一种改进的基于后缀树模型搜索结果聚类算法

刘德山
(辽宁师范大学计算机与信息技术学院 大连 116081)

摘 要 针对现有搜索结果分类算法在聚类标签筛选、聚类质量评估及控制重叠聚类方面的缺陷,提出了一种改进的基于向量空间模型与后缀树模型的检索结果聚类算法,从而完善了LINGO算法的聚类及聚类标签打分函数,增加了基本类合并过程,改善了对中文的处理效果。最后对算法的分类效果及产生标签的质量进了实验分析,基于carrot²框架,建立了Web搜索结果聚类推荐平台。验证了CQIG算法分类的准确性和聚类标签的区分性和可读性。

关键词 搜索结果聚类,后缀树模型,向量空间模型,奇异值分解
中图法分类号 TP391.1 **文献标识码** A

Improved Search Results Clustering Algorithm Based on Suffix Tree Model

LIU De-shan
(School of Computer and Information Technology, Liaoning Normal University, Dalian 116081, China)

Abstract To make up for the deficiencies in clustering label selection, clustering quality evaluating and the control of overlapping clustering in the existing search results classification algorithm, this paper proposed an improved search results clustering algorithm based on vector space model and suffix tree model. We modified LINGO algorithm's clustering function and clustering label scoring function, basic clustering merging process was added and the treatment effect of Chinese was improved. Finally, we analyzed the algorithm's classification results and the generated label's quality according to the experiment results. What's more, a platform for recommended Web search results clustering based on carrot² framework was established and CQIG algorithm's classification accuracy and clustering label's discriminative and readability were confirmed on this platform.

Keywords Search results clustering, Suffix tree model, Vector space model, Singular value decomposition

1 引言

互联网的广泛使用给人们提供了共享知识、交换信息、沟通协作的有利平台,如今网络搜索引擎已经成为了人们从网上查询信息的重要工具。然而,网络的开放性在给人们带来便利的同时,也导致信息爆炸式地增长。当前通用的搜索引擎,对于用户提交的查询,通常是按照一个排好序的列表平板式地展现搜索结果。搜索引擎产生的搜索结果可能是几千甚至几万条,如果用户所需要的信息不能展示在结果序列的前端,用户将很难挑选出自己需要的信息。

针对以上问题,利用聚类技术将检索结果进行聚类,然后为每一个聚类产生一个具有代表性的由词或短语组成的标签,再将聚类结果展示给用户成为一种比较有效的解决办法。一些对于检索结果进行聚类的系统也随即产生,如Visimo, Carrot² 都是比较成熟的系统,它们总体上都有着不错的性能,但在聚类标签的可读性、区分性及用户兴趣指导性上还存在不足,用户依然很难定位自己需要的信息。另外聚类的质量也需要进一步提高。

在聚类算法上,STC方法是一种公认较好的用于Web搜索结果聚类的算法,它包括3个主要步骤[1]:1)文档的准备;2)基本类的发现;3)基本类的合并。但STC算法缺少一个有效的相似度度量去评估词组在全局的角度对文档的重要性,缺少有效的方法来衡量STC中类别的质量。同时STC的聚类筛选方法还存在不能很好控制重叠聚类的缺陷。

SHOC、Lingo算法将向量空间模型与后缀树文档表示模型结合起来[2],既考虑了词的位置信息,又考虑了词的统计特性,在STC的基础上有了较好的发展。然而,现有的聚类算法普遍存在聚类标签可读性不强、信息量不足、区分性较差等问题,且聚类结果不能充分反映用户兴趣[3]。

本文提出了一种改进的Web检索结果聚类推荐算法,其构建后缀数组找到完整短语,结合矩阵奇异值分解产生候选聚类标签,选取更为有效的特征改进标签评分公式和聚类得分公式。同时采用了基本类合并技术,产生了更具表述性、区分性和可读性的聚类结果并有效控制了重叠聚类。本算法还引入Lucence中的中文处理方法,对于中文检索的处理也能达到满意的效果。

