基于语义关联树的分类查询扩展算法

任永功 范 丹 武佳林
(辽宁师范大学计算机与信息技术学院 大连 116029)

摘 要 查询扩展技术中引入语义计算是一个重要的研究方向。针对现有解决方法普遍存在缺少主题知识、引入无关词以及筛选函数不恰当的问题,提出了一种结合主题选取与局部反馈方法的语义关联树模型,从语义的角度进行分类查询扩展。在传统方法基础上结合Web文本分类语料库进行了有主题的分类扩展,并改进了扩展词筛选函数,增加了阈值限定,有效控制了噪音。结合用户交互与局部反馈的方法不但减少了传统相关反馈中用户的工作量而且弥补了单纯局部反馈高度依赖于初次检索结果的缺陷。在SMART平台的实验结果表明,该方法相比一般的查询扩展算法查全率及查准率均有所提高。

关键词 语义关联树,主题选取,查询扩展,Web文本分类
中图法分类号 TP391.1 **文献标识码** A

Classified Query Expansion Algorithm Based on Semantic Relation Tree

REN Yong-gong FAN Dan WU Jia-lin
(School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China)

Abstract Introducing semantic computing technology into the query expansion is an important research direction. In this paper we presented a semantic relation tree model which combines with topic selection and local feedback method, classified expand query from the perspective of semantic. Traditional methods exist for the problems, such as the lack of knowledge in the topic, the introduction of irrelevant words and the filter functions are not proper. We introduced Web text classification into the semantic relation tree model to make subject expansion with improving the word filter function and increasing the threshold limit to control noise. The combination of user interaction with the local feedback method not only reduces the user's work in traditional relevance feedback method but also solves the problem of highly dependent primary retrieval result in local feedback. The experimental results on the SMART platform show that this method can increase the rate of recall and precision.

Keywords Semantic relation tree, Topic selection, Query expansion, Web page classification

1 引言

由于Internet网络的开放性和信息发布的容易性,Web急剧膨胀,其资源以指数速度增长,导致人们查询信息时出现关键词不匹配和信息过载等难以克服的问题[1]。如何解决这些问题,从而达到高效、准确地从海量信息中寻找到更多所需信息的目的,一直是信息检索领域中一个十分重要而具有挑战性的研究课题。

查询扩展指的是利用计算机语言学、信息学等多种技术,把与原查询相关的词或者词组添加到原查询,得到比原查询更长的新查询,然后检索文档。查询扩展是解决关键词不匹配和信息过载问题的关键技术之一,它能够弥补用户查询信息的不足,改善和提高信息检索系统的查全率和查准率。

传统的查询扩展方法主要可分为全局分析和局部分析两大类。全局分析基本思想是对整个文献集的语词进行相关分析,得到每对语词的关联程度,构造叙词表,再从叙词表中选取与原查询关联程度较高的词作为扩展词进行查询扩展[2]。全局分析可以最大限度地探求词间关系,并在词典建立之后以较高的效率进行查询扩展。但是,当文档集合非常大时,建立全局词关系词典的时空复杂度是无法忍受的,并且在文档集合改变后的更新代价巨大[3]。局部分析利用初次检索得到的与原查询相关的N篇文章作为扩展用词的来源,而并非用先前计算得到的全局词关系词典[4]。局部分析扩展虽然弥补了全局分析中的不足,但却无法弥补高度依赖于初次检索结果的缺陷。局部上下文分析方法LCA(local context analysis)利用了全局分析中词共现频率的思想,避免了单纯的局部分析方法易向原查询加入不相关的词的缺点。该方法的检索实验结果明显优于传统的全局分析和局部分析方法[5]。

传统的查询扩展虽然在技术上有了很大改进,但是还不能够从根本上完全改善信息检索性能。普遍存在引入无关

到稿日期:2009-01-21 返修日期:2009-03-20 本文受国家自然科学基金项目(60603047),辽宁省科技计划项目(2008216014),辽宁省教育厅高等学校科研基金(2008341),大连市优秀青年科技人才基金(2008J23JH026)资助。
任永功(1972-),男,博士,教授,主要研究方向为数据挖掘技术等,E-mail:renyg@dl.cn;范 丹(1985-),女,硕士研究生,主要研究方向为Web挖掘、信息检索、XML数据库;武佳林(1985-),男,硕士研究生,主要研究方向为Web挖掘、XML数据库。

