

基于概念树剪枝的 LCA 查询扩展

李卫疆 王 锋

(昆明理工大学信息工程与自动化学院 昆明 650500)

摘 要 在信息检索应用实践中存在用户表达查询请求不够准确、文档与查询词不匹配以及查询优化等问题。针对这些问题,提出了基于概念树剪枝的 LCA 查询扩展方法,这种混合的查询扩展技术综合了语义和局部上下文分析这两种查询扩展方法,利用 LCA 方法检索得到的扩展词集对语义词典构造的概念树进行适当剪枝,以弥补概念树的不足,并对扩展词候选集用改进的算法重新分配权重。在 TREC 数据集的实验结果表明,与单独基于统计或者基于语义的查询扩展方法相比,基于概念树剪枝的 LCA 查询扩展方法性能有较大提高。

关键词 查询扩展,局部上下文分析方法,概念树,剪枝,相关度算法

中图法分类号 TP391.1 文献标识码 A

Method of Query Expansion Based on LCA Prune Semantic Tree

LI Wei-jiang WANG Feng

(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China)

Abstract Searching and finding out useful information from a mass stock in a very short time is becoming a tough task and many times the user will have to receive a lot of information that may not appear any useful for the users. Problems mainly come from the query which users have provided without enough accuracy, the mismatches of the queries and the expression of the documents, or query optimization. To deal with these problems, this paper proposed a novel hybrid query expansion method which synthesizes the merits of semantic query expansion and local context analysis(LCA). Firstly, we retrieved the documents by LCA method, then used these terms to trim the semantic tree, and calculated the weight of expansion term based on this improved algorithm. We compared the effectiveness of these approaches. And the results show that, although local context analysis has some advantages, the LCA prune semantic tree yields better performance than the techniques on the simple query expansion.

Keywords Query expansion, Local context analysis, Concept tree, Pruning, Relevance algorithm

1 概述

快速即时地处理大量信息,提高信息检索系统的准确率,势在必行。检索的问题主要来自于用户表达查询请求不够准确以及查询优化问题,针对这些问题,研究者们引入了查询扩展概念。

查询扩展技术指的是一种能够全面提高查询完整率的有效方法。综合一些现有的定义,可将查询扩展简单描述为:利用计算机语言学、信息学等多种技术,把与查询相关的词语或者与查询语义相关联的概念以逻辑的方式添加到原查询,得到比原查询更长的新查询,然后检索文档,以改善信息检索的查准率和召回率,这种方法更好地表达了用户的查询请求,对原查询进行扩展弥补用户查询信息不足的缺陷,很好地改善了长期困扰信息检索领域的“词不匹配”问题。研究者们对查询扩展技术进行了深入研究,并取得了可喜成果,目前该技术已成为改善信息检索性能的关键技术。

查询扩展中关键的技术是扩展词表的构造,如何确定扩展词也是研究的一个重要方向。根据不同系统运行的实现方法,查询扩展可以分为不同的运用方法,其中主要包括:通过人工进行的查询扩展,通过系统进行辅助的半自动查询扩展

