ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.5, May 2010, pp.1083−1097
doi: 10.3724/SP.J.1001.2010.03477

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于用户搜索意图的 Web 网页动态泛化*

王大玲+, 于 戈, 鲍玉斌, 张 沫, 沈 洲

(东北大学 信息科学与工程学院,辽宁 沈阳 110004)

Dynamically Generalizing Web Pages Based on Users' Search Intentions

WANG Da-Ling+, YU Ge, BAO Yu-Bin, ZHANG Mo, SHEN Zhou

(School of Information Science and Engineering, Northeastern University, Shenyang 110004, China)

+ Corresponding author: E-mail: dlwang@mail.neu.edu.cn, http://www.neu.edu.cn

Wang DL, Yu G, Bao YB, Zhang M, Shen Z. Dynamically generalizing Web pages based on users' search intentions. *Journal of Software*, 2010,21(5):1083−1097. http://www.jos.org.cn/1000-9825/3477.htm

Abstract: In this paper, based on the classification of search intentions, required information for every intention is further analyzed, and a dynamic model of generalizing Web pages based on the intentions is proposed, which dynamically creates a concept hierarchy concerning snippets, keywords, navigation types, and document formats for searched Web pages, and provides further retrieval navigations for different intentions by generalizing content, format and type of the pages, to return more relevant results to the intentions. Comparing with related work, this paper does not focus on users' intentions themselves, but on the creation of generalization models based on users' intentions and implementation for Web pages generalizing. Experimental results show that the generalization model can automatically acquire users' search intentions under navigation, and provide relevant results and further navigation based on the intentions.

Key words: Web page generalizing; user's intention; dynamically modeling; search navigation; text mining

摘 要: 基于目前对用户搜索意图的分类,进一步分析了每种用户意图的信息需求,提出了基于用户搜索意图的Web网页动态泛化模型,为搜索的Web网页动态地建立文档片段、关键词、导航类型、文档格式之间的概念层次,通过网页内容、类型和格式的泛化为不同的访问意图提供进一步的搜索导航,从而返回与搜索意图更相关的结果.与相关工作对比,重点并非获取用户意图,也不是对用户意图分类,而是基于用户搜索意图的Web网页动态泛化模型的建立及Web网页泛化过程的实现.实验结果表明,该泛化模型不仅能够通过导航自动获取用户搜索意图,而且能够基于该意图提供相关搜索结果以及进一步的搜索导航.

关键词: 网页泛化;用户意图;动态建模;搜索导航;文本挖掘

中图法分类号: TP393 **文献标识码**: A

目前,关于用户搜索意图的研究主要包括两个方面:一是使搜索引擎提供更好的交互功能,显式或隐式地获取用户意图;二是对用户意图尽可能准确地分类.然而,搜索引擎真正的目的应该是为不同的搜索意图提供不同的信息,而获取和分类用户意图仅仅是手段,但目前的研究大多局限于用户意图本身而并非如何满足这些意图.

* Supported by the National Natural Science Foundation of China under Grant No.60573090 (国家自然科学基金)
Received 2008-04-24; Revised 2008-08-07; Accepted 2008-10-09; Published online 2009-11-05

基于此,本文根据目前用户搜索意图研究中的一种较为流行的分类方法,进一步分析每类意图的信息需求以及各类意图之间的关系,提出一种基于用户搜索意图的 Web 网页动态泛化模型,通过网页内容、格式和类型的泛化为不同的访问意图提供进一步的搜索导航,从而返回与搜索意图更相关的结果.

本文第 1 节介绍目前比较流行的一种用户意图分类方法以及本文对此所作的进一步分析.第 2 节描述本文提出的基于用户搜索意图的 Web 网页动态泛化模型.第 3 节介绍基于用户搜索意图的 Web 网页动态泛化模型的建立及在该模型上的相关操作.第 4 节给出实验结果及对模型的评价.第 5 节综述相关工作并与本文工作进行比较.第 6 节是结论和进一步的研究.

1 用户搜索意图

向搜索引擎提交查询的用户均有其潜在的搜索意图/获取和分类用户意图是满足用户需求的前提.目前,一种比较流行的对用户搜索意图的分类方法是将其分为 3 类[1]:

(1) 导航型(navigational):寻找某类特殊站点,这类站点能够为用户提供该站点上进一步的导航操作.

(2) 信息型(informational):寻找 Web 站点上某种以静态形式存在的信息,这是用户通常的一种查询.

(3) 事务型(transactional):寻找某类特殊的站点,这类站点的信息能够直接被用户下载或做进一步的在线操作,如购物、玩游戏等.

在近年来关于用户搜索意图的研究中,文献[2]将上述 3 类用户意图进行了更为细致的分类,将信息型意图细分为 Directed,Advice 等 5 个子意图,将事务型意图又细分为 Download,Entertainment 等 4 个子意图;文献[3]将上述 3 类用户意图的每一项均划分为 commercial 和 non-commercial 意图;文献[4,5]将上述 3 种意图作为类标签,研究准确的分类算法;文献[6]通过分析用户在搜索结果中的鼠标移动轨迹来推断用户的导航型和信息型意图.此外,还有一些用户意图分类的研究不是以上述 3 种类型为基础,如文献[7]分类用户搜索意图为 product intent 和 job intent,文献[8]将用户意图分类为显式和隐式,但这些研究也是限于用户意图分类本身.

对于搜索引擎,获取和分类用户搜索意图的真正目的是针对不同的意图提供不同的信息,以满足用户的个性化需求.因此,无论用户意图如何分类,搞清其每种意图的信息需求是关键.就上述分类而言,导航型意图意味着用户想要获取能够提供导航信息的这类特殊站点,例如主页或包含足够导航信息的非主页;事务型意图意味着用户想要知道每个返回结果的格式,仅需要那些能够满足其在线操作的格式所对应的结果;信息型意图意味着用户在进行最通常的一种查询,并且在查询返回的结果中寻找其感兴趣的内容.就上述各类意图所需的信息而言,为满足导航型意图,应将返回结果是否导航、以何种方式导航的信息提供给用户;为满足事务型意图,应将返回结果的格式提供给用户;为满足信息型意图,在基于关键字匹配的查询中,应能够从返回结果中抽取新关键字以供用户作进一步的选择.

现在,在许多商务搜索引擎中,返回结果的 URL 或标题中能够标识该结果是否为主页,从 URL 标识的格式上也能初步判断其是否能够导航.但在返回结果中并非所有能够导航的结果都能被排在 Top-N 中,因此导航型意图不能直接得到满足;这类搜索引擎也能在返回结果的 URL 中包含对应结果的格式,但同样地,不是所有符合用户要求格式的结果均能被排在 Top-N 中.虽然像 Google 这样的搜索引擎能够提供一系列的格式列表供用户选择,只返回符合用户选定格式的结果,但是,由于提供的列表是静态的,不能随内容变化,选择不当会造成无返回结果,因此,事务型意图也不能得到很好的满足.

就 3 种意图的关系而言,一个用户开始上网搜索时,其目的可能并不明确,因此信息型意图只是一个初始的选择.当他发现感兴趣的内容时,再做进一步的操作.例如,一个从事“信息检索”方向的研究人员在上网搜索时输入“信息检索”,其意图是信息型.在返回的网页中,看到某个学者或者某个课题组对此颇有建树,于是希望进入该学者或者该课题组的主页,进一步了解他们的情况,这时,信息型意图转变成了导航型意图.或者,该研究人员在输入“信息检索”后返回的网页中,看到某一篇论文的题目和摘要令其感兴趣,于是去找该文章的pdf等可以下载阅读的格式以便下载,这时,信息型意图就转变成了事务型意图.由此可见,信息型意图是其他两类意图的源头,或者说,其他两类意图是信息型意图的目的.

一般地,用户开始并不十分明确自己的搜索意图,更重要的是,用户上述的搜索过程是根据返回结果而动态变化的.因此,如何通过导航获取用户意图并最终满足该意图至关重要.基于上述关系,如何满足用户的信息型意图成为关键问题,只有在满足信息型意图的前提下,才能更好地满足用户的导航型和事务型意图.对于信息型意图,目前一个主要问题是,无论哪个用户,只要输入相同的查询词,搜索引擎就返回相同的结果.