词、缺少主题知识以及筛选函数不恰当等问题。主要原因是传统的查询扩展主要是以查询词为中心，采用机械式的符号扩展，亦即在符号层次上进行的查询扩展，忽略了查询语义及查询概念语义间的关联的扩展，并通常缺乏领域知识。因而没有充分表达和扩展用户查询意图[6]。针对传统查询扩展的缺陷，Qianli Jin 等提出了语义关联树模型[7]。

本文在总结查询扩展技术的基础上，提出了一种基于语义关联树的分类查询扩展方法。将查询扩展技术结合到主题检索中，采用 Web 文本分类技术构建分类语料库，将分类语料库结合到语义关联树的构建算法中，并结合用户交互进行扩展词的选择。这样，用户可以用来限制搜索范围、明确搜索目的，使查询扩展更符合用户需求，更有针对性，更精确。此外，本文还对查询扩展算法的筛选函数做了进一步改进，并增加了阈值选择，进一步优化了语义空间，使扩展的准确度更高。

2 语义关联树概念的提出及模型的构造

2.1 概念的提出及元素的提取

查询扩展技术中引入语义计算是一个重要的研究方向。目前大多数的计算语义模型的目标是构造“词-词”相似度矩阵[8]。“词-词”相似度矩阵，反映在语义空间中就是任何两个节点之间都有连线。这种语义空间是一个完全图，规模庞大。但现实中考虑词与词之间关系的时候，往往仅关心那些相似度比较大的词，如果两个词的含义几乎无关，在统计意义上也很少共现，那么给出它们之间的相似度值没有实质意义。既然这么大规模的完全图并不是全有用，就应该考虑给它剪枝，把那些相似度值小的路径删去，仅留下相似度值大的那些路径。更理想的情况是，在训练过程中，就能够自动保留那些有意义的路径。在数据结构方面，比“图”更简单更灵活的模型就是“树”，因此采用树状的模型来构造语义空间，使得它能克服图状语义空间的不足。我们把这个模型称为“语义关联树”模型[9]。

设 $\mathrm{Sim}(W_k,q)$ 表示两个词 W_k 和 q 之间的先验相似度值。对任意一个给定的词 W_k，采用树状的模型来表达词 q 与所有其它词之间的关系，如图 1 左边部分所示。

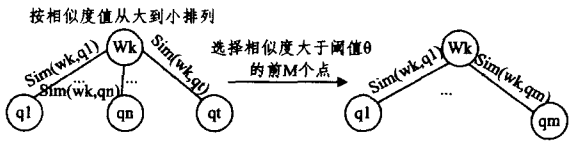

图 1 提取语义关联树元素

图 1 中，原词 W_k 是树根，而其余的词是树叶，两者之间树枝（也就是路径）的权值是两个词之间的相似度值。这里 W_K 到 q 的所有词，是按照它们和原词 W_K 之间的相似度来排序的：

$$\mathrm{Sim}(W_K,q_1)\geqslant\cdots\geqslant\mathrm{Sim}(W_K,q_n)\geqslant\cdots$$

这里相似度 $\mathrm{Sim}(w_i,q_i)$ 的计算有很多种方法，如：

1）训练语料中的共现频率；

2）在词典如 WordNet，HowNet 中的距离（本文用 HowNet 距离）；

3）共现的互信息等等。

传统方法提取出最左边的 M 个树叶（对应最相似度最大的 M 个词），但是实际问题中可能会出现 M 个词中有某些词的相似度值会很小，这样就会导致噪声的引入，因此本文提出了根据相似度阈值 θ 和 M 共同控制树叶的个数的方法，θ 保证了加入语义关联树的词汇的相似度，M 则保证了叶子节点的数量可控。