2 CQIG(Cluster Quality Improved Grouping algorithm)算法

2.1 算法涉及的相关概念

(1)后缀数组及完整短语发现

对于每一个文档片段 T,S 表示 T 的按字母表顺序排列

到稿日期:2010-12-29 返修日期:2011-04-07 本文受辽宁省教育厅科研基金(2008366)资助。
刘德山(1970-),男,硕士,副教授,主要研究方向为网络工程等,E-mail:liudeshan@dl.cn。

的所有后缀组成的数组。从 T 的第 i 个位置开始到文本末尾结束的字串用 $s[i]$ 表示(也称作半无穷串)。引入 LCP 数组可以加快字符串的查找操作。一个 LCP(longest common prefix)数组存储了 $N+1$ 个 lcp 元素(N 为文本字符串的字符总数),$lcp[i]$的值为 $s[i-1]$与 $s[i]$的最长公共前缀,$lcp[0]=lcp[N]=0$。利用后缀树组 S 及 LCP 数组 lcp 可以快速地抽取出文档片段的完整短语[13]。假设 T 由元素序列 $(t_1,t_2,t_3,\cdots,t_n)$组成,当 S 出现在 T 的 k 个位置 $p_1,p_2,p_3,\cdots,p_k$ 时,有 $\exists i,j\in 1,\cdots,k: t_{p_i-1}\neq t_{pj-1}$(左完整),同时有 $\exists i,j\in 1,\cdots,k: t_{p_i+|s|}\neq t_{pj+|s|}$(右完整),则 S 是 T 的一个完整子串。完整短语的发现可以分为两个步骤,第一步是发现左完整及右完整短语,第二步将既具有左完整性又具有右完整性的完整短语合并到完整短语集合中。

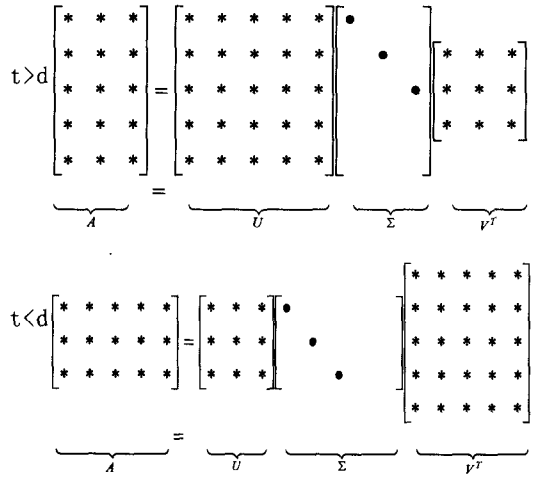
(2)矩阵的奇异值分解

假设 A 是 $t\times d$ 的实值矩阵,它的秩 $\mathrm{rank}(A)=r$。$\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_r$ 是 AA^T 及 A^TA 的 r 个非负特征值,对应的正交特征向量分别是 $x_1,x_2,\cdots,x_t$ 和 $y_1,y_2,\cdots,y_d$。则矩阵的奇异值分解(Singular Value Decomposition)可以定义为[4]:

$$A=U\begin{bmatrix}\Sigma & 0\\ 0 & 0\end{bmatrix}V^T \tag{1}$$

式中,$\Sigma=diag(\sigma_1,\sigma_2,\cdots,\sigma_r)$,而 $\sigma_i=\sqrt{\lambda_i}\ (i=1,2,\cdots,r)$被称为矩阵 A 的奇异值。而 $U=[x_1,x_2,\cdots,x]$,$V=[y_1,y_2,\cdots,y_d]$称为 A 的左、右奇异向量。