和全自动的查询扩展。

按其扩展词的不同来源,可将查询扩展方法大致分成4类:基于全局分析的、基于局部分析的、基于关联规则的和基于用户查询日志的查询扩展方法,如图1所示。

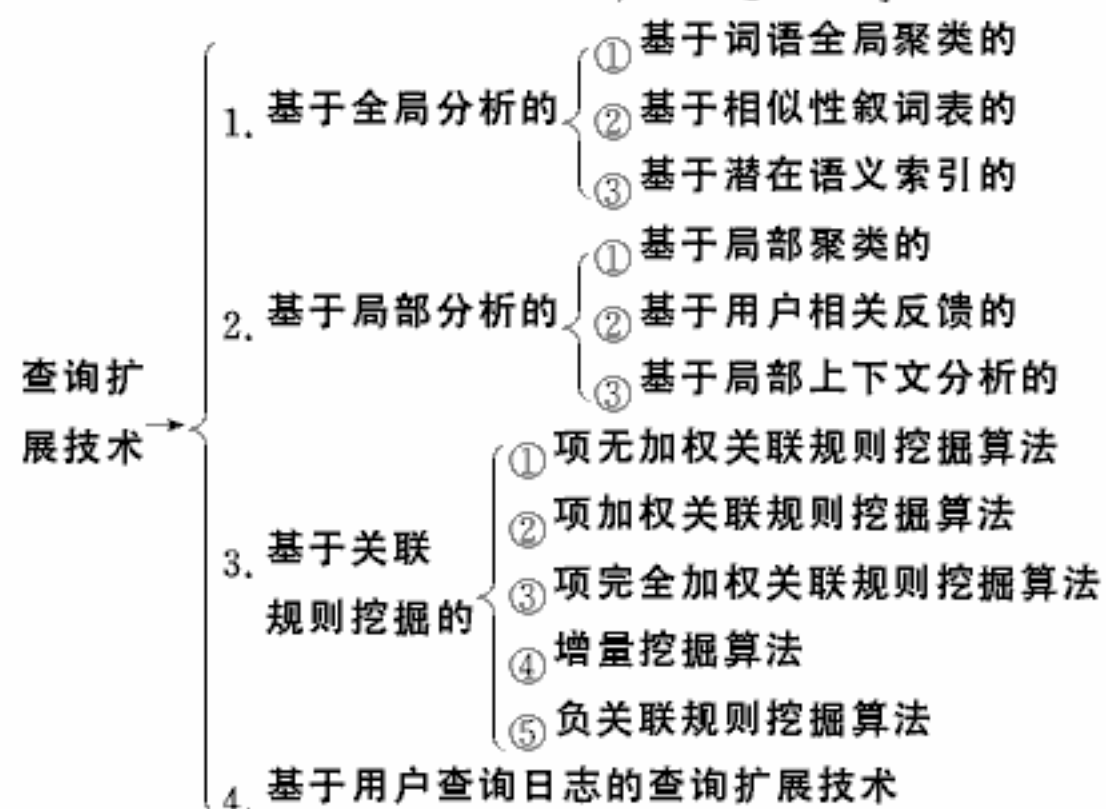


图1 查询扩展技术分类

2 相关工作及研究现状

早在20世纪70年代,为解决词不匹配问题,查询扩展就

本文受国家自然科学基金项目:基于统计机器翻译和自动文摘的查询扩展研究(61363045)资助。

李卫疆(1969—),男,博士,副教授,主要研究方向为信息检索、自然语言处理, E-mail: hrbrichard@gmail.com。

被提出来了。它对用户提供的有关实体属性查询的描述进行语义上同义或近义方面的扩展,利用新的词语来扩展初始的查询,并在扩展的查询中给词语重新加权。

1960年,Marson和Kuhns^[1]提出用高度相关的词语来扩展用户查询。1965年,Rocchio^[2]提出用相关反馈来查询,并研究了在向量空间模型中把查询扩展和词语重新加权相结合的经典技术(应用SMART系统为测试平台),并在随后出版发行。1976年,Robertson和Spark Jones共同提出了概率模型^[3],由于概率模型并没有提供扩展查询的方式,因而应各自进行查询扩展。在1977年,Attar和Fraenkel通过局部聚类的方法改进了原有的查询扩展方法^[4]。

1981年,Wu和Salton^[5]让用户把相关文档中的反馈结果提取出来,然后使用概率公式对扩展词重新加权,最后用这些词语来扩展查询,实验表明了这些方法能够提高检索的性能。

1993年,Qiu和Frei^[6]讨论了使用全局相似性叙词表来进行查询扩展的方法。受Yu、Yang和Salton提出的词语分辨值的影响,Crouch和Yang使用全局统计叙词表来扩展查询^[7]。1996年,Xu和Croft提出了局部上下文分析(Local Context Analysis,LCA)的思想^[8]。

国内外研究的深入和知识库的建设极大地丰富了知识资源。2000年,Xu和Croft提出了局部上下文分析方法^[9],该方法将全局分析的方法融入局部。Kelly和Teevan^[10]在2003年提出了隐含相关反馈模型,他们利用用户查询日志中反映出来的用户查询倾向来对原查询进行扩展,根据相关信息自动地(单指在无用户监督的情况下)进行查询扩展。两年后,Shen等在Kelly和Teevan提出的隐含相关反馈方法的基础上,提出了一种基于统计语言模型的上下文排序检索算法^[11],其根据所有用户点击文档的摘要和用户查询来进行文档的重新排序,并获得了很好的检索效果。