2.2 模型的构造

设 $W(W_1,W_2,\cdots,W_i)$ 表示初始词向量，其中总共包含 i 个词，W_i 代表第 i 个词。从 W 这个初始词向量出发，构造语义关联树[9]。

图 1 表示的是一个语义关联树元素。事实上，整个语义关联树是由很多个语义关联树元素在不同层次上搭建而成的，如图 2 所示。

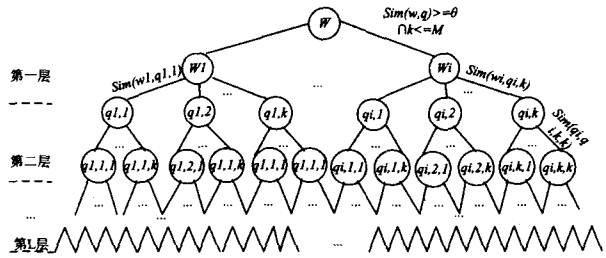

图 2 语义关联树模型（SRTM）

建立语义关联树之后，可以容易地获得一个根节点词与一个叶节点词之间的相似度值。由于对层数 L 和叶节点数 M 的控制，使得语义关联树模型可控性更强，能更好地去除噪声，因此比 LSI 系列模型更具有优势。语义关联树模型不需要预先构造庞大的语义空间结构，因此更容易被大规模应用。

3 基于语义关联树的分类查询扩展

基于语义关联树的分类查询扩展流程如下：

1）首先利用 Web 文本分类技术将 Web 文档按主题分类，并建立主题分类语料库；

2）利用用户给定的初始查询进行第一次检索；

3）使用检索系统初始检索得到的前 N 篇文章作为与原查询最相关的文章，并建立返回文档集；

4）依次在主题分类语料库和返回文档集上选词建语义关联树；

5）遍历建好的语义关联树，通过筛选函数选出与原查询最相关的词作为扩展用词。

3.1 分类语料库的建立

3.1.1 文本分类概念

文本分类是指按照预先定义的主体类别，为每个文档确定一个类别。文本分类是一种典型的有教师的机器学习，一般分为训练和分类两个阶段，具体过程如下：

1）定义阶段

定义类别集合 $C=\{c_1,\cdots,c_2,\cdots,c_m\}$，这些类别可以是层次式的，也可以是平行式的。

给出训练文档集合 $S=\{s_1,\cdots,s_2,\cdots,s_n\}$，每个训练文档 S_i 被标上所属的类别属性 C_i。

统计 S 中所有文件的特征矢量 $V(s_i)$，确定代表 C 中每个类别的特征矢量 $V(c_i)$，该步是训练阶段的一个关键。

2）分类阶段

对于测试文件集合 $I=\{d_1,\cdots,d_2,\cdots,d_t\}$ 中的每个尚待分类的文件 d_k，计算其特征矢量 $V(d_k)$ 与每个 $V(c_j)(C_j\in C)$

之间的相似度 $\mathrm{Sim}(d_k,c_j)$。

选取相似度最大的一个类别 $C_i=\max \mathrm{sim}(d_k,c_j)(c_k,c_j\in C)$作为 d_k 的类别。

3.1.2 Web 文本分类及常用方法

Web 文本分类要考虑 Web 的特征因素。首先，Web 上的网页数量巨大。第二，网页的格式非常灵活。第三，网页内的内容也不单一，通常包含了许多与主题无关的内容，这些对网页分类来说都构成了噪声。因此对 Web 文本进行分类之前要进行 Web 文本预处理去除噪声[10]。

Web 文本分类中的经典算法包括类中心法、K-最近邻(K-Nearest Neighbor，简称 KNN)算法、支持向量机(Support Vector Machine，简称 SVM)法，朴素贝叶斯(Naive-Bayes，简称 NB)法等。

本文中 Web 文本分类系统的实现基于 Svmcls 文本分类系统 2.0 版。Svmcls 文本分类使用了 KNN 和 SVM 两种分类算法，支持中英文和多种特征选择算法，提供了多种工具，可以生成中间结果和兼类分类，支持多种格式，提供了一个 3000 多篇文档的分类语料。本文在 Svmcls 系统中添加了基于朴素贝叶斯的分类方法。

