Selective Recrawling for Object-Level Vertical Search

Yaqian Zhou, Mengjing Jiang, Qi Zhang, Xuanjing Huang, Lide Wu
School of Computer Science, Fudan University
220 Handan Road, Shanghai, P.R.China
{zhouyaqian,082024071,qi_zhang,xjhuang,ldwu}@fudan.edu.cn

ABSTRACT

In this paper we propose a novel recrawling method based on navigation patterns called Selective Recrawling. The goal of selective recrawling is to automatically select page collections that have large coverage and little redundancy to a pre-defined vertical domain. It only requires several seed objects and can select a set of URL patterns to cover most objects. The selected set can be used to recrawl the web pages for quite a period of time and renewed periodically. Experiments on local event data show that our method can greatly reduce the downloading of web pages while keep the comparative object coverage.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval - search process

General Terms: Algorithms, Experimentation.

Keywords: recrawling, vertical search

1. INTRODUCTION

In recent years, object-level vertical search (OLVS, e.g. "vertical search" for products, locations, persons, specific services, or scholarly relationships) is becoming one of the main goals within the web search community. Generally speaking, the OLVS engines discover the target web resources, extract structural data objects from them and then provide more precise query upon these data objects [4]. In the above procedure, data extraction and web database schema matching have been hot spots of researches. However, the task of crawling relevant web pages has been seldom studied.

The goal of OLVS' crawler is to find *object pages*(pages contain objects). As we know, a number of web resources (such as E-Commerce portals, object home pages, interest group sites and et., al) are focusing on specific content categories and provide information for objects related to specific vertical domains. However, information on these sites tends to be fragmented and non-comprehensive. Moreover, there are great redundancy among them since many sites have similar content.

To get such information, a straightforward way is to filter out the target web pages. However, the filtering method requires a prior acquisition of a large amount of web pages, which is a consuming task [3]. [4] used a focused crawler [1] that uses the heuristic page classifier and the existing partial object relationship graph to guide the crawling process. However, as discussed by [3], the link-based search strategy will result in low efficiency since target resources of OLVS often do not link to each other, such as E-Commerce portals. [3] presented a meta-search based method which uses general search engines to find the relevant web sites by submitting queries composed of representative data instances. All of the methods do not take the redundancy into account.

In this paper, we propose a recrawling method based on navigation patterns[5] for the resource collecting task of OLVS. It can selective fetch relevant and less redundant web pages, so called selective recrawling.

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

2. ARCHITECTURE OF THE SYSTEM

In this section we propose a framework for the selective recrawling as illustrated in figure 1. It has two phases: pattern learning phase (left to the dotted line) and crawling phase(right to the dotted line). The crawling phase uses the navigation pattern(NP)-based method which has been described in[2]. Therefore, we will focus on the learning phase, specially the boxes with bold characters.

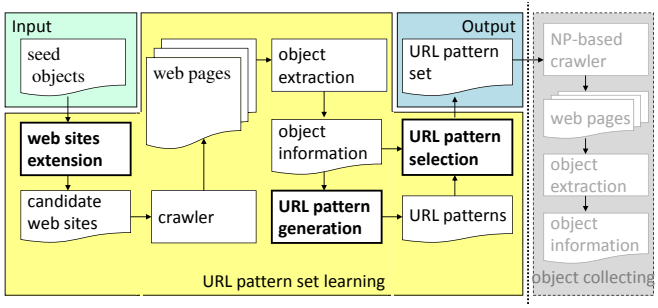

Figure 1: The Framework of the Selective Recrawling System.

As we can see from figure 1, in learning phase, the system first find the relevant web sites based on seed objects. Then it crawls all the pages from each site and concludes the URL patterns for object pages. Finally, it selects a URL pattern set that can cover most objects with least redundancy.

In crawling phase, NPs are generated based on the selected URL pattern set. They can be used to guide crawling later for quite a period of time.

2.1 Web Sites Expansion

The system takes seed objects as input, uses general purpose search engines (e.g. google, yahoo!) to find relevant web sites for the target objects. In order to make the seed less sensitive, we use a two-phase web sites expansion method. It includes two similar steps that first use seed objects as queries to retrieve seed sites and then extract more objects to retrieve more web sites.

For each query, we retrieve the top ranked URLs, and extract their corresponding web sites. Count all the sites, sort them by their retrieved times, and then choose the top sites(e.g. 5) as the seed sites.