针对上述问题,目前一些商务搜索引擎提供的一些功能在一定程度上能够满足其中某种搜索意图的信息需求,如允许用户选择结果的格式、在当前结果中继续搜索等(可以将其视为搜索导航).但是,一方面,目前还没有搜索引擎能够提供同时选择结果类型、格式及新的查询词的搜索导航,另一方面,目前搜索引擎提供的搜索导航信息不能随着返回结果的变化而动态改变.

数据泛化是一种将相对低层次的值用较高层次的概念置换来汇总数据的技术[9].该技术为用户提供了全面考察数据性质的途径.

根据前面分析的搜索意图的信息需求以及各意图之间的关系,我们认为,对搜索引擎返回结果(Web 网页)动态地建立泛化模型并通过网页内容、格式和类型的泛化提供面向搜索意图的导航,最终满足不同意图的信息需求是一条有效的解决途径.该 Web 网页泛化模型应包括如下几个方面:

(1) 内容泛化:提供新的关键字列表,这些关键字尽可能准确地描述返回结果,满足信息型意图.

(2) 类型泛化:提供导航类型列表,表中包括所有返回结果的导航类型,满足导航型意图.

(3) 格式泛化:提供格式列表,表中包括所有返回结果的文档格式,满足事务型意图.

(4) 根据 3 种意图的关系,基于上述 3 个列表,提供关于新关键字、类型和格式的进一步选择,即搜索导航,从而进一步获取用户意图并返回与该意图更相关的新结果.

基于此,本文提出一种由 Web 网页的文档片段、关键词、类型、格式之间的概念层次构成的 Web 网页泛化模型.当用户搜索时,对搜索结果的 Web 网页动态地建立泛化模型,泛化结果为用户提供进一步的搜索导航,根据新的搜索要求对返回的网页进行筛选并进一步泛化,从而返回更相关的结果和新的导航信息.

2 问题描述

2.1 Web网页、关键字、类型、格式与搜索意图

用户向搜索引擎提交一个关键字 kw 后,搜索引擎返回一个文档集合.设 S 为返回结果的 Top-m 个文档,即 $S=\{s_1,s_2,\ldots,s_i,\ldots,s_m\}$,其中,文档 $s_i(i=1,2,\ldots,m)$包括标题(title)、文档片段(snippet)(包含提交的关键字的一些句子)和超链接(URL)[10].通过 URL 可以得到每个 s_i 的内容,并且获得其格式和类型.将标题(title)、文档片段(snippet)、超链接(URL)、内容(content)、格式(format)和类型(type)分别记为 $s_i.title$,$s_i.snippet$,$s_i.url$,$s_i.content$,$s_i.format$ 和 $s_i.type$.其中,关键字(kw)可以有其子关键字,记为 $kw_1,kw_2,\ldots,kw_i,\ldots,kw_{n1}$.而且 kw_i 还可以进一步细化,分为 kw_{i1},$kw_{i2},\ldots,kw_{ij},\ldots,kw_{il}$,依此类推.这里将 Web 网页的内容、文档片段、格式、类型及其中抽取的关键字描述成如图 1 所示的概念层次图.

图 1 的概念层次图虽然在形式上不同于通常的概念层次树[9],但综观全图,文档片段(snippet)是网页(Web page)的描述,关键字、格式、类型又可以从文档片段中抽取,关键字本身又具有层次结构,整个图具有“愈低层愈具体、愈高层愈泛化”的特征,因此,可将图 1 视为 Web 网页的泛化过程.其中,从文档片段中抽取关键字、格式和类型的过程(extract)可以视为概念提升,也是泛化建模过程,根据关键字、类型或格式搜索相关文档的过程(search)视为概念下降.

针对图 1,从概念下降的角度,如果采用第 i 层的概念搜索,返回的结果是 S_i,采用第 i+1 层的概念搜索,返回的结果是 S_{i+1},采用第 i+2 层的概念搜索,返回的结果是 S_{i+2},则 $S_{i+2}\subseteq S_{i+1}\subseteq S_i$ 且$|S_{i+2}|\leq|S_{i+1}|\leq|S_i|$.在此情况下,如果用户首先选择第 i 层的概念,然后选择第 i+1 层的概念,再依次选择第 i+2 层的概念,…,这些选择不仅使返回的结果越来越少,而且越来越接近用户的需求.从概念提升的角度,在返回的文档片段中抽取关键字(内容泛化)即可返回新的关键字列表,从而满足信息型意图的需求;在返回的文档片段中抽取文档格式(格式泛化)即可返回格式列表,从而满足事务型意图;在返回的文档片段中抽取导航类型(类型泛化)即可返回类型列表,从而满足导航型

意图.因此,概念提升的结果为用户提供搜索导航,而概念下降的过程则使搜索结果更接近于用户搜索意图.

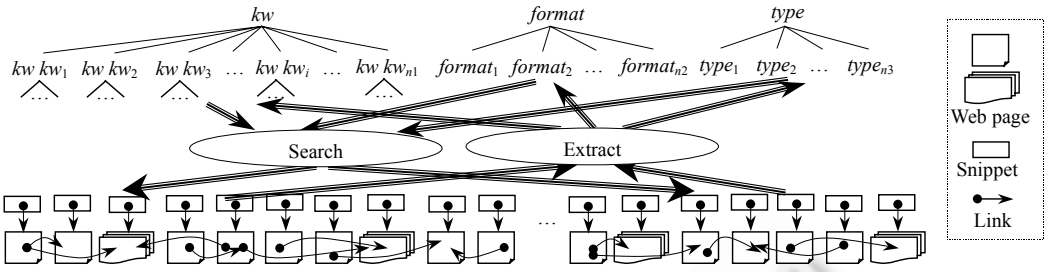


Fig.1 Concept hierarchy graph for Web pages about their content, snippet, format, type, and keyword

图 1 Web 网页的内容、片段、格式、类型及其关键字的概念层次图

2.2 Web网页泛化模型

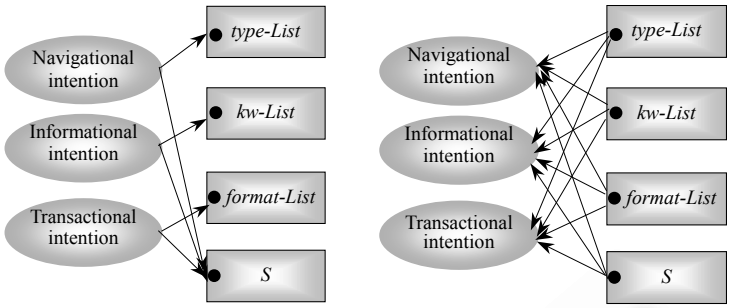
如第 2.1 节所述,将 Web 网页的内容、片段、格式、类型及其中抽取的关键字描述的概念层次图视为 Web 网页泛化模型,将泛化过程视为概念提升,将搜索相关文档的过程视为概念下降.这样,用户向搜索引擎提交一个关键字 *kw* 后,泛化建模的过程是:对于 *kw* 返回的前 *m* 个查询文档集合 *S*,从每个 $s_i \in S(i=1,2,...,m)$中获得 $s_i.type$ 和 $s_i.format$,并从 *S* 中抽取新关键字,得到的关键字集合、$s_i.type$ 集合和 $s_i.format$ 集合作为 *S.kw*,*S.type* 和 *S.format* 列表,这些列表中的信息为用户提供进一步的搜索导航,并允许用户据此导航信息提交新的搜索.显然,每个 kw_i, $type_k$ 和 $format_j(i=1,2,...,n1,j=1,2,...,n2,k=1,2,...,n3)(n1,n2,n3$ 分别是关键字、格式和类型的数目)分别对应一个返回文档集合 $S' \subseteq S$.设 kw_i,$type_j$ 和 $format_k$ 对应的 *S*′分别为 S_i,S_j 和 S_k,则 kw_i 被提交后,返回 S_i 及其从 S_i 中抽取的关键字、类型和格式集合.$type_j$ 与 $format_k$ 同理.而 $kw_i \cup type_j$,$kw_i \cup format_k$,$type_j \cup format_k$ 乃至 $kw_i \cup type_j \cup format_k$ 被提交后可返回 $S_i \cap S_j$,$S_i \cap S_k$,$S_j \cap S_k$ 及 $S_i \cap S_j \cap S_k$.