如果 A 是由词语-文档构成的关联矩阵,则潜在语义标引模型采用奇异值分解(Singular Value Decomposition)法把矩阵 A 分解成 3 个部分:$A=U\Sigma V^T$。矩阵 U 是由词语间关联矩阵 AA^T 导出的特征向量矩阵;V^T 是由文档间关联矩阵 A^TA 导出的特征向量矩阵;Σ 是 $r\times r$ 阶奇异对角矩阵,r 是矩阵 A 的秩[4]。



选取适当的 k 值,将 Σ 中最大的 k 个奇异值及其相对应的行、列保存,其他的奇异值及其相对应的行、列删除;再取 U、V 最前面的 k 个列向量分别构建 U_k、V_k 的 k 个列向量,由此得到 $A_k=U_k\Sigma_kV_k$,其中 $k(k<r)$是降维后的概念空间的维度。

降维因子 k 选取非常关键,一方面,k 应该足够大,以反映原始数据的信息与结构;另一方面 k 应该足够小,以便于过滤掉所有不相关的冗余信息及噪音。实时处理时,考虑到计算效率,可以按照如下规律选取降维因子 k,令 k 满足:

$$\frac{\sum_{i=1}^{k}\sigma_i}{\sum_{i=1}^{r}\sigma_i}\geqslant\theta$$

(3)抽象概念及短语匹配

由于在特征提取过程中,不论是抽象概念还是词组的发现都表示在同样的空间即原始词语文档矩阵的列向量空间,因此,传统的 Cosine 距离可以用于计算一个词组或者单词是否表示一个抽象概念。

因此,我们定义一个 $t\times(p+t)$的矩阵 P 表示词语文档矩阵中的列向量所表示的单词和词组。P 中将单词和词组看作伪文档,用 $tf\text{-}idf$ 值表示权重[5]。

根据矩阵 P 和奇异值分解所产生的矩阵 U 中的第 i 个列向量 U_i(即第 i 个抽象概念向量),则夹角余弦向量 m_i 能被计算出来:$m_i=U_i^TP$(Lingo 中将此值作为候选标签得分)。m_i 值超过筛选阈值的单词和词组将被挑出用来表示抽象概念。

在短语匹配阶段每一个单独的抽象概念向量将被扩展到整个 U_k 矩阵,因此矩阵 $M=U_k^TP$ 将能计算产生所有抽象概念-词组对的夹角余弦值。根据 M 可挑选出候选聚类标签。

2.2 CQIG 算法的主要步骤

首先采用了后缀树组用以发现频繁词组,并通过矩阵奇异值分解(SVD)提取抽象概念和聚类标签。在聚类内容发现阶段我们沿用了传统的向量空间模型(VSM)[6]。具体步骤如下:

(1)预处理:对输入数据的预处理是信息检索中非常重要的一个步骤,数据的处理效果直接影响到算法分类结果的质量。

预处理阶段包括 4 个步骤:

Step1 文本过滤,去除 HTML 标签、实体和除了短语边界外的非字符特征。

Step2 对文档片段的语言进行识别并选取相应的词干提取和停用词标记方法。Carrot^2 原有版本语言中并没有中文,本文引入 Lucence 中的中文处理方法,加强了对中文的识别。

Step3 词干提取,去除前缀及后缀等形式变化,将一个词的所有形式都转为一个唯一的词干,提高描述能力。但由于用户很难理解光秃的词干,因此词的原始形式也将存储。

Step4 停用词标记。根据语言相应的停用词表进行标记。我们认为停用词可以帮助我们理解语义,消除歧义,因此选择标记停用词而不是去除停用词。

(2)特征选取:主要目的是发现潜在的能够解释 LSI 中提取出的抽象概念含义的单词和词组即候选聚类标签。作为候选标签的词组必须具有以下性质:在输入文档中出现超过一定次数,不穿越短语标记,是完整短语,不以停用词开始和结尾[7]。

特征选取阶段包括 4 个步骤:

Step1 文档表示转换,将文档的表示形式从基于字符的转换为基于单词的。

Step2 文档连接,将所有的输入文档连接起来。

Step3 完整短语发现,用后缀树组发现频繁短语,在频繁短语中提取左完整和右完整短语,按字母表对左完整短语进行排序,将既具有左完整性又具有右完整性的完整短语合

并到完整短语集合。

Step4 候选类标签选取，进一步地处理完整短语，选出超过词频阈值的短语作为候选类标签。

(3)聚类标签归纳：在聚类标签归纳阶段，有意义的类描述是通过对词语文档矩阵进行奇异值分解产生的。包括4个步骤：

Step1 词语-文档矩阵的建立，对输入的文档片段集合建立词语-文档矩阵A，建立时选取超过词频阈值的词，并采用$tf\text{-}idf(p_i,d)$作为矩阵中的权值。

Step2 抽象概念的发现，对词语-文档矩阵A进行奇异值分解，产生相应的U,Σ,V矩阵。根据2.1节中的方法进行抽象概念的发现。

Step3 短语匹配，构建矩阵P并根据2.1节的方法进行短语匹配，选取最能表述抽象概念的短语。

Step4 修剪和评估。计算所有候选标签间的相似度，将超过相似阈值的标签分成组，在每一组中选取一个得分最高的标签作为候选标签。标签得分的计算参见2.3节。

(4)聚类内容发现：在聚类内容发现阶段[8]，定义一个矩阵Q，每一个聚类标签表示为Q中的一个列向量。定义$C=Q^TA$，A是原始的词语文档矩阵。这样矩阵C的元素C_{ij}就表示第j个文档属于第i的聚类的程度。对于每一个候选聚类标签，用VSM模型查询方法将标签矩阵Q当作关键词查询输入文档矩阵A，对C中元素根据聚类分配阈值进行筛选，建立以L为描述标签的聚类C，将返回的片段代表的文档归为一类将查询标签当作类标签，最后，将所有未被分类的文档放入“Other topics”类。

(5)最终聚类构成：包括聚类得分计算和聚类合并，具体过程在下一节中有详细说明。

2.3 CQIG算法的聚类质量评价方法

Lingo算法中聚类得分为ClusterScore=LableScore$*||C||$，$||C||$表示聚类中文包含档片段的个数，聚类标签的得分就是文档矩阵中标签所代表的列向量的值。这种打分方式过于简单，产生的聚类标签在统计特性上考虑不完善，没有很好反映出用户的查询兴趣，聚类标签的语义性、可读性也需要提高。另外对聚类结果的内聚性和聚类间的区分性并没有做考虑。

本文算法改进了标签得分公式和聚类得分公式：

$$\text{ClusterScore}(C_i)=\text{Score}(L(C_i))*\text{Score}(C_i) \tag{2}$$

(1)给定聚类集合$C=\{C_1,C_2,\cdots,C_n\}$，$L(C_i)$表示聚类C_i中$i\in[1,n]$类标签，p_i聚类标签所代表的短语，d为聚类C_i包含的一个文档片段，$|d|$为聚类C_i包含的文档片段数。

$$\text{Score}(L(C_i))=|L(C_i)|*f|L(C_i)|*f(query)*f(tf\text{-}idf(p_i)) \tag{3}$$

式中，$|L(C_i)|$是聚类标签$L(C_i)$出现的文档片段的个数，$f|L(C_i)|$是聚类标签$L(C_i)$中短语长度得分，$f(query)$是聚类标签中查询词得分系数，$f(tf\text{-}idf(p_i))$是标签短语的统计特性得分系数[9]。

$$f(query)=\begin{cases}\beta, & \text{if 查询词出现在短语 } p_i \text{ 中}\\ & (\text{本文中 } \beta \text{ 设置为 } 100)\\ \gamma, & \text{if 查询词没有出现在短语 } p_i \text{ 中}\\ & (\text{本文中 } \gamma \text{ 设置为 } 1)\end{cases} \tag{4}$$

$$f((|L(C_i)|)=\begin{cases}0, & \text{if } |p|=1\\ |p|, & \text{if } 2\leqslant|p|\\ \alpha, & \text{if } |p|>8\end{cases} \tag{5}$$