基于全局聚类的方法(Term Clustering)又称作基于统计词典的扩展方法,该算法是1971年由Sparck Jones^[12]提出的,它根据词的共现来对词进行聚类,并用聚类对查询进行扩展。全局聚类方法在实际应用中不是特别理想,主要的缺点在于,它不能处理词的歧义性问题,如果查询检索词有多重意义,则该方法在扩展中会加入与查询条件内容不相关的一些检索词,从而使查询的结果更含糊。Qui与Frei提出了相似性词典^[6],Jing与Croft提出了PhraseFinder^[13],他们都在原假设上做出了改进,由于用多个查询词同时共现的歧义消除效果比仅仅考虑用单个查询词的共现更好,因此扩展词的选择是通过计算与所有的查询词共现来获得的。

局部分析的查询扩展技术的提出至少要追溯到1977年,这个想法是由R. Atter和A. S. Fraenkel^[4]首次详细阐述的。该策略采用两次查询的方法解决扩展问题,初始查询条件在将其提交给搜索引擎得到中间文档集(或段落集)后再对其实行扩展。Xu和Croft^[8]在1990年提出的局部上下文分析方法,成功地解决了全局分析方法中局部聚类方法对 n 敏感和计算量大等问题。

基于关联规则挖掘的查询扩展方法是由Agrawal等人首先提出的,是KDD(Knowledge Discovery in Database)的重要内容。之后,许多研究人员展开了大量的研究,Yahia与

Jaoua^[14]、Brvno和Fronseca^[15]以及Latiri^[16]都先后研究了基于关联规则挖掘的查询扩展,并取得了积极的研究成果。

前面讲的3种查询扩展技术都是从文档集中检索出新的用户查询,而基于用户查询日志的查询扩展技术^[17]考虑的是众多用户使用检索系统时多次“反馈”的结果集,也就是用户查询日志,这对检索来说很有参考价值。

综上,全局分析因为要对整个文档集进行相关处理,当文档集合非常大时,系统在时间和空间上的开销很大,且扩展词语模糊性比较大,只能适合限定领域小范围内的文献检索,难以取得较好的扩展效果。局部分析方法比较好,但当初次查询检索出来的文档与原查询相关度不大时,大量无关的词会被加入进查询,因此准确度会降低。基于用户查询日志的查询扩展适用于查询较小文档集,在这种情况下,其查全率和查准率非常高,但该技术首先需要有一个积累的过程,须有大量的用户查询日志存在,而且不能保证有大量用户有共同的兴趣,且不同时期,他们的兴趣可能也在发生着变化。最后分析一种较新的查询扩展方法——基于关联规则挖掘的查询扩展,该技术从数据挖掘的角度对查询进行扩展,但是其查询扩展的检索效果取决于词间关联规则的质量,即数据挖掘技术,且该技术算法比较复杂。

3 查询扩展

与基于统计的扩展方法相比,根据语言学知识,利用大规模语义词典的概念查询扩展方法不需要大规模语料库的支持,也不需要长时间的训练,直接从语义概念层次上实现扩展。这方面的研究在近些年成为热点,借助语义词典如WordNet,HowNet来构造概念树的方法被广泛使用。但概念树扩展方法会加入很多无用词,随着概念树层数的增加,过多的查询词会使系统做无用的查询,增加计算量,且语义词典的非及时性也会影响查准率。

本文拟通过概念树对各查询词进行语义扩展来解决语义匹配的问题;再利用LCA方法检索扩展词表,以修正扩展词表,计算出同一语料库中与各查询词语义最相关的词,把这些词作为扩展词加到原始查询词中,用以弥补初始查询信息的不足;最后通过对查询词扩展结果集的权重排序,结合原始查询词的位置和扩展词的相关度来计算核心关键词和非核心关键词的权重,让参与查询的扩展词更接近用户的查询意图,从而提高查准率。

3.1 局部上下文分析

采用局部上下文分析方法构造统计扩展词集的方法如下:由检索系统得到与原查询最相关的前 n 个(本文取 n 为100)文档,根据查询把文档分解为固定长度的段落(文中取300个字节为一段)。

对于段落的每个词 W_i ,使用一种与 $tf-idf$ 相似的方法计算出相似度 $sim(q, W_i)$,从中选取与原查询最相关的 m 个(由于此次只是选择统计候选集,因此 m 可选择得稍大点,设 $m=100$)词组作为统计扩展词候选集。本文用式(1)计算初始查询词 q 与扩展词 W_i 之间的相似性:

$$sim(q, W_i) = \prod_{t_i \in q} [\delta + \log(af(W_i, t_i) + 1)] \times idf_c / \log(n)^{idf_i} \quad (1)$$