根据该思想,本文建立的 Web 网页泛化模型 $G_{Model}(Web,search\text{-}item)$描述为:$G_{Model}(Web,search\text{-}item)=\langle S,kw\text{-}List,type\text{-}List,format\text{-}List\rangle$.其中,$search\text{-}item=\{kw,format,type\}$.由于信息型意图是另外两种意图的源头,因此开始时,模型的输入包括搜索引擎所有内容的集合 *Web*,而 *page-item* 仅为一个关键字 *kw*,输出包括由 *kw* 从 *Web* 获得的结果集合 *S*、从 *S* 中抽取的关键字集合 *kw-List*、导航类型集合 *type-List* 和文档格式集合 *format-List*,而 *S*,*kw-List*,*type-List* 和 *format-List* 为搜索导航信息.这样,模型新的输入 $Web'=S,search\text{-}item'=\{kw_i \in kw\text{-}List(1 \le i \le n1),type_j \in type\text{-}List(1 \le j \le n2),format_k \in format\text{-}List(1 \le k \le n3),kw_i \cup type_j,kw_i \cup format_k,type_j \cup format_k,kw_i \cup type_j \cup format_k\}$,新的输出则为从搜索引擎获得的结果集合 $S' \subseteq S$ 以及从 *S*′抽取的 $kw\text{-}List' \subseteq kw\text{-}List$, $type\text{-}List' \subseteq type\text{-}List$ 和 $format\text{-}List' \subseteq format\text{-}List$.

不失一般性,对于模型的任一输入 *Web* 和 *search-item*,建模过程包括从 *Web* 中抽取满足 *search-item* 的子集 *S*、从 *S* 中抽取的关键字集合 *kw-List*、类型集合 *type-List* 和格式集合 *format-List*.

将一个意图视为一个请求,将 *S*,*kw-List*,*type-List* 和 *format-List* 视为 3 种资源,采用资源分配图描述各意图与其实现的关系如图 2 所示.

就搜索意图本身而言,导航型意图请求 *S* 和 *type-List*,信息型意图请求 *S* 和 *kw-List*,事务型意图请求 *S* 和 *format-List*(如图 2(a)所示).然而,根据信息型意图与其他两类意图的关系,无论基于哪种意图,*S*,*type-List*,*kw-List* 和 *format-List* 均被输出(如图 2(b)所示).同时,由于 *S* 随着输入的变化不断发生变化,*type-List*,*kw-List* 和 *format-List* 也将随之动态变化,所以建模过程是动态的和在线的,故称 $G_{Model}(Web,search\text{-}item)$为 Web 网页动态泛化模型.



(a) Input (resource is requested) (b) Output (resource is held)

(a) 输入(资源请求) (b) 输出(资源占有)

Fig.2 Information requirement for every intention and implementation

图 2 每种意图的信息需求及其实现

3 Web 网页泛化算法

如前所述,Web 网页建模 $\mathcal{GModel}$(*Web*,*search-item*)是从 *Web* 中获取满足 *search-item* 的结果 *S*,并从 *S* 中获取 *kw-List*,*type-List* 和 *format-List*.$\mathcal{GModel}$(*Web*,*search-item*)的详细描述如 Algorithm $\mathcal{GModel}$所示.

Algorithm $\mathcal{Gmodel}$.

Input: document set *Web*, *search-item*={*kw*,*type*,*format*}, θ_{local}, θ_{global}.

Output: *S*, *kw-List*, *type-List*, *format-List*.

Description:

1. Go to search engine for obtaining *S* according to *search-item.kw*; // default is *Web*
2. *kw-List*=*type-List*=*format-List*={};
3. For every $s \in S$
4. *s.format*=$\mathcal{Ef}$(*s*); //Extract format of *s*
5. *s.type*=$\mathcal{Et}$(*s*); //Extract type of *s*
6. If *s.type*=*search-item.type* and *s.format*=*search-item.format* Then // default is True
7. *type-List*=+*s.type*;
8. *format-List*=+*s.format*;
9. $\mathcal{Ek}$(*s*,fw_{local}-*List*,fw_{global}-*candidate-List*);
10. *kw-List*=+fw_{local}-*List*;
11. Else delete *s* from *S*
12. End If;
13. End For;
14. For every $w \in fw_{global}$-*candidate-List*
15. If $tf(w) \geq \theta_{global}$ Then *kw-List*=+*w*
16. End If;
17. End For;

在 Algorithm $\mathcal{GModel}$ 中,初始的输入默认为整个搜索引擎获得的内容,而此后该项输入就是上次返回的文档集合 *S*.输入还包括用户根据搜索导航信息给出的选择(这种选择隐式地代表了用户的搜索意图)以及为抽取关键字而给定的局部频繁阈值θ_{local}和全局频繁阈值θ_{global},其意义将在第 3.1 节详细加以介绍.输出则是返回片段集合 *S* 和 3 个列表 *kw-List*,*type-List* 和 *format-List*.如果 *search-item* 中包含了关键字条件,则应用搜索引擎 API 获取满足关键字条件的文档子集 *S*(第 1 行),然后抽取 *S* 中每个文档 *s* 对应内容的格式和类型(第 3~5 行,在第 3.2

节、第 3.3 节将对该过程进行详细描述).如果 *search-item* 包含了类型或格式条件,并且 *s* 的格式和类型满足条件,则从 *s* 中抽取局部关键字和候选全局关键字(第 9 行、第 10 行),否则 *s* 不作为返回文档(第 11 行).当返回的文档集合 *S* 全部处理完毕时,再确认全局关键字(第 14 行、第 15 行).抽取局部关键字和全局关键字的细节将在第 3.1 节中详细加以描述.

当然,对于 Google 这样具有高级搜索功能的搜索引擎,如果用户在后续搜索中根据导航信息选择了结果的格式,则算法的相关部分(第 4 行)可简化.

3.1 网页内容泛化算法

在 Algorithm $\mathcal{GM}odel$ 中,当得到了返回文档片断集合 *S* 后,主要的处理包括网页内容泛化、导航类型泛化和文档格式泛化,本节详细介绍网页内容泛化、即从 *S* 中获取关键词的过程.

对于每个返回文档 $s\in S$,由于用户能够直接看到的仅仅是 $s.title$,$s_i.snippet$,$s_i.url$,因此可以认为用户进一步的搜索需求是凭借 *s.title* 和 *s.snippet* 确定的.同时,受在线处理的响应时间所限,本文仅从 *s.title* 和 $s_i.snippet$ 中抽取关键字.由于 *s.title* 和 $s_i.snippet$ 极短,而且不是完整的内容,所以难以采用更精确的关键字抽取方法.本文采用了一种简单的、基于词频的抽取关键字的算法.

定义 1(局部频繁阈值和局部频繁词). 对于返回的每个文档片段 $s_i\in S(i=1,2,\ldots,m)$,设 $tf(w)$为词 *w* 在文档 s_i 中的频数,对于一个给定的阈值 θ_{local},如果 $tf(w)\geq\delta_{local}$,则 *w* 为 s_i 的一个局部频繁词,θ_{local} 称为局部频繁阈值.

定义 2(全局频繁阈值和全局频繁词). 对于返回结果集合 *S*,设 $df(w)$为文档集合 *S* 中包含词 *w* 的文档数,对于一个给定的阈值 θ_{global},如果 $df(w)\geq\theta_{global}$,则 *w* 为 *S* 的一个全局频繁词,θ_{global} 称为全局频繁阈值.

显然,每处理一个文档 $s\in S$,即可得到 *s* 中的局部频繁词,而只有处理完 *S* 中的每个 *s*,才能得到 *S* 中的全局频繁词,本文的抽取关键字的算法是抽取 *S* 中的全局频繁词和其中每个 $s\in S$ 中的局部频繁词作为关键字.由于局部频繁词和全局频繁词均是从文档片段中获得,所以这些频繁词实际上是与查询词共现的那些词.

目前,抽取文档特征常用的方法是 TFIDF 方法[11],旨在从一个文档集合中抽取在某篇文档中频繁出现,而在其他文档中不频繁出现的词.本文抽取关键字的目的是为信息型意图提供搜索导航,因此,抽取的词应该被足够多的文档所包含,以保证该词被选择后能够返回足够多的结果.基于此,本文将抽取全局频繁词作为关键字;同时,当网上出现一些新的内容时,或许相关的文档并不足够多,如果仅考虑全局频繁词,则这些新文档中的词将不会被抽取,为了加以弥补,本文将抽取那些在一篇文档中出现频数足够多的词,即局部频繁词也作为关键字.因此可以说,本文的方法是在保证足够多(找全局频繁词)的同时又不遗漏足够新(找局部频繁词)的词.TFIDF 方法显然不能获得全局频繁词,同时根据定义 1,得到局部频繁词只需计算 TF 即可.

