Computer Science and Application 计算机科学与应用, 2013, 3, 232-238
http://dx.doi.org/10.12677/csa.2013.34040 Published Online July 2013 (http://www.hanspub.org/journal/csa.html)

The Study and Implementation of Web User Mining System Based on the Similarity of Words*

Chengxia Liu[1,2], Feiying Wu[2]

[1]Computer School, Beijing University of Posts and Telecommunications, Beijing
[2]Computer School, Beijing Information and Technology University, Beijing
Email: cecilia7812@163.com

Received: May 13th, 2013; revised: May 27th, 2013; accepted: Jun. 5th, 2013

Abstract: Nowadays, as web mining is extremely prevalent, it is easy to collect huge amounts of data but to figure out which materials are useful to analyze after de-noising is more important. This article discusses how to use the result of user's searching keywords clustering as the label of the client for operational analysts to refer to. The similarity between isolated words is calculated by turning the word semantic distance based on world knowledge or classification system. Then the similarity between clients (keyword sets) is defined as the Euclidean distance of a similarity matrix constituted by the similarities between keyword sets which determined by word frequency and word weight. The "depmix" package which based on the Hidden Markov Model in "R" software is used as the clustering algorithm and the user clustering result is displayed at last using the real data of the users of a search engine.

Keywords: The Similarity of Words; The Similarity Between Clients (Keyword Sets); User Clustering

基于关键词相似度的 Web 用户挖掘研究与实现*

刘城霞[1,2]，吴菲滢[2]

[1]北京邮电大学计算机学院，北京
[2]北京信息科技大学计算机学院，北京
Email: cecilia7812@163.com

收稿日期：2013 年 5 月 13 日；修回日期：2013 年 5 月 27 日；录用日期：2013 年 6 月 5 日

摘　要：在 Web 挖掘极度盛行的今天，收集大量网络数据已经不是问题，而如何在海量数据中抽取去噪后的有用数据成为要解决的关键问题。本文研究将网站用户的搜索关键词分析聚类，作为用户的兴趣、爱好标签，以供运营分析人员参考。文中根据世界知识或分类体系计算词语语义距离后转化为词语相似度的方法，将词语间距离依据词频、词权重等因子加工计算出关键词集合间相似度矩阵后，用欧式距离表示其关键字集的相似度；之后聚类算法利用现有 R 软件中开源算法包——基于隐马尔科夫模型的 depmix 算法包进行的用户聚类算法。最终用某搜索引擎用户的真实数据，经过数据去噪后所得实验数据进行聚类，并于前台展示聚类及用户周边相关结果。

关键词：词语相似度；关键词集合相似度；用户聚类

1. 引言

近些年来互联网行业飞速发展，机构、团体和个人越来越多地依赖互联网发布信息、查找信息，这成就了互联网上的海量数据，但同时这些无结构的、动

*资助信息：北京市人才强教计划——骨干教师(PHR201008428)，北京市教委科技发展计划项目(KM201110772013)资助。

态的 Web 页面的复杂程度远远超过了文本文档，所以人们要想找到自己想要的数据依然犹如大海捞针一般。Web 挖掘是将传统的数据挖掘技术和 Web 结合起来，就能解决这些问题。

如果说 Web 使用挖掘是通过挖掘访问者在网站上留下的痕迹来获取有用的信息，那么 Web 用户挖掘则是要寻找 Web 用户的根源。通过对 Web 用户信息的统计分析，能够帮助运营商以较低成本获得准确度较高的客户兴趣倾向、个性化需求以及新业务发展趋势等信息。

本文是基于搜索引擎用户进行的研究，致力于根据不同用户的不同搜索关键词，为用户打上相应的个性标签，以便运营分析人员对用户的兴趣、爱好有更精确的方向上的把握，并且做出更好的信息推送及搜索引擎优化。

2. 关键词间相似度算法

2.1. 相似度计算的基本概念

相似度计算中有两个主要的概念为：“概念”与“义原”。“概念”是用来描述词汇的语义的，一个词语可以用一个或多个概念来表示。这种描述方法叫做用“知识表示语言”来描述词语的语义，而这种用来构成“知识表示语言”的“词汇”就叫做“义原”。