其中, sim 表示词与该查询之间的相关程度,用于决定与原查

询相关的扩展用词。 W_i 为第 i 个统计扩展词, n 表示排在前面的段落数, 因子 δ 是一个恒定的平滑参数, 为防止 sim 的值为零, 通常是一个较小的因子。 $af(W_i, t_i)$ 是文章中出现的词与查询用词共同出现的频率, 它量化了概念 W_i 和查询词语 q_i 之间的相关性。

$$af(W_i, t_i) = \sum_{j=1}^n p_{fi,j} \times p_{fc,j} \quad (2)$$

这里的 $p_{fi,j}$ 表示词 W_i 在第 j 个段落中出现的频率, 并且 $p_{fc,j}$ 表示查询词 t_i 在第 j 个段落中出现的频率。其中:

$$idf_c = \max(1, \frac{\log_{10} N / np_c}{5}), idf_i = \max(1, \frac{\log_{10} N / np_i}{5}) \quad (3)$$

这里的 N 是整个文档集中段落数, np_c 表示包含概念 W_i 的段落数, np_i 表示包含 q_i 的段落数。

最后根据 $sim(q, W_i)$ 的排序, 将 100 个最相关的概念添加到初始查询 q 中, 便形成了统计扩展词候选集。对得到的统计扩展词候选集中的概念和原始查询紧密程度的结果 $sim(q, W_i)$ 进行归一化, 利用式(4)计算统计扩展词权重:

$$Weight-LCA(W_i) = \frac{sim(q, W_i) - \min(sim(q, W_i))}{\max(sim(q, W_i)) - \min(sim(q, W_i))} \quad (4)$$

3.2 语义概念树的构造

与基于统计学的扩展方法相比, 根据语言学知识, 利用大规模语义词典的概念查询扩展方法不需要大规模语料库的支持, 也不需要长时间的训练, 直接从语义概念层次上实现扩展。这方面的研究在近些年成为热点, 借助语义词典如 WordNet, HowNet 来构造概念树的方法被广泛使用。根据概念之间的语义关系和领域知识的层次结构特点, 构造能反映概念之间语义关系的结构——概念树(Concept Tree)。

基于语义词典为初始查询词构造的概念树, 主要包括原查询词的同义、近义词、上义下义词等, 这类词可记为 $semT = \{semT_1, semT_2, \dots, semT_m\}$ 。定义初始查询词的语义概念森林 $ConcF\{SenseT_1, SenseT_2, \dots, SenseT_n\}$ 。其中 $SenseT_i$ 为第 i 个语义概念树。定义概念森林的目的是解决多义词的问题。例如 carrot 这个词在 WordNet 中有两个意思, 一个是名词胡萝卜, 另一个意思是红发人, 因此 carrot 这个语义森林有两个语义树。对于多义词, 其语义概念树实际上是一个概念森林。如果一个词只有一个意思, 那么其语义森林就退化成了一个语义树。

本文采用语义相似度算法来计算概念 c_i 与根结点的相似度。式(5)是计算概念树中任意概念结点间的相似度的。

$$sim(c_i, c_j) = \frac{Dep(NearestCoAncestors(c_i, c_j))}{Dep(c_i) + Dep(c_j)} \quad (5)$$

其中, $sim(c_i, c_j)$ 为概念树上的任意两个概念 c_i 和 c_j 的语义相似度, $NearestCoAncestors(c_i, c_j)$ 指概念 c_i 和 c_j 的最近共同祖先。

本文为使计算简单, 只考虑概念 c_i 与原始查询概念 $root$ 间的语义扩展词权重 $Weight-Sem(c_i)$ 。可将式(5)改为:

$$Weight-Sem(c_i) = \frac{Dep(NearestCoAncestors(c_i, root))}{Dep(c_i) + Dep(root)} \quad (6)$$

3.3 概念树剪枝策略

对统计扩展词候选集修正概念树, 将满足一定条件的扩

展词加到最终查询扩展词集中。查询扩展词集记为 $Query-Exp(W_1, W_2, \dots, W_i)$ 。

具体步骤如下:

①粗略剪枝。去掉概念树上的叶子结点(如 physical entity, Entity 等)和概念权重趋近于零的概念结点, 因为随着扩展的层数增加, 扩展出的词汇数会急剧增加, 必须控制扩展的深度和广度。这些结点应尽早剪枝, 以求简洁。