基于上述思想,网页内容泛化算法 $\mathcal{E}k(s,fw_{local}\text{-}List,fw_{global}\text{-}candidate\text{-}List)$(即 algorithm $\mathcal{GM}odel$ 的第 9 行)描述如 Algorithm $\mathcal{E}k$ 所示.

Algorithm $\mathcal{E}k$.

Input: Document $s\in S$.

Output: Local frequent word list fw_{local}-*List*, candidate global frequent word list fw_{global}-*candidate-List*.

Description:

1. Delete stopwords from *s*;
2. Stem the words in *s* and construct word set *W*;
3. fw_{local}-*List*={};
4. For all $w\in W$
5. Compute $tf(w)$;
6. If $tf(w)\geq\theta_{local}$ Then fw_{local}-*List*=+$\langle w,tf(w)\rangle$ // The structure of fw_{local}-*List* is $\langle word,frequency\rangle$
7. End If
8. fw_{global}-*candidate-List*=+$\langle w,1\rangle$; // fw_{global}-*candidate-List* has the same structure as fw_{local}-*List*
9. End For;

10. Revert all $w \in fw_{local}$-*List* or $w \in fw_{global}$-*candidate-List*; // If *w* was stemmed, revert it

Algorithm $\mathcal{E}k$的输入是 Algorithm $\mathcal{GM}odel$正在处理的文档 *s*,输出是局部频繁词列表和候选全局频繁词列表.该算法首先应用自然语言处理技术去除停用词并取词干,表示成为 Bag-of-Words 形式(第 1、2 行).处理时,涉及两个集合:fw_{local}-*List* 和 fw_{global}-*candidate-List*,它们的结构均为⟨*word*,*frequency*⟩,*word* 为一个词,*frequency* 为 *word* 的频数.fw_{local}-*List* 为局部频繁词集合,*frequency* 为 *tf*(*word*),根据定义 1,该集合是从一个文档中获取的,因此,处理了任意一个 $s \in S$ 中的词 $w \in s$,即可确认该文档中的局部频繁词(第 6 行),当 Algorithm $\mathcal{E}k$算法完成返回 Algorithm $\mathcal{GM}odel$ 后,即可将 fw_{local}-*List* 中的词和对应频数加入 *kw-List*(见 algorithm $\mathcal{GM}odel$ 算法的第 10 行). fw_{global}-*candidate-List* 为候选全局频繁词集合,*frequency* 为 *df*(*word*),根据定义 2,只有文档集合 *S* 中的所有文档均处理完毕,才能确认全局频繁词,因此,在 Algorithm $\mathcal{E}k$ 中只能使 *s* 中出现的词的 *df* 值增 1(第 8 行),返回 Algorithm $\mathcal{GM}odel$后,当所有 $s \in S$ 处理完毕后,将 fw_{global}-*candidate-List* 中频数满足全局频繁阈值的词加入 *kw-List* (见 algorithm $\mathcal{GM}odel$算法的第 14~17 行).

3.2 文档格式泛化算法

文档格式泛化,即从 *S* 中抽取文档格式.很多文档中,在其 URL 的最后一个域(这里将用"/"分割开来的部分称为一个域)包含有格式的标识,如"www.fas.org/irp/crs/RL31798.pdf"表示该文档为 pdf 格式.因此,只要在 *s.url* 中获取最后的域即可.但也有些文档的 URL 没有这样明确地给出文档格式,如"www.spss.com/data_mining/".分析发现,这类文档的功能实际上与 htm 等效,虽然它们并非均为 htm 格式,但就满足事务型意图的需求而言,区分 html,htm,xml 以及其他类似形式的格式并无意义,因此,这种情况下将这类文档均视为 htm 格式.

基于上述思想,文档格式泛化算法 $\mathcal{E}f(s)$(即 Algorithm $\mathcal{GM}odel$的第 4 行)描述如 Algorithm $\mathcal{E}f$ 所示.

Algorithm $\mathcal{E}f$.

Input: Document $s \in S$.

Output: *s.format*.

Description:

1. *format-field*=the last field of *s.url*;
2. If "."∈*format-field* Then
3. *s.format*=Delete "." from *format-field*
4. Else *s.format*=htm;
5. End If;

Algorithm $\mathcal{E}f$ 的输入是 Algorithm $\mathcal{GM}odel$正在处理的文档 *s*,输出是 *s* 的格式.根据前述的格式获取方法,首先获取 *s.url* 的最后一个域(第 1 行),如果该域中含有".",则其后的部分即为文档格式(第 3 行),否则均视为 htm 格式(第 4 行).

3.3 导航类型泛化算法

导航类型泛化,即从每个文档 $s \in S$ 中抽取文档类型,其难点在于分析 *s* 的哪部分内容能够表征以及如何表征文档的类型.根据前面的分析,抽取文档类型的目的是满足导航型意图的信息需求,而该意图的信息需求主要是寻找能够为用户提供进一步的导航操作的站点,这类站点的典型是"主页"或具有主页性质的站点,因此,本文目前将类型 type 划分为"主页导航 home"、"非主页导航 non-home"和"非导航 other".这里的"home"是狭义的主页,即某个人、群体网站的主页,"non-home"则是那些虽非主页、但具有导航性质的网站.

定义 3(主页导航). 设 *s* 为搜索引擎返回的一个文档,如果 *s.content* 是一个主页,则将 *s* 视为主页导航.

定义 4(非主页导航). 设 *s* 为搜索引擎返回的一个文档,如果 *s.content* 包含足够多的与 *s.title* 相关的导航信息,且 *s.content* 不是一个主页,则将 *s* 视为非主页导航.

根据定义 4,如果 *s* 为一篇文章,而其中部分术语或参考文献以超链接形式给出,或者 *s* 为一本书,而其中每节的内容以超链接形式给出,则均不能视为非主页导航,因为它们不具有足够强的导航功能.一个以文本信息为

主的网页,如果其中以超链接形式嵌入很多不相关的广告,则也不能视该页为非主页导航.

主页能够提供搜索导航是不言而喻的,很多非主页也能提供搜索导航,但其导航质量有别于主页.如“东北大学”主页能够提供关于东北大学信息的全面搜索;“东北大学信息学院”页面虽然也能够提供导航,甚至其中包含了进入东北大学主页的链接,但对于搜索东北大学的用户而言,显然“主页导航”更适合.

根据网页设计的形式,如果文档 s 是一个主页,则其典型的形式是 *s.url* 只有 1 个域(这里“域”的定义与第 3.2 节相同),例如“www.jos.org.cn”.而具有导航性质的非主页的 *s.url* 形式则比较复杂.从格式上看,除了 txt 文件以外,其余格式的文件均可加入超链接,所以,理论上它们都可能是非主页导航.但就网站设计的习惯而言,pdf,doc,ps,txt,xls,ppt,rtf 等格式很少用于设计网站,特别是从导航的意义上看,这类文档难以具备定义 4 的导航功能.

基于上述思想,网页类型泛化算法 $\mathcal{Et}(s)$(即 algorithm $\mathcal{GModel}$ 的第 5 行)描述如 Algorithm $\mathcal{Et}$ 所示.

Algorithm $\mathcal{Et}$.

Input: Document $s \in S$.

Output: *s.type*.

Description:

1. If *s.url* doesn't contain "/" or only contains one "/" Then *s.type*="home"
2. Else *format-field*=the last field of *s.url*;
3. If "."∈*format-field* and *format-field*-"."∈{pdf,doc,ps,txt,xls,ppt,rtf} Then
4. *s.type*="other"
5. Else *s.type*="non-home"
6. End If
7. End If

Algorithm $\mathcal{Et}$ 的输入是 Algorithm $\mathcal{GModel}$ 正在处理的文档 *s*,输出是 *s* 的类型(主页导航、非主页导航、其他).根据前述主页的 URL 形式,如果在 *s.url* 不出现"/"或最后出现一个"/",则 *s.url* 只有 1 个域,满足主页形式,*s* 为主页(第 1 行);否则,如果该 *s* 为 pdf,doc,ps,txt,xls,ppt,rtf 等格式的页面,则 *s* 被视为非导航(第 2~4 行).其余情况下,*s* 被视为非主页导航(第 5 行).