朴素贝叶斯算法主要包括以下两个计算步骤：

Step 1　计算特征词属于每个类别的几率向量。

Step 2　在新 Web 网页到达时，根据特征词分词，然后按公式计算该文本 d_i 属于类 c_j 的几率。

具体就是利用下列公式通过类别的先验概率和词的分布来计算未知文本属于某一类别的概率：

$$P(C_j\mid X)=\frac{P(C_j)P(X\mid C_j)}{P(X)} \tag{1}$$

其中，$P(C_j\mid X)$为样本 X 属于类 C_j 的概率，$P(X\mid C_j)$为类 C_j 中含有样本 X 的概率。在所有 $P(C_j\mid X)(j=1,2,\cdots,m)$中，若 $P(C_K\mid X)$值最大，则文本 X 归为 C_K 类。假设文本中词(属性)的分布是条件独立的，则 $P(C_j\mid X)=P(C_j)P(X\mid C_j)$。

其中，

$$P(C_j)=\frac{C_j\text{ 中文本个数}}{\text{总文本个数}} \tag{2}$$

$$P(d_i\mid C_j)\frac{d_i\text{ 在类 }C_j\text{ 中出现的次数}}{C_j\text{ 中所有词的个数}} \tag{3}$$

根据朴素贝叶斯进行 Web 分本分类：

1)训练过程为：

Step 1　扫描训练文本。

Step 2　对 Web 文本进行处理，为特征选择做准备。

Step 3　进行特征选择，得到最优的特征子集。

Step 4　优化特征子集。

Step 5　代入概率公式计算结果，并将结果存储入文件。

2)分类过程为：

Step 1　扫描训练文本。

Step 2　找到每个文本具有的属性(词)X。

Step 3　按照训练结果的文件提供的数据，找到相应的概率。

Step 4　比较得到最大的概率所属的类别，得出结论。

根据以上方法，对待处理的 Web 文本进行分类，将分类结果存储在分类语料库中。

3.2　语义关联树构造算法及查询扩展算法的实现

3.2.1　分类构造算法

给定初始词向量 $W(W_1,W_2,\cdots,W_i)$，i 表示词向量 W 的长度。以 W 作为根节点构建语义关联树 $SRTM(W,\theta,L,R,M)$，其中 θ 代表纳入语义关联树的相似度显著性阈值，L 为语义关联树的层数，R 为依据主题分类语料库扩展的层数($R\leqslant L$)，M 表示每层最多的节点的个数。与传统固定的扩展算法相比，本算法引入了 θ,L,R,M 4 个可动态设定的参数，θ 保证了纳入扩展的词汇与原始关键字的相似性，R 决定了何时依据主题分类语料扩展、何时依据局部反馈结果扩展，语义关联树规模和复杂度可以通过 L 和 M 的不同设置得到有效控制。通过限定语义关联树每层节点个数以及树的层数，即可动态灵活地控制查询扩展词的规模及相似程度。

$\varphi[\mathrm{Sim}(W_i,q_i),\theta]$为相似度优化函数，有：

$$\varphi[\mathrm{Sim}(p,q),\theta]=\begin{cases}\mathrm{Sim}(p,q)\mid \mathrm{Sim}(p,q)\in \mathrm{Sim}(W) & \mathrm{Sim}(p,q)\geqslant\theta\\ 0\mid \mathrm{Sim}(p,q)\in \mathrm{Sim}(W) & \mathrm{Sim}(p,q)<\theta\end{cases}$$

其中，$\mathrm{Sim}(p,q)$表示属于词集合 W 的词 p,q 之间的相似度，θ 为相似度显著性阈值，用户可根据需求设定 θ 值，当需要较高的查全率时，θ 的设置应较小，以保证更多的相关词汇加入到扩展；当需要较高的查准率时，应适当增大 θ 以免加入过多相关度不大的词汇而引入噪声。