Then we download all pages from seed sites and extract all objects from them to extend the seed objects and then repeat the same process to find more sites. At the end, the top N(e.g. 300) sites are used as the candidate sites.

2.2 URL Pattern Generation

The pages that can be extracted at least one object are regarded as object pages. If we take a look at the object pages, we may find out that most of these pages' URLs are almost the same except their parameters, since they are obviously generated from database.

Simplified from [5]'s page structure based method, we cluster the URLs to patterns by only taking the URLs themselves into account. We simply take their common prefixes as patterns since different parameters are usually at the end of URLs. Each URL not matching any patterns is regarded as a single pattern.

2.3 URL Pattern Selection

URL pattern generation can reduce the pages' traveling within a site, furthermore, URL pattern selection can reduce the pages' traveling in all the candidate sites. All the extracted objects will be clustered to some distinct objects. If a distinct object can be extracted from the web pages whose URL can match a pattern, we call the pattern covers the object. After object extraction and URL pattern generation, we have several URL patterns and each pattern covers several objects.

For OLVS, many pages contain same objects. An incremental greedy method is used to select a subset of URL patterns to cover most distinct objects and leave least redundant pages. The method first selects a pattern that can cover maximum distinct objects; and then it selects a pattern that can cover maximum new distinct objects; and such process is repeated until it can't find any pattern that can cover at least k(e.g. 1, 2, or 3) new distinct objects.

3. EXPERIMENTS

Our experiments are based on an event extraction system. An object is described as an event tuple of three elements, (name,location,time). Most of the events are concerts, dramas, sports games and et. al. For example, the system will extract the title of the concert as name, where it is held as location, and when it takes place as time. We use an object extraction tool to extract event information from the structure, semi-structure and free text web pages. In these experiments, the number of seed events is 5, the number of seed sites is 5 and candidate sites is not more than 300.

The first row lists the results of the total downloaded pages; the second row lists the results of all the generated URL patterns; the third, forth and fifth rows list the results for the selected URL pattern sets. If a distinct object is accepted by a human, it is annotated as correct. The different numbers in the bracket of SR stand for different threshold k in URL pattern selection. This threshold controls the tradeoff between object coverage and web page downloading.

Table 1: Compare the number of URL patterns(the 2nd col) that are used to guide the crawling, the size of the web pages(the 3rd col) that need to download, the number of distinct events(the 4th col) and correct events(the 5th col)that be extracted from web pages.

method	pattern	size	distinct	correct
full-download		5.79Gb	13551	8141
recrawling	1755	949mb	13551	8141
SR(1)	1139	918mb	13551	8141
SR(2)	487	723mb	11484	7672
SR(3)	423	709mb	10707	7391

As we can see from table 1, NP based recrawling method can largely reduce web downloading as compared to full downloading, and selective recrawling can further reduce downloading for 3.3-25.3% as compare to non-selective recrawling. In fact, if the performance of object extraction and clustering tool is improved the percentage can even be increased further. The number of extracted objects decreases along with the threshold increases, while the precision of the extracted objects increases along with it increases. It is due to the selected patterns can guide the crawler to fetch easy pages for object extraction.

In order to test how the old URL pattern set works on new crawling, we use the old NP which learned a week ago and the new learned NP to crawl the web pages at the same time. And find that the numbers of distinct events that are extracted from corresponding pages are comparative.

4. ACKNOWLEDGMENTS

The authors would like to thank Li Dai of IBM for her help in reviewing this work and Dr. Xiangyang Xue for his valued discussion. This work was partially supported by Chinese National Science Foundation(60503070), the foundation of Huawei, 973 Program(Project No. 2010CB327900) and the Shanghai Committee of Science and Technology, China (Grant No. 08511500302).

5. REFERENCES

[1] S. Chakrabarti *et al*, "Focused crawling: a new approach to topic-specific Web resource discovery," *WWW*, 1999.

[2] J. P. Lage *et al*, "Automatic generation of agents for collecting hidden web pages for data extraction," *Data and Knowledge Engineering*, 2004.

[3] L. Lin *et al*, "Meta-search Based Web Resource Discovery for Object-Level Vertical Search," *WISE*, 2006.

[4] Z. Nie *et al*, "Object-level Vertical Search," *CIDR*, 2007.

[5] M. L. A. Vidal *et al*, "Structure-Driven Crawler Generation by Example," *SIGIR*, 2006.