4 实验及评价

基于 Web 网页动态泛化的思想,本文实现了一个原型系统.根据 3 种搜索意图关系的分析,信息型意图是源头,因此,原型系统的初始界面像通用搜索引擎那样要求用户提交查询关键字.当初始查询提交后,原型系统返回查询结果和类型列表 *type-List*、格式列表 *format-List*、新关键字列表 *kw-List*.查询结果与通用搜索引擎功能相同,用户在此点击感兴趣的文档,而 3 个列表为用户提供了进一步的搜索导航信息.

本文基于用户搜索意图的 Web 网页动态泛化思想,相关算法及原型系统的实现是在目前通用搜索引擎基础上进行的,相关文档的获取和搜索结果的排序均借助于搜索引擎的功能.但由于本文工作中分析了用户搜索意图的信息需求及其关系,因此提供了与用户搜索意图更加相关的结果以及进一步的搜索导航.

4.1 实验设计

本文的实验基于通用搜索引擎 Google,查询提交及获取结果均来自于 Google API.采用文献[12]给出的查询词(见表 1)进行查询.

Table 1 Query words (keywords)

表 1 查询词(关键字)

Affirmative action	Alcoholism	Architecture	Amusement parks	Bicycling	Blues
Citrus groves	Cheese	Classical guitar	Computer vision	Cruises	Death valley
Field hockey	Gardening	Graphic design	Gulf war	Hiv	Java
Lyme disease	Lipari	Mutual funds	Parallel architecture	National parks	Recycling cans
Rock climbing	Shakespeare	San francisco	Stamp collecting	Sushi	Table tennis
Telecommuting	Vintage cars	Volcano	Zen buddhism	Zener	

由于基于用户搜索意图的 Web 网页动态泛化的工作尚未见诸报道,因此,实验将通用搜索引擎的结果与本文开发的原型系统的结果即是否采用 Algorithm *GModel* 的结果进行对比,评价采用返回的前 10 个结果的准确率 *P*@10,而与其他相关工作的对比则采用定性分析方法.除了第 4.2.2 节的实验以外,其余实验均设置δ_{local}=3, θ_{global}=30,并从 Google API 获取前 500 个文档片断,而第 4.2.2 节实验的文档片断数可变.

这里以关键字 data mining 为例,进一步说明实验的分析方法.表 2 是 data mining 从本文模型返回的前 10 个结果及其格式和类型分析以及从所有返回结果中抽取的新关键词.

Table 2 Search results and their analysis for "data mining"

表 2 "data mining"的搜索结果及分析

No.	Url	Type		Format		New keyword
		Result	TF	Result	TF	
1	en.wikipedia.org/wiki/Data_mining	Non-Home	T	htm	T	Software
2	www.anderson.ucla.edu/faculty/jason.frand/ teacher/technologies/palace/datamining.htm	Non-Home	F	htm	T	Text
3	www.thearling.com/text/dmwhite/dmwhite.htm	Non-Home	F	htm	T	Oracle
4	datamining.typepad.com/	Home	F	com	T	Knowledge
5	www.statsoft.com/textbook/stdatmin.html	Non-Home	T	html	T	Web
6	www.autonlab.org/tutorials/	Non-Home	T	htm	T	Discovery
7	www.eco.utexas.edu/~norman/BUS.FOR/course.mat/Alex/	Non-Home	T	htm	T	Statistics
8	www.the-data-mine.com/	Home	T	com	T	Applications
9	www.kdnuggets.com/	Home	T	com	T	Visualization
10	www.oracle.com/technology/products/bi/odm/index.html	Non-Home	T	html	T	Social Media

表 2 中,类型和格式的结果是根据第 3.2 节、第 3.3 节中的算法判定的;而对其评价(T 或 F)则是通过点击看其实际网页的类型和格式是否与判定结果相符来判定的,相符为 T,否则为 F.例如,根据定义 4,表 2 中第 2、3 项的类型均为"非主页导航",第 4 项的类型为"主页导航",但第 2 项实际上只是一篇文章,不具有导航功能,而第 3、4 项对应的网页不存在,对此也视为 F.在后面关于类型和格式查询结果的准确率计算中,T 为准确,F 为不准确.

4.2 实验结果及分析

4.2.1 导航类型泛化实验

应用表 1 给出的查询词在本文开发的原型系统中进行查询,在返回的 *Type-List* 中分别选择"主页导航"和"非主页导航",返回的前 10 个结果的准确率 *P*@10 如图 3、图 4 所示.其中,应用本文原型系统获得的结果称为 With model,从搜索引擎直接返回的结果标识为 Without model.

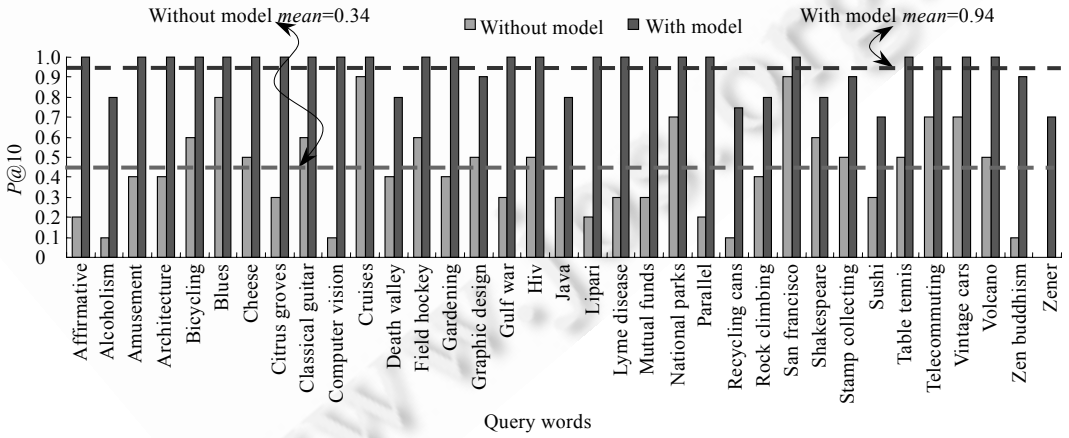


Fig.3 Experimental results for navigation type (Home navigation)

图 3 导航类型实验结果(主页导航)

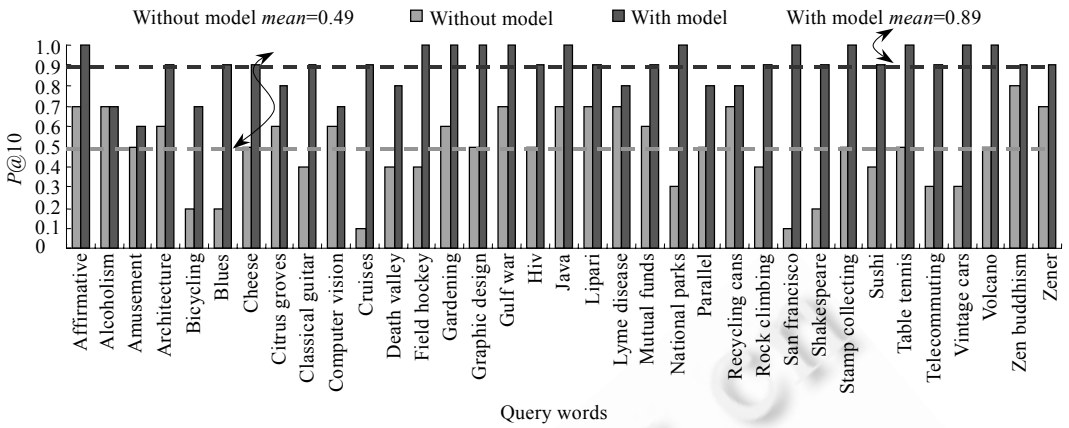


Fig.4 Experimental results for navigation type (Non-Home navigation)

图 4 导航类型实验结果(非主页导航)

实际上,在通用搜索引擎的返回结果中,排在 Top-10 的文档大多具有导航功能,有些查询(如 Cruises,San Francisco 等)返回结果中 Top-10 的主页导航较多,而有些查询(如 Affirmative action,Alcoholism,Gulf war,Zen buddhism 等)返回结果中 Top-10 的非主页导航较多.就第 1 节分析的导航型搜索意图的信息需求而言,区分主页导航、非主页导航和非导航文档是有意义的,而本文的网页动态泛化方法能够通过导航获取用户的导航型搜索意图、为该意图的用户提供这样的选择,并根据选择返回更加精确的结果.