则建树算法如下：

Step 1　输入初始查询词向量 $W(W_1,W_2,\cdots,W_i)$；

返回 I 值；/* I 值为用户输入关键词长度 */

for (int $i=0,i<I,i++$)；

Step 2　读入 W 中关键词 W_i 根节点，构建子树 Sub_Tree(W_i)。

① Set Sub_Tree(W_i)=ϕ；

② Get $\mathrm{Sim}(W_i,q_j)$；/* 计算任一词 q_j 与 W_i 的相似度。本文用 HowNet 中的距离度量相似度 */

③ If $\mathrm{Sim}(W_i,q_j)>\theta$ Then $q_j\in$ vector S；/* 取所有 $\mathrm{Sim}(W_i,q_j)>\theta$ 的节点存入节点容器 S 中 */

End if；

④ Rank(S)；/* 对 S 中的节点按相似度从大到小排序后存入节点容器 F 中 */

⑤ If $F.size()<M$ Then $F\in$ Sub_Tree(W_i)；/* 当满足相似度大于 θ 的节点个数小于每层最多节点数 M 时，取 F 中的所有节点作为 W_i 的子节点。 */

End if；

⑥ Else Get $F[1]\sim F[M]\in$ Sub_Tree(W_i)；/* 否则取 F 中的前 M 个节点作为 W_i 第一层节点 $q_1\sim q_m$ */

Step 3　依次读取 $q_j(l\in[1,M])$ Repeat Step 1，构建子树 Sub_Tree(q_j)；

Step 4　判断语义关联树层数 L_j。

当 $L_j\leqslant R$ 时从主题分类语料库相应的分类中选取扩展词($R\leqslant L$)。

当 $R<L_j<L$ 时从根据局部分析法返回的前 N 篇文档库中选择扩展词。

当 $L_j=L$ 时 Sub_Tree(W_i)子树构建树停止。

参数 R 决定了建树过程中的扩展词来源，过程如图 3 所示，实际应用中当认为分类语料对扩展的影响较大时，可适当增大 R 设置，反之可适当减小 R，甚至将 R 设置为 0。

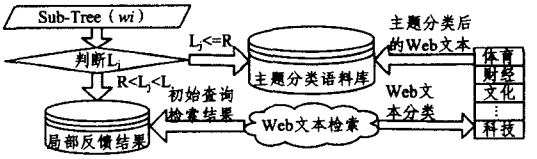

图3 扩展词选取过程

Step 5 遍历 W 各子树中所有节点，Repeat Step 2－Step 4，最终生成的 SRTM(W,θ,L,R,M)由 I 棵子树构成。

End for；

3.2.2 查询扩展算法

首先引入语义关联树所要用到的几个函数：

1)函数 $\mathrm{Sim}_v(w_i,q_{l_{i,j}})$，值为两节点间的最短路径权值。

$$\mathrm{Sim}_v(w_i,q_{l_{i,j}})=\mathrm{Max}[path(w_i,q_{l_{i,j}})]$$

$$path(w_i,q_{l_{i,j}})=\mathrm{Sim}(w_i,q_{l_{i,1}})\times\cdots\times\mathrm{Sim}(q_{l_{i,j-1}},q_{l_{i,j}})$$

$q_{l_{i,j}}$ 表示与根节点 w_i 连通的处于语义关联树第 L 层的非根节点，根节点和非根节点之间的相似度均可如此计算，计算复杂度仅与语义关联树的层数相关。

2)函数 $\mathrm{Appear}(q_{l_j},W)$，计算语义关联树第 j 层词与整个词向量 W 的相关权重[11]：

$$\mathrm{Appear}(q_{l_j},W)=\mathrm{Overlay}(\mathrm{SRTM}(W,\theta,L,R,M),q_{l_j})/i$$

若原主题词向量 W 总共有 i 个词，在语义空间中就有 i 个子树，则 $\mathrm{Overlay}(\mathrm{SRTM}(W,\theta,L,R,M),q_{l_j})$ 表示在这些子树中有多少个包涵盖了词 q_{l_j}。

