0.0087	0.0	0.1	0.297	0.0275
29	中午-绳子	0.0	0.0087	0.0	0.1	0.2848	0.02
30	公鸡-远行	0.0	0.0087	0.0	0.1	0.297	0.02

表 3 不同方法与 MC 人工值的皮尔逊相关系数

研究方法	相似度方法	语义词典	皮尔逊系数
Z.Wu[24]	基于深度与路径	WordNet	0.7464
D.Hao[25]	基于深度与路径	WordNet	0.8161
X.Liu[26]	基于深度与路径	WordNet	0.8018
Z.Wu[24]	基于深度与路径	《同义词词林》	0.8457
D.Hao[25]	基于深度与路径	《同义词词林》	0.8252
X.Liu[26]	基于深度与路径	《同义词词林》	0.8086
陈宏朝[5]	基于深度与路径	《同义词词林》	0.8560
本文方法	基于路径与语义编码	《同义词词林》	0.8668

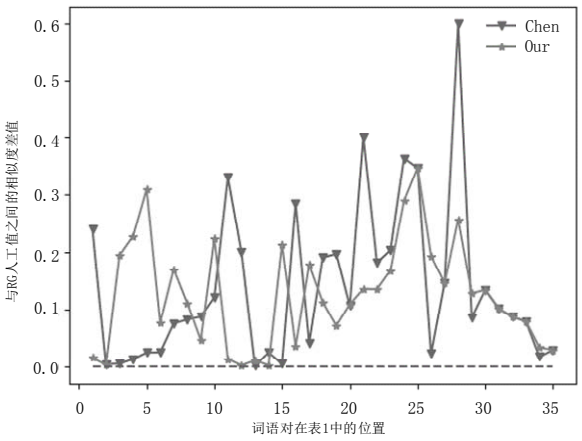

图 4 本文方法与陈宏朝方法的比较(M_1)

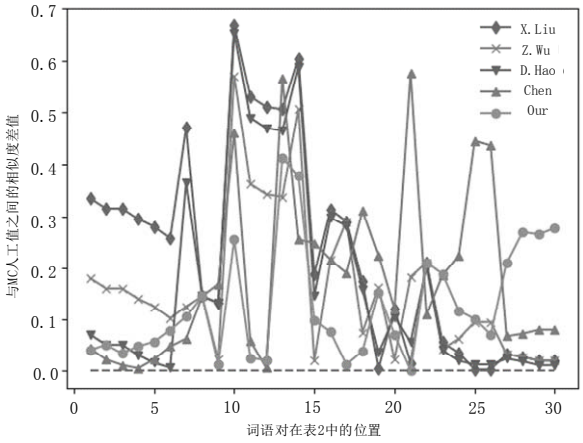

图 5 本文方法与其他四种方法的比较(M_2)

本文在使用同义词词林进行词语相似度计算的过程中提出了不同的方法。本文的方法主要是从词林语义编码出发，针对语义编码的编排具有树形结构这一特点，使用语义编码的相似度来计算词语之间的相似度；同时，将基于路径的方法与语义编码相结合，将不同层次的加权处理应用到语义编码处理中，实现了基于路径和语义编码的词语语义相似度计算方法研究。

从表 1 的实验分析中可以看出：本文提出的方法在相似度计算中取得了很好的效果，且与 RG 人工判定值的皮尔逊相关系数达到了 0.8974。

从表 2 和表 3 的对比实验中可以看出，文本提出的方法，区别于以往研究者基于深度和路径的方法，同时本文方法在与 MC 人工值的皮尔逊相关系数中也达到了 0.8668，在不同方法对比中也具有一定的优势。

图 4 和图 5 更清晰的展现出不同算法在词语相似度计算中的差异，图中横坐标表示词语对编号，纵坐标表示每一对词语在相似度算法下的计算结果到对应人工值之间的距离，距离越小，表示词语相似度准确度越高。图 4 是陈宏朝方法和本文方法在 M_1 数据集上的比较结果，图中显示出本文算法在词语相似度计算结果上距离人工值更近，表明本文的算法在相似度算法计算上得到了提升。图 5 是本文算法与其他四种算法在 M_2 数据集上的比较结果，图中显示文本算法仍具有很好的相似度值。整体分析表明，本文算法在词语相似度计算中具有更高的准确性。

4 结束语

本文提出了一种新的词语相似度计算方法，与传统基于统计的计算方法不同，该方法利用同义词词林的词林编码，使用局部敏感哈希算法将词语信息保存到二进制中，并利用词林的路径结构对词林编码进行加权，使用海明距离来计算词语之间的相似度。该方法充分考虑了词语之间的语义关系，从词语本身的语义信息和词语之间的语义信息两方面来提高词语之间的相似度。在两个标准数据集上对不同方法进行对比实验，实验表明本文提出的方法能够有效提高词语之间的相似度。

参考文献

[1] 王小林，陆骆勇，邰伟鹏.基于信息熵的新的词语相似度算法研究[J].计算机技术与发展,2015,25(9):119-122

[2] 吕亚伟，李 芳，戴龙龙.基于 LDA 的中文词语相似度计算[C].北京化工大学学报,2016,43(5):79-83

[3] Natalia Loukachevitch，Aleksey Alekseev.Use of neighbor sentence co-occurrence to improve word semantic similarity detection[C].International FRUCT Conference on Intelligence, Social Media and Web (ISMW FRUCT),2016,58(7):1-7