另外,实验中并非所有的查询词返回结果的 *Type-List* 均能包含 3 种类型,一些查询的返回结果中没有 other 类型,则返回的 *Type-List* 将不显示它,从而使用户不会盲目地选择.

4.2.2 文档格式泛化实验

像 Google 这样的搜索引擎提供了格式选择的功能,但本文所提供的格式选择是根据返回结果动态变化的,对于一些查询,返回结果中可能不包含某些格式,则返回的 *Format-List* 将不显示它们.

应用表 1 给出的查询词分别在本文开发的原型系统中进行查询,在它们返回的结果中,绝大多数均有且仅有 html 和 htm 类型,其中 Affirmative action 还包含 pdf,Travel 还包含了 asp,Field hockey 还包含了 xml,Recycling cans 还包含了 cfm 和 pdf,Table tennis 还包含了 php,Telecommuting 和 Volcano 还包含了 shtml.即从 Google API 获取的前 500 个文档片段中,主要以 html 和 htm 以及其他没有明确标识(亦按照 htm 处理)的文档片段为主,其原因是这些查询词均为较短且较常用的查询词,而非专业术语,因此可供下载阅读的格式如 pdf,ps 等较少且未被搜索引擎排在前面.对于一些专业术语,如查询 association rule mining 可以得到 html,htm,pdf,ps 等格式,而查询 Identifying User Goals from Web Search Results 则可以得到 html,htm,jsp,cfm,pdf,mspx,ppt 等格式.

另外,这是对搜索引擎前 500 个文档片断的分析结果,当片断数目增加或减少时,该结果会发生变化.由于该参数可变,用户可以根据结果和需求对其设置.为此,在实现算法和实验中将从 Google API 获取的片断数目设置为变量,对其加以改变,文档格式的数量将随之变化.总之,当搜索词内容从通用向专业变化、搜索词数量由少向多变化以及返回片断数量设置由少向多变化时,均会引起文档格式由少向多的变化.因此,*Format-List* 避免了用户对结果格式选择的盲目性.

4.2.3 网页内容泛化实验

应用表 1 给出的查询词(query word)在本文开发的原型系统中进行查询,返回的 *kw-List* 中包含了从这些查询词对应返回结果中抽取的新关键字(new keyword).根据返回的 *kw-List*,继续选择 New Keyword,将得到以 Query Word+New Keyword 为关键字的搜索结果.这里选择一些 Query Word 和 New Keyword,在 Google 搜索引擎根据 Query Word 得到的搜索结果中寻找第 1 个与 New Keyword 相关的结果所在的位置.表 3 展示了这个结果.

Table 3 Location of first result related with new keyword in Google search results

表 3 Google 搜索结果中第 1 个与新关键词相关结果的位置

Query word	New keyword	Location	Query word	New keyword	Location	Query word	New keyword	Location
Affirmative action	Middot	>100	Architecture	Green	14	Architecture	Amp	>100
Bicycling	Tours	18	Bicycling	Aboutcom	>100	Blues	Hobcom	>100
Cheese	Specialty	12	Cheese	Order	11	Classical guitar	Liona	26
Computer vision	Brown39s	>100	Death valley	Californianevada	>100	Field hockey	Home	30
Field hockey	Stx	12	Gardening	Tools	11	Gardening	Design	15
Gulf war	Stories	12	Gulf war	Warquot	>100	Hiv	Causes	15
Lipari	Joanna	18	Lipari	Battle	22	National parks	Mountains	17
Parallel architecture	Nmsu	12	Recycling cans	Receptacles	19	Recycling cans	Newspapers	23
Rock climbing	Magazines	>100	San francisco	Advanced	12	San francisco	Area39s	>100
Shakespeare	Craig	11	Table tennis	Equipment	18	Table tennis	Dvds	17
Vintage cars	European	14	Vintage cars	Customs	21	Volcano	Mountains	11
Zen buddhism	Texts	12	Zen buddhism	Japanese	13	Zener	Engine	31

表 3 中的 Location 均大于 10,这表明,如果不应用模型输出的 *kw-List*,用户至少要翻 1 页以上才能找到与新关键字相关的结果(这里的新关键字是从 Google API 的前 500 个文档片断中抽取的,改变片断数目和频数阈值均会影响 *kw-List* 的结果),而在模型输出的 *kw-List* 中选择新关键字,则可直接获得相关的结果.

4.2.4 运行时间实验

由于原型系统是对搜索引擎获得的结果进行进一步的分析,所以查询响应时间比直接从搜索引擎获得结果的响应时间更长一些.从搜索引擎获取的结果越多,分析时间越长.这里分别取前 50,100,150,200,250,300,350,400 个文档,计算表 1 中每个查询词的分析时间再取平均值(在 IBM ThinkPad,987MHz,0.99GB 内存,Windows XP 2002 环境中运行,采用 Java 编程).这里的分析时间为“给出结果所用时间”减去“获取信息所用时间”,结果分别为 15.657 14ms,31.342 86ms,46.485 71ms,58.428 57ms,74.685 71ms,88.885 71ms,101.342 9ms,117.057 1ms,如图 5 所示.

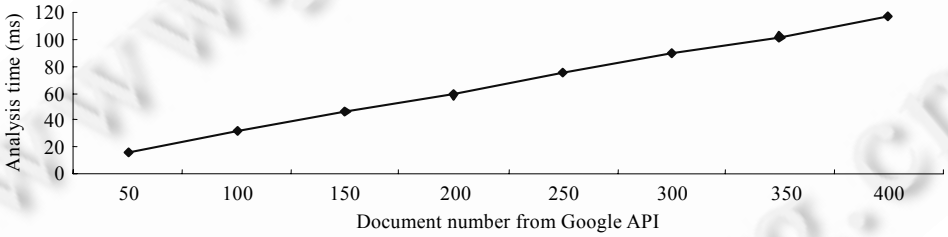


Fig.5 Time of Web page analysis

图 5 分析网页的时间

图 5 表明,分析所用时间与文档数目近似成线性关系.从 Algorithm *GModel* 算法可知,分析文档的时间复杂度为 $O(|S|)(|S|$为返回结果集 S 中的文档片断数目).与所得到的导航信息和更加相关的查询结果相比,这样的响应时间是可以接受且值得的.

4.3 泛化模型建模方法的讨论

本文提出的基于用户搜索意图的 Web 网页泛化模型涉及网页内容、文档格式和导航类型的泛化,即从文档片段中抽取关键字,根据网页特征判断其格式和内容.

对于导航类型泛化,根据导航型意图的需求,本文定义了主页导航、非主页导航和非导航 3 种类型,并给出了判断方法.第 4.2.1 节中的实验结果表明,本文所提出的方法,特别是非主页导航类型的判断方法还需要进一步细化.对于文档格式的泛化,则是根据 URL 判断网页的实际格式,因此其准确性主要依赖于 URL 给定格式本身的准确性.此外,对于事务型意图所需的一些在线操作,如购物、玩游戏等,目前模型提供的格式尚不能给出更准确的信息,为此将进一步考虑从文档片段中获取信息.

对于网页内容的泛化,本文采用简单的基于词频的方法,其中涉及 3 个阈值的选取,即从搜索引擎获取的网

页数目、局部频繁阈值和全局频繁阈值.从搜索引擎获取的网页数目越多,得到的关键词就越多,也就越能更全面地反映网页的内容,但分析时间也越长.图 5 所展示的平均时间及分析文档的时间复杂度 $O(|S|)$为其提供了参考依据.而局部频繁阈值和全局频繁阈值的选取则面临当前频繁模式挖掘中所遇到的共同问题,即如果阈值大,则获得的关键字太少,可能会遗漏一些重要的关键字;如果阈值小,则获得的关键字太多,其中会包含很多不重要的关键字.为此,根据 Web 文档的情况(仅仅是一些包含查询词的短句),本文推荐的局部频繁阈值为 3~4,而全局频繁阈值则为获取的网页数目的 6%~10%.特别是上述阈值均可根据用户需求进行调整,用户可在响应时间、获取关键字数目诸方面求得平衡.此外,目前网页内容的泛化结果还仅仅是单个词,这在一些情况下也不能准确地反映网页的内容,为此,将进一步考虑从文档片段中抽取频繁词组或短语作为关键字.