$$f(tf\text{-}idf(p_i))=\sum f(tf\text{-}idf(p_i,d)) \tag{6}$$

$$tf\text{-}idf(p_i,d)=(1+\log(tf(p_i,d)))*\log(1+N/df(p_i)) \tag{7}$$

式中，$df(p_i)$表示短语p_i出现的文档片段的个数，$tf(p_i,d)$表示短语p_i在整个文档集中出现的频度。

通过以上公式计算出的标签得分可以更好地反映出标签的统计特性，给定查询词中出现的短语较大的权值也能更好地反映出用户的查询兴趣，通过标签长度的得分系数限制也提高了聚类标签的语义性、可读性。

算法1 聚类标签产生算法

输入：文档片段

输出：候选聚类标签集

```
for each d∈D do/*每一个文档片段*/
    A←构建词-文档矩阵;/*选取未被标记成停用词并且词语出现
    频率大于给定阈值的词建立词-文档矩阵*/
end for
Σ,U,V←SVD(A);/*奇异值分解降低矩阵A的维度并产生抽象概
    念*/
k←0;
r←rank(A);
repeat
    k←k+1;
    q←(∑_{i=1}^{k}σ_i)/(∑_{i=1}^{r}σ_i)
until q<候选标签阈值θ;
P←根据 p_f 构建短语矩阵;
for each U_k^T P 的列向量 do
    找到最匹配的频繁词组←m_i;/* m_i 为 U_k^T P 中值最大的列元素
    */
    set of cluster lables←用短语表示抽象概念作为候选类标签;
    Score(lable)←根据本文方法计算标签得分;
end for
    V←候选标签和 Score(lable);/*根据标签及其得分建立候选类
    标签矩阵V*/
    Z←将候选标签视为文档建立词-文档矩阵;
    标签间的相似度←Z Z^T;/*计算候选标签的相似度*/
    根据相似度阈值去除相似标签，选取得分最高的一个作为候选聚
类标签;
```

(2)一个好的结果聚类推荐需要具有一个有较强指导意义的聚类标签和一组有良好的内聚性和区分性的类别[10]。要求在一个聚类内部的文档有很强的关联性，并能通过聚类标签概括聚类中文档的含义，保证不同聚类间有较强的区分性。本文采用聚类内相似度$\text{sim}_{\text{in}}(C_i)$和聚类间相似度$\text{sim}_{\text{out}}(C_i)$两方面来衡量候选聚类的聚合性。$Cohesion(C_i)$表示候选聚类的聚合性，$Cohesion(C_i)$值越大聚类质量越高。因此有：

$$Score(C_i)=Coherency(C_i)*|d| \tag{8}$$

$$Coherency(C_i)=\frac{\text{sim}_{\text{in}}(C_i)}{\text{sim}_{\text{out}}(C_i)} \tag{9}$$

候选聚类$o(c)$的聚类中心定义如下式：

$$o(c)=\frac{1}{|d|}*\sum_{d\in D(c_i)}v(d) \tag{10}$$

式中，$D(C_i)$表示候选聚类C_i的文档集合，$|d|$表示C_i的文档数量，$v(d)$是以向量表示的文档。本文以聚类中文档与聚

类中心[11]的夹角余弦值来评价其聚类内相似度 $sim_{in}(C_i)$，以聚类 C_j 中文档与 C_i 的聚类中心的夹角余弦值来表示聚类 C_j 与聚类 C_i 的聚类间相似度 $sim_{out}(C_i)$，则：

$$sim_{in}(C_i)=\frac{1}{|d|}*\sum_{v(d)\in|D(c_j)|}\cos(v(d),o(c)) \quad (11)$$

$$sim_{out}(C_i)=\frac{1}{n}*\sum_{i,j=1}^{i\neq n}\frac{1}{|d(c_j)|}*\sum_{d\in D(c_j)}\cos(v(d),o(c_i)) \quad (12)$$