与一般的语义词典不同，“知网”中的概念层次树并不仅仅是一个归结了所有 “概念”的概念层次体系树，而是将每一个“概念”用一系列的“义原”来描述清晰。

“知网”将义原分为以下几个大类[1]：

1) Event|事件

2) entity|实体

3) attribute|属性值

4) aValue|属性值

5) quantity|数量

6) qValue|数量值

7) SecondaryFeature|次要特征

8) syntax|语法

9) EventRole|动态角色

10) EventFeatures|动态属性

这些义原大致被归为 3 组：

第 1 组，第 1~7 类义原，称之为“基本义原”，用来描述单个概念的语义特征；

第 2 组，第 8 类义原，称之为“语法义原”，用于描述词语的语法特征，主要是词性的表达；

第 3 组，第 9、10 类义原，称之为“关系义原”，用于描述概念和概念之间的关系。

2.2. 词语相似度算法分析

对于两个词语 W_1 和 W_2，如果 W_1 有 n 个概念：S_{11}，S_{12}，…，S_{1n}，W_2 有 m 个概念：S_{21}，S_{22}，…，S_{2m}，把两词语间的相似度问题定义为两组概念间的相似度问题。

1) 义原相似度的计算

义原间的相似度计算是概念相似度的计算的基础，因为所有概念最终都会归结于用义原来表示。本文采用通过语义距离来计算义原结点间相似度的办法，即假设两个义原在此层次体系中的路径距离为 d，则这两个义原间的语义距离[2]可由

$$\mathrm{Sim}(W_1,W_2)=\frac{\alpha}{Dis(W_1,W_2)+\alpha} \tag{1}$$

计算得来，具体化为：

$$\mathrm{Sim}(P_1,P_2)=\frac{\alpha}{d+\alpha} \tag{2}$$

其中 α 为可调节参数，P_1 和 P_2 表示两个不同义原，d 是 P_1 和 P_2 在层次树中的路径长度，记为一正整数。

2) 虚词概念的相似度的计算

因为在“知网”的知识描述语言中，虚词概念只用“{句法义原}”或“{关系义原}”这两种方式进行描述，所以计算虚词概念的相似度就等价于计算其对应的句法义原或关系义原之间的相似度。

3) 实词概念的相似度的计算

本文采用的相似度计算方法是通过部分相似度的合成来代替整体相似度。首先建立两个整体中的各个部分之间一一对应的关系，随后计算各个配对间的相似度，加权求和。若某一部分的对应为空时其相似度定义为一个比较小的常数 δ，和具体词与义原的相似度定义为同一级别。

4) 特征结构和集合的相似度计算

①特征结构的相似度计算

特征的定义是一个“属性：值”对，特征结构就是“属性：值”对的集合。在特征结构中，每个“特

征”的“属性”是唯一的。将特征结构的相似度转化为各个特征之间的相似度的均值。两个特征的相似度就等价于其“值”的相似度。

②集合的相似度计算

两个集合的相似度计算的简单模型如下：

A．一个集合与它本身的相似度为 1；

B．假设两个集合都有 n 个元素，其中 m(m < n)个元素相同，又假设两个元素的相似度只能是 0(不同)或 1(相同)，那么这两个集合的相似度应该是 m/n。

而两个集合各个元素之间的一一对应关系如下：

a) 生成两个集合中所有元素两两间相似度的矩阵；

b) 在相似度矩阵中挑出最大相似度后，将其对应的两个元素关联为一对；

c) 从相似度矩阵中取出这些已建立关联对的元素及其相互间的相似度；

d) 重复第 b、c 两步，直至某一集合中所有元素都已与另一集合中某元素生成关联对；

e) 将没有建立起对应关系的元素与空元素对应。

根据上述方法建立起两个集合元素的一一对应关系后，集合的相似度等于其元素对的相似度的算术平均。

5) 实词概念相似度的计算

对一个实词的描述可以表示为一个特征结构，该特征结构含有以下四个特征：

第一基本义原描述：即实词最重要的一个基本义原，两个概念相对应的这一部分的相似度记为：$Sim_1(S_1, S_2)$；

其它基本义原描述：即除第一基本义原以外所有基本义原的集合，这一部分的相似度记为：$Sim_2(S_1, S_2)$；

关系义原描述：即语义表达式中所有的关系义原，表示为一个特征结构，这一部分的相似度记为：$Sim_3(S_1, S_2)$；

关系符号描述：即所有的关系符号描述式，也表示为一个特征结构，这一部分的相似度记为：$Sim_4(S_1, S_2)$。

考虑以下情况：当 Sim_1 非常小，但 Sim_3 或者 Sim_4 却相对较大时，若直接计算相似度平均值会导致整体的相似度仍然较大，但不合理的现象。因此很多改进的词语相似度算法也就应运而生[4-6]，本文改进实词间相似度算法后公式如下：

$$\mathrm{Sim}(S_1, S_2) = \sum_{i=1}^{4} \beta_i \prod_{j=1}^{i} \mathrm{Sim}_j(S_1, S_2) \tag{3}$$