3)函数 $\mathrm{Sim}(W,q_{l_j})$ 计算语义关联树第 j 层词 q_{l_j} 与整个词向量 W 的相似度：

$$\mathrm{Sim}(W,q_{l_j})=\sum_{1}^{i}\mathrm{Sim}_v(w_i,q_{l_j})\times\mathrm{Appear}(q_{l_j},W)$$

4)函数 $\mathrm{EvlSimTree}(W)$，计算初始查询 W 所有子树的平均相似度：

$$\mathrm{EvlSimTree}(W)=\sum_{i=1}^{i}\mathrm{SimTree}(W_i)/i$$

其中，$\mathrm{SimTree}(W_i)$ 表示 W_i 子树的所有节点的相似度总和。由于实际应用中用户在输入原始查询的时候并不能保证每个词都是对扩展有益的，每一个噪声节点的引入都会导致语义关联树中不相关的扩展词成几何级数增长。因此本文提出了筛选函数 $\mathrm{EvlSimTree}(W)$，当某个子树的相似度总和小于平均相似度过多时(此时用一有效性阈值 ξ 表示)说明这些词与扩展词集整体的相关性较低，加入这些词可能会降低扩展的精度。因此这个子树所代表的词将不加入扩展集合，保证了树中节点更符合用户需要，同时缩减了语义空间。

5)扩展词集合 $\mathrm{Sub}(q_{l_j},\phi,\xi)$：

$$\mathrm{Sub}(q_{l_j},\phi,\xi)=\{p\mid p\in q_{l_j},\mathrm{Sim}_v(W,p)\geqslant\phi\cap\mathrm{SimTree}(W_i)-\mathrm{EvlSimTree}\in[-\xi,0]\cup(0,+\infty)\}$$

则查询扩展算法如下：

Step 1 遍历语义关联树所有节点，设当前遍历为第 j 层词 q_{l_j}，计算出权重 $\mathrm{Appear}(q_{l_j},W)$。

For k=1 To i；

计算 $\mathrm{Appear}(q_{l_j},W)$；

Next；

Step 2 调用函数 $\mathrm{Sim}_v(W_i,q_{l_{i,j}})$ 及 $\mathrm{Appear}(q_{l_j},W)$，得 q_{l_j} 与 W 相似度。

Step 3 调用函数 $\mathrm{SimTree}(W_k)$，求得所有子树的平均相似度 $\mathrm{EvlSimTree}(W)$。

For k=1 To i；

$\mathrm{SumSimTree}(W)=\mathrm{SumSimTree}(W)+\mathrm{SimTree}(W_k)$；

Next；

$\mathrm{EvlSimTree}(W)=\mathrm{SumSimTree}(W)/i$；

Step 4 设定阈值 ϕ,ξ，对 $\mathrm{SimTree}(W_i)$ 和 $\mathrm{EvlSimTree}$ 进行比较筛选，得出最终扩展词集合 $\mathrm{Sub}(q_{l_j},\phi,\xi)$。

ϕ,ξ 是查询扩展词的有效性阈值(用户可根据该方法在具体测试集上的性能需求确定具体数值)，满足该阈值要求的词即为查询扩展词，至此查询扩展完成。

4 实验与结果分析

4.1 实验过程

1)收集实验数据，这里采用五大标准测试集 ADI，CRAN，CISI，NPL，CACM。这些测试集中包括待检索文档、原始的查询文本，并且给出了一个相关文件，专门标注查询和与该查询相关的文档编号。

2)利用爬虫从网络上爬取大量 Web 文档。

3)利用改进后的 Svmcls 文本分类系统版将 Web 文本按主题分成财经、体育、科技、娱乐、旅游、健康等多个类别，按类别存储成主题分类语料库[12]。

4)使用初始查询检索所有文档，并将检索的所有文档按照相似度递减的顺序排列。取前 N 篇文档作为检索出的文档，建立结果库。

5)用 VC＋＋实现基于语义关联树的分类查询扩展算法。

6)用 SMART 系统进行对比实验。

表1 列出实验所用的工具。

表1 实验所用的工具列表

工具	来自
Web 文本分类	Svmcls 文本分类系统 2.0 版
实验平台	康奈尔大学 SMART 系统
分词工具	中文最大匹配分词 海量 HLSwknlTool
相似度计算	HowNet(http://hownet.kookge.com)