式中，$|d(C_j)|$ 表示候选聚类 C_j 的文档数量，n 表示全部聚类集合 C 中聚类的个数。

(3)本文采用重叠聚类合并过程，但并没有采用层次聚类，因为考虑在聚类的数量、区分性、可读性适当的情况下层次聚类会使用户查找过程更繁琐，也会增加聚类时间。这部分采用如下方法：

限定一个重复合并阈值 MT，当两个待合并，给定两个聚类 A,B，判断其是否需要合并，若需要合并则比较两个聚类的大小，将规模小的聚类合并到规模大的聚类中，并去除重复文档。对每一个合并组选取得分最高的聚类并将其余的低分聚类与之合并。

算法 2　最终聚类形成算法

```
输入:文档片段
输出:最终聚类推荐结果
for each snippets
    ClusterScore(Ci)←Score(L(Ci)) * Score(Ci);/ * 根据本文方法计算聚类得分 * /
end for
for each 待合并聚类 setA,setB / * 根据本文方法进行重复聚类合并,setA,setB 为一对待合并聚类 * /
    boolean 判断两聚类是否需要合并;
    if(需要合并 and size(A)< size(B)) / * size(A),size(B)分别表示聚类 A,B 中的文档数 * /
        保留 B 中文档,将 A 中文档加入,并去除重复;
    else
        保留 A 中文档,将 B 中文档加入;
    end if
end for
for each merge group
    ClusterScore(i)←根据本文方法计算待合并组中每个聚类得分;
    Sort(ClusterScore);
    选取得分最高的聚类并将其余的合并到这个类中;
    Final Cluster←产生最终聚类推荐结果;
end for
```

另外，carrot² 原有版本对中文效果不佳，本文算法加入了中文词法分析及中文停用词表，加强了对中文的处理效果。

3　CQIG 算法实现及实验结果分析

3.1　基于 Ambient 数据集的聚类质量评测

Ambient 是一个数据集，用以评估信息检索分类质量。它由 44 个类组成，每个类都包含一个主题设置和 100 个网页文件列表(收集自网络搜索引擎)，测试数据集将它们标注为相应的类别。由于测试数据集在聚类结果中又被重新划分为多个类别，因此根据 Ambient 数据集可以计算本文算法与 Lingo 及 STC 算法的 Recall、Precision、F-Score[12] 及 NMI 的值对比。从 44 个结果中随即选取了 10 个，如图 1—图 4 所示。

Recall(召回率)的值能直接反映出聚类算法发现聚类的能力。从图 1 这组结果对比可以看出，在召回率方面 CQIG 的总体效果要好于 Lingo 和 STC，并且在每个主题上都保持着比较高的召回率，而 STC 浮动较大相对不稳定。由于结合了后缀树模型和向量空间模型的优势，因此 CQIG 算法具有相对高而稳定的聚类能力。

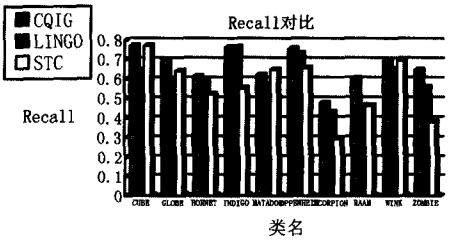


图 1　召回率对比

Precision(准确率)的值能反映出聚类算法发现聚类的准确程度。图 2 结果显示在 Ambient 数据集中 CQIG 算法准确率明显高于其他两种算法，平均值达到了 0.9。可见采用了本文的聚类质量评分公式后算法的准确性非常高。