其中，$\beta_i(1 \le i \le 4)$是可调节的各部分相似度权重参数，且有：$\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1, \beta_1 \ge \beta_2 \ge \beta_3 \ge \beta_4$。$\beta$ 反映了 Sim_1 到 Sim_4 对于总体相似度所起到的作用依次递减。其意义在于，主要部分的相似度值对于次要部分的相似度值起到制约作用，即若主要部分相似度比较低，那么次要部分的相似度对于整体相似度所起到的作用也要降低，且可以保证一个词和它本身的相似度为 1。

举例：两个实词“鸟”、“牛”的相似度计算。

查阅词库得到以下概念的解释：

牛 N——value|属性值，behavior|举止，stubborn|倔，undesired|莠

牛 N——character|文字，surname|姓，human|人，ProperName|专

牛 N——livestock|牲畜

鸟 N——bird|禽

可以看到，“牛”共有表示性格、姓氏以及动物三个解释，而“鸟”只有禽畜一种解释。下面来具体分析这两个词的相似度：

a) 第一基本义原描述

“牛”的三组基本义原为：“属性值”，“文字”，“牲畜”；“鸟”的基本义原为：“禽”。

根据查询词库可得：

“属性值”没有相同根义原，默认最大距离 20，相似度 Sim1(属性值) = 0.074，

“文字” 义原距离为 9，相似度 Sim1(文字) = 0.151，

“牲畜” 义原距离为 2，相似度 Sim1(牲畜) = 0.44。

b) 其它基本义原描述

“牛”的三组其它基本义原:“举止”，“倔”，“莠”；“姓”，“人”，“专”；空。

“鸟”的其它基本义原为空。

因此，由公式(sum + δ*(m − n))/m，其中 sum 为两组其它基本义原中依次挑出相似度最大的义原对直至一组基本义原被挑空的相似度之和(由于“鸟”没有其他基本义原，因此 sum 为 0)、m 为两组其它基本义原中较多义原一组的义原个数、n 为较少一组的义

原个数，可得这两个关键词的其它基本义原的相似度 Sim_2 如表 1：

c) 关系义原描述：此处两词没有关系义原的解释，因此 Sim3 默认为 1.0。

d) 关系符号描述：此处两词没有关系义原的解释，因此 Sim4 默认为 1.0。

最终“牛”与“鸟”的相似度描述

$$\begin{aligned}Sim &= \beta_1 * Sim_1 + \beta_2 * Sim_1 * Sim_2 \\ &+ \beta_3 * Sim_1 * Sim_2 * Sim_3 \\ &+ \beta_4 * Sim_1 * Sim_2 * Sim_3 * Sim_4\end{aligned}$$

取 $\beta_1 = 0.5$，$\beta_2 = 0.2$，$\beta_3 = 0.17$，$\beta_4 = 0.13$ 得

$$\begin{aligned}&Sim_{(属性值组)} = 0.5*0.074 + 0.2*0.074*0.2 \\ &+0.17*0.074*0.2*1 + 0.13*0.074*0.2*1*1 = 0.044\end{aligned}$$

$$\begin{aligned}&Sim_{(文字组)} = 0.5*0.151 + 0.2*0.151*0.2 \\ &+0.17*0.151*0.2*1 + 0.13*0.151*0.2*1*1 = 0.091\end{aligned}$$

$$\begin{aligned}&Sim_{(牲畜组)} = 0.5*0.44 + 0.2*0.44*1 \\ &+0.17*0.44*1*1 + 0.13*0.44*1*1*1 = 0.44\end{aligned}$$

取最大值，得到“鸟”与“猪”的相似度为 Sim = 0.44。

2.3. 用户相似度算法

在确定独立关键词间相似度算法后，便可以得到两用户关键词集合的相似度矩阵。传统的文本相似度算法是基于关键词向量的算法[7,8]，通常可以通过文献标题、关键词和摘要合并形成特征向量空间来提高文献表示的精度[9]，但这增加了计算的维度。而且传统的文本聚类方法都是将文档表示成关键词特征空间中的一个向量，其取值非 0 即 1，没有考虑关键词部分的相似性。本文基于以上两点，采用了基于关键词加权的文献相似度计算方法，在不增加特征向量空间维数的情况下，考虑了关键词之间的部分相似性，提高了相似度计算的精度。