4.2 实验结果及分析

查询扩展是对初始查询的补充和优化，其扩展词数量会影响到检索的精度，扩展词规模过大反而会加入噪声。语义关联树可以灵活高效地生成不同规模的语义空间，通过参数设置的实验表明，在层数 L 不超过 5 层、扩展用词数量不超过 30 时，扩展的效果较好，所以并不是规模越大扩展效果越好。图4 所示为扩展用词数量对查询性能的影响。

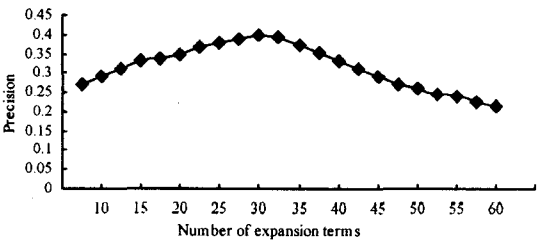

图4 扩展用词数量对查询性能的影响

选择本文提出的语义关联树模型(图中用 SRTM 表示)、LCM、LSI 3 种方法进行查询扩展。词相似度采用 HowNet

(下转第 277 页)

[2] Huang J, et al. Image Indexing Using Color Correlograms [M]. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE): 762-768

[3] PASS G, et al. Comparing images using color coherence vectors [M]. New York, NY, USA, ACM Press: 65-73

[4] Swain M J, Ballard D H. Color indexing [J]. International Journal of Computer Vision, 1991, 7(1): 11-13

[5] Stricker M, Orengo M. Similarity of color images [J]. Proc. SPIE Storage and Retrieval for Image and Video Databases, 1995, 2420: 381-392

[6] 李明，吴艳，吴顺君. 基于小波多通道特征级融合的彩色纹理图像分析 [J]. 光子学报，2004，33(8)：999-1003

[7] Mallat S. A theory for multiresolution signal decomposition: The wavelet representation [J]. IEEE Transactions on PAMI, 1989, 11(7): 674-693

[8] Kokare M, et al. Texture image retrieval using new rotated complex wavelet filters [J]. Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics, 2005, 35(6): 1168-1178

[9] Smith J R, Chang S F. Tools and techniques for color image retrieval[J]. SPIE, 1996, 2670: 426-437

[10] Muller H, Muller W, Squire D M, et al. Performance evaluation in content-based image retrieval: overview and proposals [J]. Pattern Recognition Letters, 2001, 22(5): 593-601

[11] 傅蓉，许宏丽. 基于小波多尺度分析的彩色图像检索方法 [J]. 中国图像图形学报，2004，9(1)：1326-1330

[12] 周明全，耿国华，韦娜. 基于内容图像检索技术 [M]. 北京：清华大学出版社，2007

[13] Chun Y D, Seo S Y, Kim N C. Image retrieval using BDIP and BVLC moments [J]. Circuits and Systems for Video Technology, IEEE Transactions on, 2003, 13(9): 951-957

(上接第 241 页)

距离计算，同时以不选择任何查询扩展算法的原始检索作为对比样本 Base。$SRTM(W,\theta,L,R,M)$取 $\theta=0.1$，$L=5$，$R=2$，$M=10$。基于不同算法检索结果的查全率与查准率如图 5 所示。

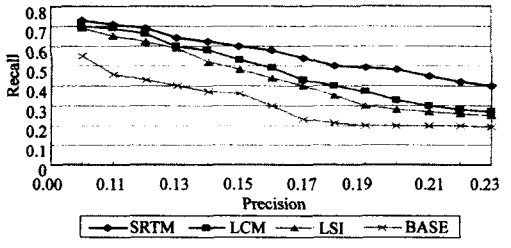

图 5 实验结果对比

由图可以看到，基于语义关联树的查询扩展算法(SRTM)较 LCM 和 LSI 有明显优势，在查准率大于 0.2 时依然能保持较高的召回率。这一方面是由于引入了基于用户交互的主题选取过程，使扩展的范围更加精确；另一方面是因为运用了多个筛选函数，使扩展词筛选更严格，避免了噪声的引入。而基于 LSI 的效果甚至比 LCM 模型更差，这主要是因为 LSI 引入了太多噪声。