②用广度优先策略遍历 N 个语义子树 $STree_i$ 上的每个非根结点, 考察其统计权重 $Weight-LCA$:

if $c_i \in Sta-Candidates$ then $Weight-Sta(c_i) = Weight-LCA(c_i)$

if $c_i \notin Sta-Candidates$ then $Weight-Sta(c_i) = 0$

对于第 i 个子树上的结点, 如果满足条件: $\max(Weight-Sta(c_j)) < r_1, 1 < j \leq n_i$ (其中 n_i 为 $STree_i$ 上的结点个数, 取 $r_1 = 0.05$), 则从概念森林中删除该子树。

③将概念树中满足一定阈值要求的概念结点加到查询扩展词集 $Query-Exp$ 中。

遍历每棵子树 $STree_i$ 上的结点, 将 $Weight-Sta(c_j) \geq r_2, 1 < j < n_i$ 的结点加入查询扩展词集中。其中 n_i 为 $STree_i$ 上的结点个数, 取 $r_2 = 0.35$ 。

④从统计扩展词候选集 $Sta-Candidates$ 中选取与初始查询词相关度较高的扩展词添加到查询扩展词集 $QueryExp-Set$ 中。

若 $Weight-LCA(c_i) \geq r_3$, 取 $r_3 = 0.6$; 同时在 WordNet 中, 该词的定义描述中出现了初始查询词, 且该词不是一个多义词, 则将该词加入到查询扩展词集中。

若 $Weight-LCA(c_i) \geq r_4$, 取 $r_4 = 0.8$, 说明该词与初始查询词极度相关, 则直接将该词加入到查询扩展词集中。这种词或许和原始查询词在语义上不相关, 但在一定时间或特定领域内高度相关, 这种情况在实际检索中更有效。

⑤综合概念权重和统计权重, 对查询选扩展词集 $Query-Exp$ 中各个扩展词重新加权:

$$Weight(c_i) = \frac{(1+\alpha) * (Weight-Sem+\delta) * (Weight-Sta+\delta)}{\alpha * (Weight-Sem+\delta) + (Weight-Sta+\delta)} \quad (7)$$

取 $\delta = 0.1$, 小项 δ 是为了防止权重为零的情况, 调节因子 α 可以调节两种查询扩展方法对综合权值的贡献程度, 具体依实际情况而定, 本文设定 $\alpha = 1$ 。

4 实验与分析

本文在 TREC2&3 的 Associated Press 数据集 (AP880 212-AP901231) 上对所提出的查询扩展方法进行了实验, 表 1 列出了测试集合。

表 1 实验数据统计信息

数据集	值
内容	AP in TREC 2&3
包含文档数目	158240
平均文档长度	261
词语数目	41316279
平均每篇文档包含唯一词 (不重复词) 数目	194

实验采用 WordNet 2.1 提供的接口 WordNet.Net 来构造概念树。

表 2 列出了其中的 3 个初始查询词和其扩展词的实例。

表2 初始查询词与其扩展词实例

初始查询词	语义扩展词及权重	统计扩展词及权重	按权重排序的综合扩展词及权重
monitor	proctor(0.333), supervisor(0.250), admonisher(0.333),...	monitor consultation(0.328), monitor Grope(0.334), supervisor(0.300),...	monitor Grope(0.163), proctor(0.162), supervisor(0.373),
copyright	document(0.333), procure(0.333),secure(0.333) papers(0.250),...	registration(0.425), copyright symbol(0.341), copyright low(0.157),...	registration(0.168), copyright symbol(0.163), document(0.223),
image	picture(0.333), prototype(0.250), trope(0.250),...	winimage(0.369), softimage(0.215), imagemagick(0.215),...	picture(0.162), prototype(0.210), winimage(0.227),...

首先要确定结果集返回为多少个时,查询扩展系统可以达到最好效果。考虑到用户在查看结果时,第一个页面大概有10多个检索结果,我们分别取10,20,30,40为返回结果集的个数,使用TREC2&3(AP)数据集进行查询,在扩展词个数不同的情况下计算检索结果的平均查准率,实验结果见表3。

表3 不同扩展词个数下检索结果的查准率

扩展词个数	10	20	30	40
平均准确率	0.357	0.448	0.439	0.389

从表3的实验结果可知,当输出20个左右扩展词时,检索结果的平均查准率最高。当扩展词个数过小或者过大时,都不能得到较高的查准率。本实验将查询选扩展词集的前20个扩展词输出,返回给用户,以便更好地提供他们所需要的信息。

接下来的实验是对非扩展查询方法、局部上下文分析方法和基于剪枝概念树的LCA查询扩展方法进行测试对比。本文采用以下指标:召回率(Recall)、查准率(Precision)、F-measure和 $Pr@n$ (n 取10),用这些性能评价指标来测试本文所提出的基于剪枝概念树的LCA查询扩展方法的查询质量,测试结果如表4所列。