要求得各用户之间距离矩阵，首先要生成每个用户与每个关键词之间的用户－关键词相似度矩阵。用户－关键词相似度矩阵并不仅仅是将用户每个搜索关键词与关键词集合中关键词间的相似度计算算术平均值罗列出来，而是加入词频、权重等因子的制衡的加权平均相似度。其中：某关键词的词频即为该关键词在所有的关键词总集合中出现的频次；某关键词的权重为该关键词与所有的关键词总集合中的每一个关键词相似度的最大值。那么最终用户(关键词集合)与关键词间的相似度就由以下算法确定：

对于每一个关键词－用户搜索关键词对，其相似度、用户搜索关键词词频以及用户搜索关键词的乘积进行求和后比上每一对后两者的乘积求和所得的商便是最终的用户－关键词的加权平均相似度。

n 个关键词集合(用户)与 m 个特征关键词首先构成了用户-关键词矩阵(n × m)，定义为：

$$\begin{bmatrix} Q'_{11} & \cdots & Q'_{1m} \\ \vdots & \ddots & \vdots \\ Q'_{n1} & \cdots & Q'_{nm} \end{bmatrix} \tag{5}$$

其中，Q'_{ij} 定义如下：

$$Q'_{ij} = \frac{\sum_{t=1}^{\beta}\left(Q\left(kw_{it}, kw_j\right) \times T\left(kw_{it}\right) \times W\left(kw_{it}\right)\right)}{\sum_{t=1}^{\beta}\left(T\left(kw_{it}\right) \times W\left(kw_{it}\right)\right)} \tag{6}$$

其中，$Q(kw_{it}, kw_j)$表示关键词集合(用户)i 的第 t 个关键词与关键词集合 W 中第 j 个关键词的相似度；$T(kw_{it})$是关键词集合 i 的第 t 个关键词在关键词集合中出现的频次；$W(kw_{it})$是关键词集合(用户)i 的第 t 个关键词的权重。由于 Q 的取值在[0,1]之间，显然 $Q'_{ij} \in [0,1]$。Q'_{ij} 的定义考虑了将关键词综合加权来表示关键词集合(用户)在特征空间中的取值。

使用欧式距离表示关键词与关键词集合(用户)的相似度。即两用户间距离表示为两用户与每个关键词相似度的差平方的和的开平方，相似度 S 定义公式如下：

$$S = \sqrt{\sum_{k=1}^{m}\left(Q'_{ik} - Q'_{jk}\right)^2} \tag{7}$$

由以上定义可得出关键词与用户的相似度(S)，并可以以此为依据聚类出与关键词相关的用户群。

Table 1. Base similarity of bird and cow
表1. “鸟”与“牛”其它基本义原相似度

其他基本义原似度		牛		
		属性值3个	文字3个	牲畜无
鸟	无	(0.2*3)/3 = 0.2	(0.2*3)/3 = 0.2	默认为1

3. 相似度算法设计

3.1. 词语间相似度算法

词语间相似度计算算法描述：

step1：若两词语中有一个词语未收录到词库中，则默认相似度为 0；

step2：分别计算两词语相关联的所有解释概念(义原/基本词)的相似度，取其最大值为两词语相似度；

step3：计算两个解释概念的相似度时，若两概念一为实词一为虚词，则相似度默认为 0；若两概念均为虚词，则返回其虚词义原间相似度；若两概念均为实词，则分别计算其第一基本义原相似度、其他基本义原相似度、关系义原相似度以及关系符号相似度，依照(2)计算其相似度并返回。

3.2. 用户间相似度算法

算法涉及到关键词集合(KeywordVector)、关键词与关键词集合相似度(Similarity_V_W)和关键词集合间相似度(VectorSimilarity)三个类，其关系如下：

一个 KeywordVector 实例提供所有关键词的详细信息及个关键词间相似度矩阵的 map；一个 Similarity_V_W 实例在初始化时，便根据 clients 文件计算出所有不重复关键词向量及个个关键词的词频、id，并查询 KeywordVector 实例中的相似度矩阵生成所有用户与关键词间的相似度矩阵。而一个 VectorSimilarity 实例通过查询一个 Similarity_V_W 的实例，提供了计算用户间相似度的计算方法。