4.4 用户搜索意图获取方法讨论

本文的 Web 网页泛化模型是基于用户搜索意图建立的,但在实现时并非直接获取用户意图,而是在用户提交的初始搜索关键字对应的返回结果的基础上,通过网页内容、文档格式和导航类型的泛化,为不同的搜索意图提供进一步的搜索导航,进而通过用户进一步的选择,使其获得更相关的搜索结果.图 2 给出了每种搜索意图及其实现,其后的处理过程均基于此.

当前,关于用户兴趣和搜索引擎的研究提出了许多方法.这些方法可分为两类:一类是自动地收集用户信息,根据这些信息隐式地分类用户意图,这里称为隐式方法(implicit approach);另一类方法是显式地给出搜索类别而让用户选择,这里称为显式方法(explicit approach).表 4 给出了从用户意图获取、是否与用户交互、是否建立用户模型、Web 网页模型建立基础、初始模型构建、数据集规模和模型更新几个方面对显式方法、隐式方法及本文提出方法的比较结果.

Table 4 Comparison of methods for search intention

表 4 获取搜索意图方法的比较

Aspects	Approach		
	Implicit approach	Explicit approach	Approach of this paper
Acquisition of intention	Implicit	Explicit	Implicit
Interaction with users	No	Yes	Yes
Creating user model	Yes	No	No
Web page modeling base	Access history	Access history	Results returned
Initial model assigning	Given by experts	Brain-storm	Extracted from results returned
Data set size	Large	Large	Large or small
Model result updating	By user model	Interacting with user	By results returned

从表 4 可知,本文提出的方法具有无须建立用户模型、无需大规模数据集、与用户直接交互、根据返回结果自动更新网页模型等优点.特别是“与用户直接交互”的过程是通过给出导航信息而隐式获取用户搜索意图并提供满足该意图结果的过程,所提供的导航信息和结果随文档内容动态变化.

当然,仅仅考虑网页内容、文档格式和导航类型的泛化以及目前的泛化方法,对于满足信息型、导航型和事务型搜索意图也存在一定的不足.为此,将进一步细化上述搜索意图,并分析其信息需求,以提供更加相关的搜索结果.

5 相关工作及讨论

在用户搜索意图或搜索目的的研究方面,文献[1]将搜索意图分类为导航型、信息型和事务型 3 类;文献[2]将上述 3 类用户意图细分为更多的子意图,并开发了一个意图分类工具,以这些类别为类标签,对新的搜索意图进行分类;文献[3]将上述 3 类用户意图进行了横向划分,并以此为类标签,采用 SVM 方法分类新的搜索意图;文献[4]以上述 3 类意图为类标签,实现了一种更加准确的意图分类算法;文献[6]认为用户在查询返回结果上的点击尚不能全面反映用户意图,因而提出通过分析用户在结果上的鼠标移动推断用户意图,其分类采用导航型查询和信息型查询,并对模糊的查询予以消歧处理;文献[7]将查询意图设置为 product intent 和 job intent,以查询词和点击的 URL 为数据源建立点击图,应用半监督学习方法分类查询意图;文献[8]将查询意图分为显式和隐式查

询,应用Web Log数据源并引入句法分析技术进行分类;文献[5]开发了一个自动识别用户意图的框架,应用有监督和无监督学习方法发现用户搜索目的;文献[13]建立了一个概率推演模型,采用依据句法的特征从Web搜索结果中发现用户查询目的;文献[14]针对搜索目的的识别问题,提出了用户点击行为分布和链接分布作为潜在特征进行搜索目的的预测.这些方法主要集中在如何精确地分类和预测用户搜索意图或搜索目的方面,但却并未考虑如何针对搜索意图提供进一步的帮助.文献[8]虽然对不同意图的查询提供了返回结果,但这种分类(即显式和隐式)实际上是针对前述3类查询意图中信息型查询问题而言的,而且应用Web Log数据也是一种静态分析方法.文献[15]提出了泛化和精炼的问题,但工作重点落在查询的精炼而非Web文档的泛化上,通过建立查询目的图来理解用户查询目的,因此其导航只是根据查询词而未考虑文档的动态变化.本文的工作未直接分类和预测用户意图,却是基于文献[1–4,6]关于搜索意图的分类来研究搜索结果的泛化问题,通过Web网页内容、文档格式和导航类型的泛化为各种搜索意图提供相关的搜索导航.这里对搜索意图的分类是隐式的.模型通过概念提升为用户提供搜索导航,通过概念下降精炼用户搜索,使搜索结果更接近于用户搜索意图.

在用户搜索兴趣的分析方面,文献[16]提出将Web作为一个图的泛化查询结果的方法,从URL和领域方面对查询结果进行泛化,并建立查询与泛化结果之间的映射;文献[17]提出了一种基于用户搜索行为建立用户模型的方法;文献[18]提出了“提交相同关键词的不同用户可能需要不同的查询结果”的问题,进而针对不同的用户提供不同的结果;文献[19]提出了自动识别用户兴趣的方法,通过建立用户模型为用户定制搜索结果;文献[20]提出辅助导航检索方法,将查询视为导航的开始,基于用户的查询历史进行查询导航.其他关于用户兴趣的研究这里不再一一列举.这些研究的主要目的是获取不同的用户信息,并根据用户的访问行为和访问历史探索其不同的兴趣,以便当这些用户搜索时,即便提交相同的查询词,亦可根据其兴趣提供不同的返回结果.这些方法为改进搜索引擎的质量提供了基础,但这些改进大多基于用户兴趣模型,用户模型的建立和维护也是一个正在研究的问题.本文的方法通过与用户交互获得用户搜索意图,而未建立用户模型,因此不同于这些工作.

关于搜索引擎的研究还包括搜索结果的重排[12]、自适应搜索引擎的功能和交互能力[18]、元搜索的设计[21]、搜索结果的聚类[22,23]等.特别是文献[22]提出了提供不同层次关键字的方法,该方法中关键字的层次结构类似于本文的网页内容泛化结果,但该方法仅考虑了关键字的抽取,而不同于本文的内容、格式、类型的共同泛化.

在文本特征抽取的研究中,基于TFIDF权重[11]、与查询词共现[24]、频度与分布[25]、与查询词语义相似性[26]、位置权重词[27]等的特征抽取是一些较为典型的文本特征抽取技术.根据搜索引擎返回的标题和文档片段的特点(短、不完整),本文采用频繁词抽取方法.由于返回的文档片段是对应网页内容中包含查询词的一些句子,所以,抽取的频繁词实际上是与查询词共现频度较高的那些词,以此对Web网页内容进行泛化并为信息型搜索意图提供搜索导航信息,符合查询词选择时从粗糙到精炼的过程.此外,在抽取频繁词时,本文考虑了局部频繁和全局频繁两种情况,以保证获取出现足够多的全局频繁词的同时又不遗漏足够新的局部频繁词.

6 结 语

获取用户意图以及对用户意图分类是当前的热点研究问题.本文基于当前一种比较流行的搜索意图分类方法(导航型、信息型、事务型),进一步分析了各种意图的信息需求及意图之间的关系,从满足用户搜索意图的观点出发,提出了基于用户搜索意图的Web网页动态泛化模型,实现了对搜索结果的网页内容、文档格式和导航类型泛化.由于模型的输出随返回内容动态变化,因此,这种动态泛化结果能够为不同的搜索意图提供进一步的搜索导航,进而使其获得更相关的结果.

为了更好地满足用户的搜索意图,进一步的研究工作包括:(1) 探索新的用户搜索意图,并基于此改进本文的泛化模型;(2) 进一步改进文档特征的抽取方法,由仅仅抽取单个频繁词扩展到短语或词组,并在抽取过程中引入领域知识;(3) 进一步细化网页格式和类型的泛化结果;(4) 进一步研究搜索内容的选择以及返回结果的重排;(5) 探索和引入更多、更科学的搜索结果评价方法[28],对泛化模型及搜索结果进行更全面、客观的评价,并根据评价结果对模型进行改进.