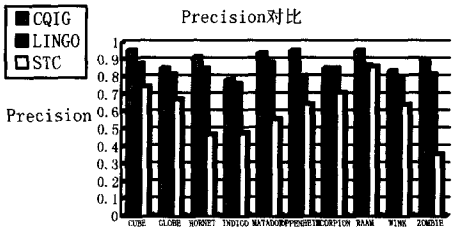


图 2　准确率对比

F-Score 是一种衡量特征集在类别之间的辨别能力的方法，图 3 中结果对比表明本文的 CQIG 算法对聚类的判别能力是最强的，其次是 Lingo 算法，STC 算法最弱。由于本文 CQIG 算法采用了聚类合并，因此算法能产生更具辨别性、区分性的聚类。

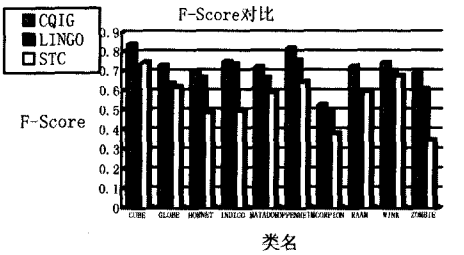


图 3　F-Score 对比

NMI 为正规化互信息，是对查准率和召回度统计信息的一种均衡性度量，用来避免真实聚类和算法结果因共享信息所导致的噪音[12]。当 NMI 的值接近 1 时表示聚类标记接近原有的类别标记，当 NMI 的值接近 0 时说明聚类过程只是进行了一个随机的划分。图 4 实验结果表明本文算法聚类的均衡性最好。

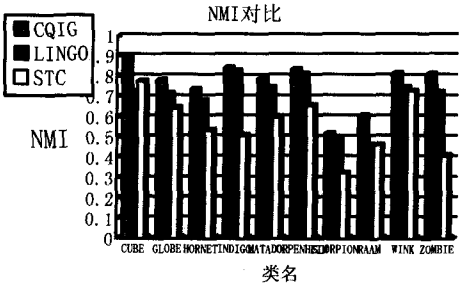


图 4　NMI 对比

3.2 基于 Web 搜索结果聚类推荐平台的实验结果对比分析

本文采用了 carrot² 平台作为基础框架，建立了 Web 搜索结果聚类推荐平台。通过各种大型搜索引擎 API 获得源数据，通过网页清洗，分词，提取特征项，建立 VSM，对 CQIG、STC 及 Lingo 进行聚类，聚类后把聚类结果展现给用户。

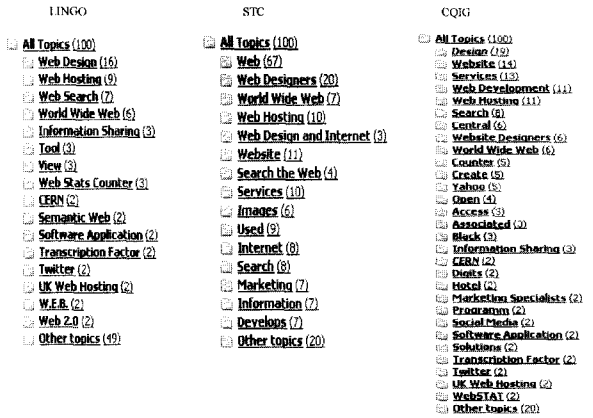


图 5 关键词：Web

从图 5 这个关键词的检索结果对比可以看出 Lingo 算法虽然标签可读性较好，但是在 100 个结果中有 49 个未找到合适的分类，显示出的类别的内聚性也不好。在 STC 算法中查询词是 Web，而推荐聚类中 Web 标签下的就有 67 个，这个标签处于第一位但对用户的指导意义并不大。因此从标签的可读性和指导性上看我们的 CQIG 算法都是较好的。另外本文算法采用了聚类处理过程，因此在聚类的区分性及控制重叠聚类的效果上 CQIG 也是最好的。