用户－用户相似度算法描述：

依次从关键词集合中取出一个关键词

step1：根用户 id 以及关键词 id 分别计算两用户与同一关键词在“用户－关键词”相似度矩阵中的 id；

setp2：根据这两个 id 查询相似度矩阵得到两用户与同一关键词的相似度，并求其差平方；

step3：重复第 step1-step3 步直至关键词集合中所有关键词已使用过，过程中将相似度的差平方累加；

step4：将最后的累加和开平方后得到的就是两用户间的相似度。

```
public double get_sim_VV(int client1_id, int client2_id)
{int i;
Double sum = 0.0;
//将两关键词集合与每一个关键词的相似度的差平方求和后再开平方
for(i = 0;i<关键词个数;i++){
//计算用户与关键词相似度在相似度矩阵中的位置，即其 id
int key1 = 取得 client1_id 的第 i 个关键词；
int key2 = 取得 client2_id 的第 i 个关键词；
Double temp_differ = 两用户与同一关键词的相似度差；
sum += temp_differ*temp_differ;//计算相似度，求差平方求和}
return Math.sqrt(sum);//求和后的差平方开平方}
```

用户－关键词相似度算法的算法描述：

step1：依次从关键词集合中取出一个关键词

step2：根据 id 分别计算两用户与同一关键词在“用户－关键词”相似度矩阵中的 id；

step3：根据这两个 id 查询相似度矩阵得到两用户与同一关键词的相似度，并求其差平方；

step4：重复第 step1-step3 步直至关键词集合中所有关键词已使用过，将相似度的差平方累加；

step5：将最后的累加和开平方后得到的就是两用户间的相似度。

```
private Double getVWsim(int[] vcti, int j)//vcti 为关键字集合
{Double sum_sim_t_w, sum_t_w;
sum_sim_t_w = sum("stw",vcti,j);
//计算
```

$$Q'_{ij}=\frac{\sum_{t=1}^{\beta}\left(Q\left(kw_{it},kw_j\right)\times T\left(kw_{it}\right)\times W\left(kw_{it}\right)\right)}{\sum_{t=1}^{\beta}\left(T\left(kw_{it}\right)\times W\left(kw_{it}\right)\right)}$$ ()的分母

```
sum_t_w = sum("tw",vcti,j);
//计算
```

$$Q'_{ij}=\frac{\sum_{t=1}^{\beta}\left(Q\left(kw_{it},kw_j\right)\times T\left(kw_{it}\right)\times W\left(kw_{it}\right)\right)}{\sum_{t=1}^{\beta}\left(T\left(kw_{it}\right)\times W\left(kw_{it}\right)\right)}$$ ()的分子

```
Double vwSim = sum_sim_t_w/sum_t_w;//计算关键词－用户相似度
return vwSim;}
```

4. 结果分析

4.1. 词语相似度结果分析

两组词语相似度对比如表 2：

可以看到，绝大部分结果还是比较合理的：

Table 2. Result of word similarity
表 2. 词语相似度结果分析

词语相似度	鸟	人	殡葬	服务	腐败
主题	0.0429	0.5795	0.0429	0.0664	0.0429
器官	0.1493	0.1667	0.1404	0.0664	0.0429
捐献	0.0741	0.0741	0.0429	0.1379	0.0444
春天	0.4444	0.0444	0.0429	0.0740	0.0444

a) “主题”与另一组中关键词的“人”相似度最大；

b) “器官”与另一组中关键词的“人”相似度最大；

c) “捐献”与另一组中关键词的“服务”相似度最大；

d) “春天”与另一组中关键词的“鸟”相似度最大；

也有部分结果不够合理，例如“捐献”与“人”、“器官”与“人”的相似度都偏低，原因是“器官”、“捐献”只有单一概念解释，因此计算相似度时含义分析不够丰满。这也从一个侧面反映了某些定义不合理或不一致之处，需要进一步改进。