References:

[1] Broder AZ. A taxonomy of Web search. SIGIR Forum, 2002,36(2):3−10.

[2] Rose DE, Levinson D. Understanding user goals in Web search. In: Feldman SI, Uretsky M, Najork M, Wills CE, eds. Proc. of the 13th Int'l Conf. on World Wide Web. New York: ACM Press, 2004. 13−19.

[3] Dai HH, Zhao LZ, Nie ZQ, Wen JR, Wang L, Li Y. Detecting online commercial intention (OCI). In: Les C, David DR, Arun I, Carole AG, Michael D, eds. Proc. of the 15th Int'l conf. on World Wide Web. New York: ACM Press, 2006. 829−837.

[4] Jansen BJ, Booth DL, Spink A. Determining the user intent of Web search engine queries. In: Williamson CL, Zurko ME, Patel-Schneider PF, Shenoy PJ, eds. Proc. of the 16th Int'l Conf. on World Wide Web. New York: ACM Press, 2007. 1149−1150.

[5] Ricardo A, Liliana CB, Cristina N. The intention behind Web queries. In: Crestani F, Ferragina P, Sanderson M, eds. Proc. of the 13th Int'l Conf. on String Processing and Information Retrieval (SPIRE 2006). Berlin, Heidelberg: Springer-Verlag, 2006. 98−109.

[6] Qi G, Eugene A. Exploring mouse movements for inferring query intent. In: Myaeng SH, Oard DW, Sebastiani F, Tat-Seng C, Leong MK, eds. Proc. of the 31st Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. 2008. 707−708.

[7] Li X, Wang YY, Alex A. Learning query intent from regularized click graphs. In: Myaeng SH, Oard DW, Sebastiani F, Tat-Seng C, Leong MK, eds. Proc. of the 31st Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. 2008. 339−346.

[8] Strohmaier M, Prettenhofer P, Lux M. Different degrees of explicitness in intentional artifacts: An exploratory study of user goals in a search query log. In: Gordon AS, Havasi C, Lux M, Strohmaier M, eds. Proc. of the Workshop on Common Sense Knowledge and Goal-Oriented Interfaces. Aachen: CEUR-WS.org, 2008.

[9] Han JW, Kamber M. Data Mining: Concepts and Techniques. 2nd ed., Beijing: Chinese Machine Press, 2006. 157−225.

[10] Ruvini JD. Adapting to the user's Internet search strategy. In: Brusilovsky P, Corbett AT, Rosis F, eds. Proc. of the 9th Int'l Conf. on User Modeling (UM 2003). Berlin, Heidelberg: Springer-Verlag, 2003. 55−64.

[11] Kantrowitz M, Mohit B, Mittal V. Stemming and its effects on TFIDF ranking. In: Nicholas JB, Peter I, Mun-Kew L, eds. Proc. of the 23rd Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2000. 357−359.

[12] Haveliwala TH. Topic-Sensitive PageRank: A context-sensitive ranking algorithm for Web search. IEEE Trans. on Knowledge and Data Engineering, 2003,15(4):784−796.

[13] Chang YS, He KY, Yu S, Lu WH. Identifying user goals from Web search results. In: Proc. of the IEEE/WIC/ACM Int'l Conf. on Web Intelligence. Los Alamitos: IEEE Computer Society, 2006. 1038−1041.

[14] Lee U, Liu ZY, Cho JH. Automatic identification of user goals in Web search. In: Ellis A, Hagino T, eds. Proc. of the 14th Int'l Conf. on World Wide Web. New York: ACM Press, 2005. 391−400.

[15] Strohmaier M, Lux M, Granitzer M, Scheir P, Liaskos S, Yu ES. How do users express goals on the Web?—An exploration of intentional structures in Web search. In: Weske M, Hacid MS, Godart C, eds. Proc. of the Web Information Systems Engineering—WISE 2007 Workshops. Berlin, Heidelberg: Springer-Verlag, 2007. 67−78.

[16] Leskovec J, Dumais ST, Horvitz E. Web projections: Learning from contextual subgraphs of the Web. In: Williamson CL, Zurko ME, Patel-Schneider PF, Shenoy PJ, eds. Proc. of the 16th Int'l Conf. on World Wide Web. New York: ACM Press, 2007. 471−480.

[17] Sendhilkumar S, Geetha TV. User's search behavior graph for aiding personalized Web search. In: Ghosh A, De RK, Pal SK, eds. Proc. of the the 2nd Int'l Conf. on Pattern Recognition and Machine Intelligence (PReMI 2007). Berlin, Heidelberg: Springer-Verlag, 2007. 357−364.

[18] Sugiyama K, Hatano K, Yoshikawa M. Adaptive Web search based on user profile constructed without any effort from users. In: Feldman SI, Uretsky M, Najork M, Wills CE, eds. Proc. of the 13th Int'l Conf. on World Wide Web. New York: ACM Press, 2004. 675−684.

[19] Qiu F, Cho J. Automatic identification of user interest for personalized search. In: Carr L, Roure DD, Iyengar A, Goble CA, Dahlin M, eds. Proc. of the 15th Int'l Conf. on World Wide Web. New York: ACM Press, 2006. 727−736.

[20] Pandit S, Olston C. Navigation-Aided retrieval. In: Williamson CL, Zurko ME, Patel-Schneider PF, Shenoy PJ, eds. Proc. of the 16th Int'l Conf. on World Wide Web. New York: ACM Press, 2007. 391−400.

[21] Han EH, Karypis G, Mewhort D, Hatchard K. Intelligent metasearch engine for knowledge management. In: Proc. of the 2003 ACM CIKM Int'l Conf. on Information and Knowledge Management. New York: ACM Press, 2003. 492−495.

[22] Ferragina P, Gulli A. A personalized search engine based on Web-snippet hierarchical clustering. Software—Practice and Expert, 2008,38(2):189−225.

[23] Zeng HJ, He QC, Chen Z, Ma WY, Ma JW. Learning to cluster Web search results. In: Sanderson M, Järvelin K, Allan J, Bruza P, eds. Proc. of the 27th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2004. 210−217.

[24] Matsuo Y, Ishizuka M. Keyword extraction from a single document using word co-occurrences statistical information. Int'l Journal on Artificial Intelligence Tools, 2004,13(1):157−169.

[25] Li JW, Baik DK. A model for extracting keywords of document using term frequency and distribution. In: Gelbukh AF, ed. Proc. of the 5th Int'l Conf. on Computational Linguistics and Intelligent Text Processing (CICLing 2004). Berlin, Heidelberg: Springer-Verlag, 2004. 437−440.

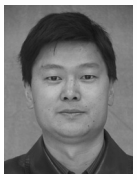
[26] Abhishek V, Hosanagar K. Keyword generation for search engine advertising using semantic similarity between terms. In: Maria LG, Robert JK, Donna S, Chrysanthos D, Frank D, eds. Proc. of the 9th Int'l Conf. on Electronic Commerce. New York: ACM Press, 2007. 89−94.

[27] Hu XH, Wu B. Automatic keyword extraction using linguistic features. In: Workshops Proc. of the 6th IEEE Int'l Conf. on Data Mining (ICDM 2006). Los Alamitos: IEEE Computer Society, 2006. 19−23.

[28] Liu YQ, Fu YP, Zhang M, Ma SP, Ru LY. Automatic search engine performance evaluation with click-through data analysis. In: Williamson CL, Zurko ME, Patel-Schneider PF, Shenoy PJ, eds. Proc. of the 16th Int'l Conf. on World Wide Web. New York: ACM Press, 2007. 1133−1134.

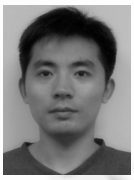
王大玲(1962－),女,辽宁沈阳人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为 Web 挖掘与信息检索.

于戈(1962－),男,博士,教授,博士生导师,主要研究领域为数据库理论与技术.



鲍玉斌(1968－),男,博士,副教授,CCF 高级会员,主要研究领域为数据仓库,数据挖掘.

张沫(1983－),女,硕士生,主要研究领域为 Web 挖掘,信息检索.



沈洲(1982－),男,硕士生,主要研究领域为 Web 挖掘,信息检索.