表4 3种检索技术实验结果对比

检索方式	Precision	Recall	F-measure	Pr@10
传统非扩展检索方法	0.397	0.293	0.337	0.346
LCA方法	0.309	0.443	0.364	0.397
LCA剪枝概念树	0.451	0.441	0.446	0.475

从实验结果看,无论是查准率还是召回率方面,本文提出的混合查询扩展方法相对于不加扩展的tf-idf算法以及基于局部上下文分析算法在性能上都有一定的提高。

对表4中的召回率、准确率、F-measure指数和 $Pr@10$ 构建折线图,如图2所示。

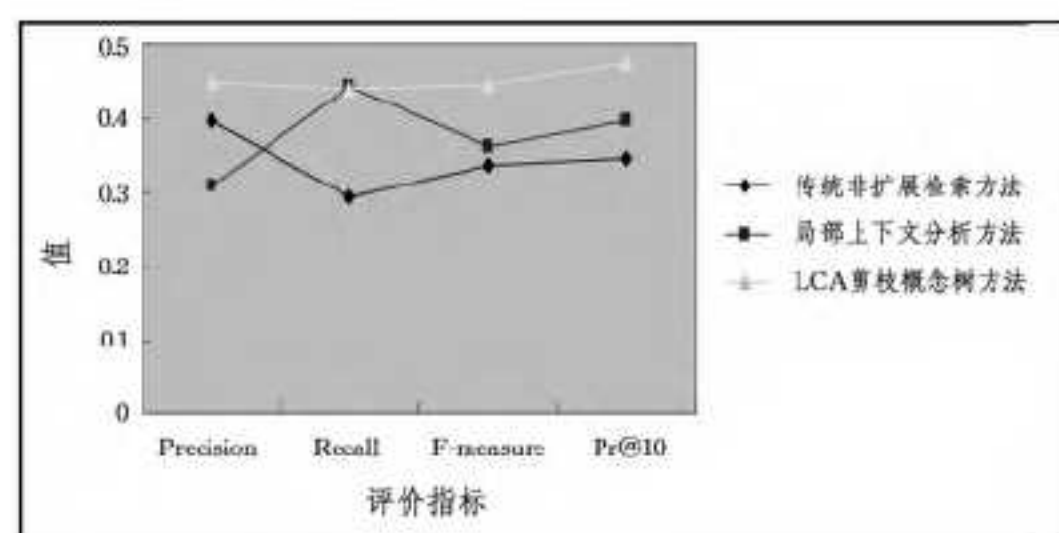


图2 3种系统性能评估折线图

从图2中更直观地看到,在大数据集文档中,局部上下文分析的查询扩展系统提高了检索性能,但该方法在提高召回

率的同时却因为数据集很大而引入了较多的不相关扩展词而降低了查准率。而基于概念树剪枝的LCA查询扩展技术在提高召回率的同时,查准率也有所提高,选出的20个扩展词既能更准确地反映用户的需求,在结果集中相关文档数又相对较高,也提高了前10条查询扩展结果集的准确率,这表明该方法在实际应用中有较好的实用价值。

结束语 本文提出的基于概念树剪枝的LCA查询扩展技术,首先使用局部上下文分析方法做局部扩展查询,然后使用WordNet对原查询进行扩展生成语义概念树,最后在综合这两种方法的基础上再做查询扩展得出最终扩展词集。通过实验比较了传统非扩展检索方法、基于上下文分析查询扩展系统和基于概念树剪枝的局部上下文分析查询扩展系统的召回率、准确率、F-measure指数和 $Pr@10$ 。可以看出,在检索性能上,基于概念树剪枝的LCA查询扩展系统都高于单独使用概念查询扩展和局部上下文分析查询扩展系统。