4.2. 用户相似度结果分析

用户关键词集合如下：

1 {“坚守”，“岗位”，“默默”，“张歆艺”}

2 {“魅力”，“非凡”，“四季”，“女装”}

3 {“北京”，“企业”，“信用”，“网络”}

4 {“江泽民”，“九寨沟”，“互联网”}

5 {“中国”，“邮政”，“储蓄”}

6 {“银行”，“中国矿业大学”，“徐州”}

由于此处计算的是用户间欧氏距离，距离与其相似度成反比，则没有绝对最大距离，只有当距离为 0.0 是其相似度确定为 1，其距离对比如表 3。

可以得到：3、4 用户相似度较高比较合理，其中 3、4 用户都关心网络、地名；而 1、5 用户相似度较高则无理可循；同样含有地名“徐州”的用户 6 与用户 3 相似度也较高，他们还共同关心企业方面搜索内容，反观用户 6 与用户 4 的相似度就相对较低，虽然同样都包含地名，但两者间并无其他共性；用户 2 与所有其他用户的相似度中，与用户 1 距离最小也比较合理。

当然也有用户相似度结果不太合理的，原因主要是用户的搜索关键词本身比较偏僻，导致计算词语相似度是已经不合理或者该用户的搜索关键词词频较低及该用户的搜索关键词在关键词集合中的权重较低。

4.3. 用户聚类结果分析

由于用户在进行搜索行为时，会主观根据某一词条搜索结果的满意程度决定下一搜索关键词的内容。同时随着时间的推移社会热点的变动，搜索关键词的主流内容也会随之变动。因此可以认为用户的搜索关键词与用户的兴趣[10]及时间有着密切联系。

在用户聚类过程中，实验采用了 R 软件的开源软件包中的隐马尔科夫模型用于聚类。通过聚类，测试数据中 84 位用户共聚类出 24 类用户群，以“电影”主题相关的聚类结果结果为例分析如表 4：

可以看到前两位用户与“电影”主题有较大联系，而后 5 位用户与“电影”主题关联渐弱。原因是挑选主题相关用户类时是挑选与该主题相似度最大的用户的所在类，所以并不是该类所有用户都与该主题相

Table 3. Result of user similarity
表 3. 用户相似度结果分析

	1	2	3	4	5	6
1	/	0.6248	0.6893	0.7352	0.5863	1.0148
2	/	/	0.8427	0.9438	0.8142	1.2286
3	/	/	/	0.5975	0.8331	0.6917
4	/	/	/	/	0.8488	0.9522
5	/	/	/	/	/	1.0764

Table 4. Cluster of movie title
表 4. “电影”主题相关的聚类结果

	id	注册方式	关键词标签
用户1	1275	手机注册	ipad，明天，新闻，iteye
用户2	1231	邮箱注册	2012，娱乐，音乐，李双江
用户3	1100	手机注册	小升初，怪圈，安阳，教育局
用户4	1304	邮箱注册	洛阳，众，托，货运，有限公司，物流，公司
用户5	1616	邮箱注册	赛，成功
用户6	1631	手机注册	黄岩岛，油价，下调，盘古，搜索，百度
用户7	1224	手机注册	我，爱，你

似度很大；还有就是聚类结果中聚为一类的用户会有各自不同的关注面，因此被选用户类中只有部分用户与主题相关较大。如希望得到更好的效果，可以参考文献[3]的方法。

参考文献 (References)

[1] 董振东, 董强. 知网[URL], 2003. http://www.keenage.com/zhiwang/c_zhiwang_r.html

[2] 刘群, 李素建. 基于《知网》的词汇语义相似度计算[D]. 北京: 中国科学院计算技术研究所, 2002.

[3] 江敏, 肖诗斌, 王弘蔚, 施水才. 一种改进的基于《知网》的词语语义相似度计算[J]. 中文信息学报, 2008, 22(5): 84-89.

[4] 王小林, 王义. 改进的基于知网的词语相似度算法[J]. 计算机应用, 2011, 31(11): 3075-3090.

[5] 杨金柱, 刘金岭. 基于词语上下文的文本分类研究[J]. 计算机技术与发展, 2011, 21(8): 145-149.

[6] 张涛, 杨尔弘. 基于上下文词语同现向量的词语相似度计算[J]. 电脑开发与应用, 2005, 18(3): 41-43.

[7] Y. Yang, J. O. Pedersen. A comparative study on feature selection in text categorization. Proceedings of the 14th International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 1997: 412-442.

[8] 金希茜. 基于语义相似度的中文文本相似度算法研究[D]. 浙江工业大学, 2009.

[9] 魏建香, 苏新宁. 基于关键词和摘要相关度的文献聚类研究[D]. 南京大学, 2008.

[10] 张文东, 易轶虎. 基于兴趣相似性的 Web 用户聚类[J]. 山东大学学报, 2006, 41(3): 45-48.