假设扩展语料库的规模为 w，初始查询向量 W 的长度为 i，语义关联树的层数为 L，每层最大词数为 M，此时的时间复杂度为 $O(w\times i\times L\times M)$，考虑到实际应用中 W，L，M 都比较小，该时间复杂度趋于 $O(w)$。因此构建语义关联树获取查询扩展词的时间复杂度较低，时效性比较高。

实验表明本文提出的语义关联树不但有效降低了词相似度矩阵的工作量，而且参数可控，能够根据需求灵活地生成不同的语义空间，更重要的是它能够弥补传统方法可能引入大量噪声的缺陷，结合了领域知识也使扩展更有效。

结束语 本文提出了一种结合主题选取与局部反馈进行分类查询扩展的方法，用户输入查询的同时在主题分类语料库中选取与自己意愿最相近的分类，再结合初次检索回的文档构建语义关联树，通过筛选函数的严格筛选，选出最佳的扩展词进行第二次检索，最终得到扩展结果。通过构造语义关联树有效降低了词相似度矩阵计算的工作量。现阶段用户可根据扩展精度的需求人为设置参数，灵活高效地生成不同规模的语义空间，但如何根据不同需求自适应的设置参数还有待于今后进一步研究。根据从网络上下载的 Web 文档按主题进行分类，建立了主题分类语料库，但这个语料库还是不完善的，今后我们会进行进一步的研究，取得更准确的领域知识。在网络检索环境下纳入用户个人偏好，以及搜索结果自动聚类推荐，提供更准确更个性化的查询扩展结果及检索结果将是下一步研究工作的重点。

参 考 文 献

[1] Furnas G W, Landauer T K, Gomez T K, et al. The vocabulary problem in human-system communication[J]. Communication of ACM, 1987, 30(11): 964-971

[2] Deerwester S, Dumain ST, Fumas G W, et al. Indexing by latent semantic analysis[J]. Journal of ACM Transaction on Information System, 2000, 18(1): 79-112

[3] Xu J X, Croft W B. Query expansion using local and global document analysis[C]// Frei H P, Harman D, Schauble P Wilkinson R, eds. Proceedings of the 19th Annual Internation SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 1996: 4-11

[4] Rocchio J J. Relevant Feedback in Information Retrieval. Chapter 14, Prentice-Hall INC, 1997: 313-323

[5] Xu J X, Croft W B. Improving the effectiveness of information retrieval with local content analysis[J]. ACM Transaction on Information System, 2000, 18(1): 79-112

[6] Zhang Cheng-qi, Qin Zhen-xing, Yan Xiao-wei. Association-based segmentation for Chinese-crossed query expansion[J]. IEEE intellingent Informatics Bulletin, 2005, 5(1): 18-25

[7] Qianli Jin, Jun Zhao, Bo Xu. Query expansion based on term similarity tree model[C]// Proceedings of IEEE Natural Language Processing and Knowledge Engineering. 2003: 400-406

[8] Ponte J, Croft W. A language modeling approach to information retrieval[C]// Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrival. 1998: 275-281

[9] 赵军，金千里，徐波. 面向文本检索的语义计算[J]. 计算机学报，2005，12(28)：2068-2077

[10] Shen D, Sun J-T, Yang Q, et al. A comparison of implicit links for Web page classification[C]// WWW'06: Proceedings of the 15th International Conference on Word Wide Web. New York, NY, USA, ACM Press, 2006: 643-650

[11] 桑艳艳，刘培刚，李勇. 基于语义计算的查询扩展的查询优化研究[J]. 情报学报，2007，26(5)：704-710

[12] Wang Y, Hodges J E. Document clustering using compound words. In IC-AI, 2005: 307-313