参考文献

- [1] Sanjuan E, Ibekwe-Sanjuan F. Combining language models with NLP and interactive query expansion[C] // Proceedings of the Focused Retrieval and Evaluation, and 8th International Conference on Initiative for the Evaluation of XML Retrieval. Berlin, Heidelberg: Springer-Verlag, 2010: 122-132
- [2] Rocchio J J. Document Retrieval Systems-Optimization and Evaluation[D]. Harvard, 1966
- [3] Robertson S E, Jones K S. Relevance weighting of search terms[J]. Journal of the American Society for Information Science, 1976, 27(3): 129-146
- [4] Attar R, Fraenkel A S. Local feedback in full-text retrieval systems[J]. Journal of the Association for Computing Machinery, 1977, 24(3): 397-417
- [5] Wu H, Salton G. The estimation of term relevance weights using relevance feedback[J]. Journal of Documentation, 1981, 37(4): 194-214
- [6] Qiu Y, Frei H P. Concept based query expansion [C] // Proceeding of the 16th annual international ACM SIGIR Conference on Research and Development in Information Retrieval. 1993: 160-169
- [7] Crouch C J, Yang B. Experiments in automatic statistical thesaurus construction[C] // Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, 1992. ACM, 1992: 77-88
- [8] Xu J, Croft W B. Query expansion using local and global document analysis[C] // Proceeding of Annual International ACM Sigir Conference on Research & Development in Information

Retrieval. 1996;4-11

- [9] Xu J, Croft B. Improving the effectiveness of information retrieval with local context analysis[J]. ACM Transaction on Information Systems, 2000, 18(1): 79-112
- [10] Kelly D, Teevan J. Implicit feedback for inferring user preference: a bibliography[C]// ACM SIGIR Forum. 2003; 18-28
- [11] Shen X, Tan B, Zhai C. Context-sensitive information retrieval using implicit feedback[C]// Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. 2005; 43-50
- [12] Jones K S. Automatic keyword classification for information retrieval[M]. 1971
- [13] Jing Y F, Croft W B. An association thesaurus for information

retrieval[C]// RIAO Conference Proceedings. 1994; 146-160

- [14] Yahia S B, Jaoua A. Discovering knowledge from fuzzy concept lattice[M]// Data mining and computational intelligence. Springer, 2001; 167-190
- [15] Fonseca B M, Golgher P B, De Moura E S, et al. Discovering search engine related queries using association rules[J]. Journal of Web Engineering, 2003, 2(4): 215-227
- [16] Latiri C C, Yahia S B, Chevallet J P, et al. Query expansion using fuzzy association rules between terms[J]. Proceedings of JIM, 2003
- [17] Cui H, Wen J, Nie J, et al. Query expansion by mining user logs [J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(4): 829-839

(上接第 461 页)

表 3 频繁一项集

名称	事务数	url 频繁数	query 频繁数
data1	1×10^7	1782	7231
data2	2×10^7	3461	13282
data3	3×10^7	5113	20891
data4	4×10^7	6922	29182
data5	5×10^7	8211	37261

表 4 频繁二项集

名称	事务数	url 频繁数	query 频繁数
data1	1×10^7	688	3892
data2	2×10^7	1361	7912
data3	3×10^7	1929	1125
data4	4×10^7	2416	15271
data5	5×10^7	3213	17828

搜狗实验室的搜索引擎日志中, 设置 $minsup=0.001$ 时, 从表 3 可以看出, 频繁一项集随着事务数的增加而基本上呈现正相关, 同时 query 频繁集明显多于 url 频繁集合。从表 4 可以看出, 频繁二项集比频繁一项集明显减少, url 减少的幅度大于 query 减少的幅度。找到了频繁一项集和频繁二项集, 就可以根据关联规则得经典应用, 进行推荐。