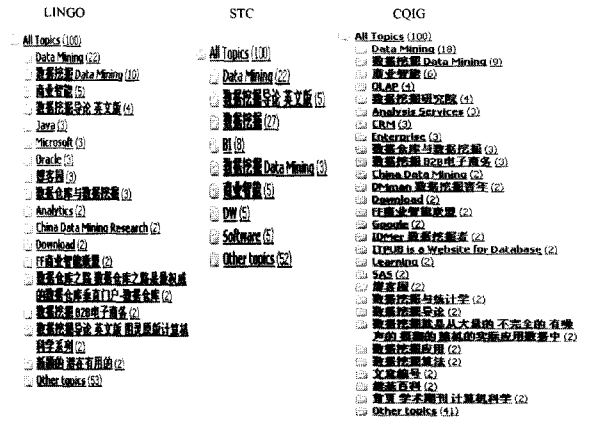


图 6 关键词：数据挖掘

从图 6 的关键词检索结果对比可以看出 Lingo 算法对于中文的处理能力虽然好于 STC，但未分类的文档却是最多的，在推荐的类别中中文标签的语义性、可读性较差。我们的 CQIG 算法产生的标签信息量较充足，中文标签的可读性和区分性效果最好。

结束语 本文实现了一种改进的 Web 检索结果聚类算法 CQIG，其将向量空间模型与后缀树模型的优点结合起来。在 Lingo 算法的基础上改进了聚类标签及聚类的得分计算方法，产生了更具可读性、更易理解的标签，加入了重叠聚类合并过程，既保留了重叠聚类的优势，又有效控制了重叠聚类的数量，最终的聚类结果对用户的选择更具指导性，同时我们加强了对中文的处理效果。下一步我们会在时效性方面进行加强，并进一步完善中文处理效果。

参 考 文 献

[1] Wu Jiang-ning, Wang Zhi-jiang. Search Results Clustering in Chinese Context Based on a New Suffix Tree[C]//Proceedings of the IEEE 8th International Conference on Computer and Information Technology(CIT2008). Sydney, Australia, July 2008

[2] Janruang J, Kreesuradej W. A New Web Search Results Clustering based on True Common Phrase Label Discovery[C]//CIMCA '06. Procceedings of the International Conference on Computational Inteligence for Modeling Control and Agents Web Technologies and International Commerce. Washington, DC, USA, IEEE Computer Society, 2006

[3] Carmel D, Roitman H. Enhancing cluster labeling using wikipedia[C]// Proceedings of SIGIR '09. Boston, Massachusetts, USA, 2009

[4] Osiński S. Reduction D. Techniques for Search Results Clustering[D]. Department of Computer Science, The University of Sheffield, UK, 2004

[5] Zhang Wei, Xu Bao-wen. ISTC: A new method for clustering search results [J]. Wuhan University Journal of Natural Sciences, 2008, 13(4): 501-504

[6] Osiński S. An Algorithm for Clustering of Web Search Results [D]. Master thesis, Department of Computing Science, Poznań University of Technology, 2003

[7] 骆雄武，万小军，杨建武，等. 基于后缀树的 Web 检索结果聚类标签生成方法[J]. 中文信息学报，2009，23(02)：83-88

[8] Osiński S, Stefanowski J, Weiss D. Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition[C]// Soft Computing, Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'04 Conference. Zakopane, Poland, 2004: 359-368

[9] Chim H. A new suffix tree similarity measure for document clustering[C]//Proceedings of the 16th international conference on World Wide Web. 2007: 121-130

[10] Wang Xuan-hui, Zhai Cheng-xiang. Learn from Web Search Logs Organize Search Results[C]// Proceedings of SIGIR '07 the 30th annual international ACM SIGIR conference on Research and development in information retrieval. New York. NY, USA: ACM Press, 2007: 87-94

[11] Zeng Hua-jun, et al. Learning to Cluster Web Search Results[C]// Proceedings of SIGIR'04. Sheffield, South Yrkshire, UK, 2004

[12] 朱君，曲超，汤庸. 利用单词超团的二分图文本聚类算法[J]. 电子科技大学学报，2008(03)

[13] Zhang Dong. Towards Web Information Clustering[D]. Nanjing: Southeast University, 2002