[4] M engjia Fan，Yangsen Zhang，Jiayuan Li.Word similarity computation based on HowNet[C].2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD),2015,38(7):1487-1492

[5] 陈宏朝，李飞，朱新华等. 基于路径与深度的同义词词林词语相似度计算[J].中文信息学报,2016,30(9):80-88

[6] Dean J. Jones，Gunjan Mansingh. Not just dissimilar, but opposite:An algorithm for measuring similarity and oppositeness between words[C].International Conference on Data Science and Engineering (ICDSE),2016:1-6

[7] G A Miller，C Fellbaum. Semantic network of English[M]. B.Levin,lexical&conceptual semantics,Amsterdam: Elsevier Science Publisher,1991

[8] C Baker. The Berkeley FrameNet project[C] Proceedings of the COLING-ACL,Montreal,Cannda,1998:86-90

[9] S Richardson,W B Dolan. MindNet:Acquiring and structuring semantic information from text[C]. Proceedings of COLING-ACL,Quebec,Canada,1998:1098-1102

[10] 梅家驹. 同义词词林[M]. 上海：上海辞书出版社，1983.

[11] 知网，HowNet Knowledge Database[EB/OL]. http://www.keenage.com/

[12] Chu-Ren Huang, Ru-Yng Chang, Hsiang-Pin Lee. Sinica BOW (Bilingual Ontological Wordnet):Integration of Bilingual WordNet and SUMO[C]. International Conference on Natural Language Processing and Knowledge Engineering,2003,10:825-826

[13] 张 俐，李晶皎等.中文 WordNet 的研究及实现[J].东北大学学报,2003(4):327-329

[14] Churen Huang and Shukai Hsieh. Chinese Wordnet:Design,Implementation,and Application of an Infrastructure for Cross-Lingual Knowledge Processing[C].Journal of Chinese Information Processing,2010,2

[15] Shan Wang, Francis Bond. Building the Chinese Open Wordnet (COW): Starting from Core Cilinsets[C].International Joint Conference on Natural Language Processing,2013:10-18

[16] Abhijit Adhikari,Shivang Singh,Animesh Dutta,Biswanath Dutta. A Novel Information Theoretic Approach for Finding Semantic Similarity in WordNet[C].2015 IEEE Region 10 Conference,2015,12(10):1-6

[17] Mohamed Ben Aouicha,Mohamed Ali. G2WS:Gloss-based WordNet and Wiktionary semantic Similarity measure[C]. 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications,2015,12(10):1-7

[18] Chongchong Zhao，Aonan Cai.The Similarity Calculation of Concept Names[C].International Conference on Cloud Computing and Internet of Things,2016:93-96

[19] 田久乐，赵蔚. 基于同义词词林的词语相似度计算方法[J].吉林大学学报(信息科学版),2010,06:602-608

[20] Peiying Zhang，Zhanshan Zhang，Weishan Zhang.An approach of semantic similarity by combining HowNet and Cilin[C]. 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber,Physical and Social Computing,2013:1638-1643

[21] Tao Chi,Hanshi Wang,Lizhen Liu,Wei Song,Chao Du. Text Similarity Calculation Method based on Ontology Model[C]. 2014 International Conference on Cloud Computing and Internet of Things,2014,13(11):213-217

[22] LiHong Xu,ShuTao Sun,Qi Wang.Text Similarity Algorithm Based on Semantic Vector Space Model[C].2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS),2016,10:1-4

[23] Indyk P, Motwani R. Approximate nearest neighbor-towards removing the curse of dimensionality[C]. Proceeding of the 30th Symposium on Theory of Computing, 1998:604-613

[24] Z.Wu，M.Palmer. Verbs semantics and lexical se-lection[C]. Proceedings of the 32nd Annual Meeti-ngon Association for Computational Linguistics， ACL94，Association for Computational Linguistics，Stroudsburg，PA，USA,1994:133-138

[25] D.Hao，W.Guo，T.Peng. An approach for calcula-ting semantic similarity between words using wor-dnet[C]. Proceedings of the second international Conference on Digital Manufacturing and Automati-on，Zhangjiajic，China,2011:177-180.

[26] X. Liu，Y. Ghou，R. Zheng. Measuring semantic si-milarity in WordNet[C]. Proceedings of the Sixth international Conference on Machine Learningand Cybernetics，Hong Kong，China,2007:3431-3135.

[27] 张沪寅，刘道波，温春艳.基于知网的词语语义相似度改进算法研究[J]. 计算机工程,2015,2(41):151-156

[28] Haifeng Hu，Liang Zhang，and Jianshen Wu. HammingDistance based Approximate Similarity Text Search Algorithm[C]. 7th International Conference on Advanced Computational Intelligence,2015,3:27-29

[29] G.A.Miller，W.G.Charles. Contextual corralates of semantic similarity[J]. Language and Cognitive Processes,1991,6:1-28

[30] H.Rubenstein，J.B.Goodenough. Contextual correlates of Cilinunymy[C]. Proceedings of the ACM8(10),1965:627-633

王松松：
Mobile：15800402042
Email：kevinelstri@foxmail.com
高伟勋：
Mobile：18016061920
Email：gwx@shnu.edu.cn
徐逸凡：
Mobile：17621378703
Email：437320541@qq.com