通用搜索引擎的数据较为稀疏, 垂直型搜索引擎, 例如电商、团购网站, 其数据会更加的稠密。频繁集合挖掘的效果与数据稠密性紧密相关。

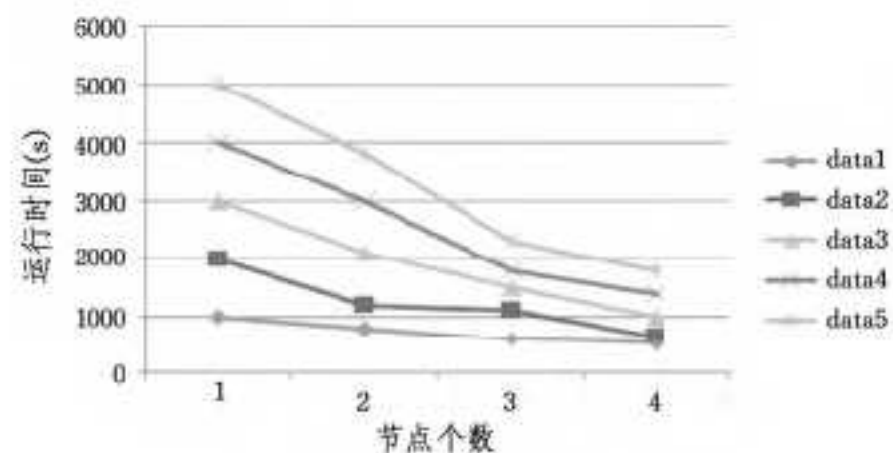


图 6 不同节点数的运行时间对比

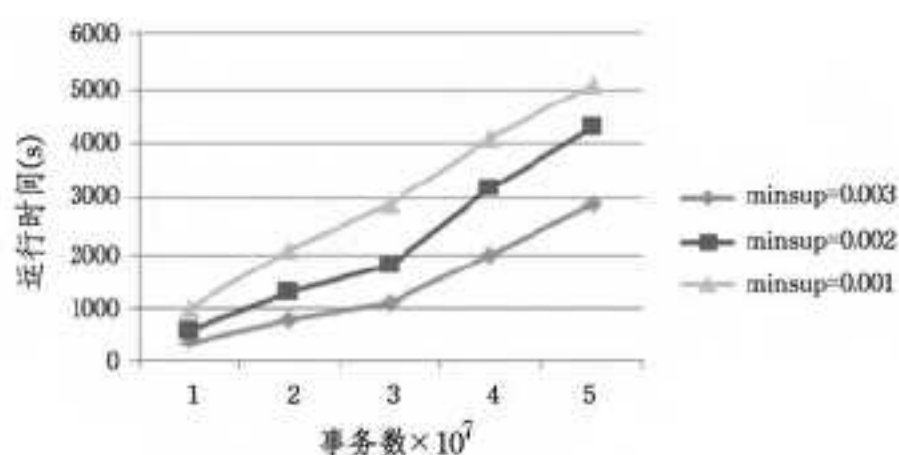


图 7 不同事务数的运行时间对比

从图 6 可以看出, 随着节点数增加, 算法的运行效率会得到很大提高, 数据规模越大效率提升越明显, 在第一个和第二个节点的提升较快, 后续节点提升变缓。从图 7 可以看出, 在不同的支持度下, 运行时间随着事务数的增加, 几乎呈线性增加。

因此, 可根据数据集的大小选择合适的节点数目并设置合适的支持度阈值。

结束语 在搜索引擎中根据 Web 日志, 寻找 query 和 url 的频繁集, 有利于发现用户获取信息的习惯, 帮助站点提供更好的个性化服务。本文设计了基于 Hadoop 的并行 FP-growth 算法数据挖掘模型, 有利于性能、架构优异性的迁移。通过对日志的处理效率和结果分析, 发现 Hadoop 分布式架构能够很好地满足大规模日志处理的需求。当数据量越大时, 并行 FP-growth 算法的运算性能提升越明显。

参考文献

- [1] 董志安, 吕学强. 基于百度搜索日志的用户行为分析[J]. 计算机应用与软件, 2013, 7(2): 17-20
- [2] 陈富赞, 刘青, 李敏强, 等. 一种基于会话聚类算法的 Web 使用挖掘方法[J]. 系统工程学报, 2012, 1(7): 129-136
- [3] 刘建国, 周涛, 汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展, 2009, 1(10): 1-15
- [4] 蓝祺花, 吴博. 频繁项集挖掘算法研究[J]. 计算机与现代化, 2009, 3(9): 60-65
- [5] 吕婉琪, 钟诚, 唐印洪, 等. Hadoop 分布式架构下大数据集的并行挖掘[J]. 计算机技术与发展, 2014, 24(1): 22-25, 30
- [6] 周诗慧, 殷建. Hadoop 平台下的并行 Web 日志挖掘算法[J]. 计算机工程, 2013, 6(3): 43-46
- [7] 张俊, 李鲁群, 周熔. 基于 Lucene 的搜索引擎的研究与应用[J]. 计算机技术与发展, 2013, 23(6): 230-232
- [8] Naganathan E R, Narayanan S, Kumar K R. FP-Growth Based New Normalization Technique for Subgraph Ranking[J]. International Journal of Database Management Systems, 2011, 31
- [9] Jiao Ming-hai, Yan Ping, Jiang Hui-yan. Research and application on Web information retrieval based on improved FP-growth algorithm[J]. Wuhan University Journal of Natural Sciences, 2006, 11(5): 1065-1068
- [10] 章志刚, 吉根林. 一种基于 FP-Growth 的频繁项集并行挖掘算法[J]. 计算机工程应用, 2014, 2(2): 103-106