Focused Web Crawling Mechanism based on Page Relevance

Emil Gatial[1], Zoltan Balogh[1], Michal Laclavik[1], Marek Ciglan[1] and Ladislav Hluchy[1]

[1]Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 845 07 Bratislava, Slovakia
{emil.gatial, balogh.ui, laclavik.ui, marek.ciglan, hluchy.}@savba.sk

In this article we propose the system for focused web crawling used in a domain of job offers. This work is performed within the scope of NAZOU project. The results of this paper will be focused on algorithms for web page content analysis in order to specify the depth of crawling and identification of page relevance. The open source Arachnid tool is used as a ground crawling technique for web page downloading and furthermore it is extended by the proposed decision algorithm that will identify whether the page falls into the specified domain or not. In this article, the simple text analysis technique is described, which is based on searching the keywords from the specified domain, its synonyms and prescribed word orders. We conclude with the probability evaluation of the page relevance and other possible techniques that could enhance crawling process.

Introduction

The web crawler is a program that automatically traverses the web by downloading the pages and following the links from page to page [Koster 1999]. A general purpose of web crawler is to download any web page that can be accessed through the links. Contrary to this approach, a focused crawler tries to download only pages with specific topic to avoid the irrelevant web documents and reduce network traffic.

The most recent scientific results in the domain of large scale web crawler engines come from techniques described for search engines such as Google, Yahoo, etc.

One of the first focused crawlers was described in paper [Chakrabarti et al. 1999]. This focused crawler is based on the keyword relevancy evaluation and is composed of two hypertext mining programs: *classifier* and *distiller*. The *classifier* component evaluates the relevance of the page and the *distiller* identifies the hypertext links that points to many relevant pages.

Another very interesting focused crawler was proposed by Cho [Cho et al. 1998]. The system processes the “most important” pages first for the efficiency reasons. Authors took into account various measures of importance for a page e.g. similarity to a driving query, number of pages pointing to this page (backlinks), pagerank and location (in a hierarchy). The pagerank defines the qualitative attribute that is evaluated as a weighted sum of the pageranks of the pages which point to it.

There are three kinds of focused crawling strategies:

- BestFirst crawler: priority queue ordered by similarity between topic and page where link was found.
- PageRank crawler: crawls in pagerank order, recomputes ranks every 25th page.
- InfoSpiders: uses neural net, back propagation, considers text around links.

Overview of proposed crawler

Proposed kind of focused crawler is divided into two components. The former crawls the web sites,

makes decision about the depth of site crawling. The component employs an Arachnid, an open source tool for web crawling, and implements the simple site mapping and decision algorithm. The latter computes the document relevancy according to the keywords given for particular domain. The components are tightly coupled to provide real functionality of focused crawler.

Web crawler component

As mentioned earlier, the component uses Arachnid tool based of Java. It allows defining the steps after the page is downloaded by the interface extension as it is depicted on an example of source code bellow. The crawler starts with the predefined URL list of initiation sites.

```
public class Spider extends Arachnid {
  SiteMap site;
  public Spider( String base ) throws MalformedURLException {
    super( base );
    site = new SiteMap( base );
  }
  protected void handleLink( PageInfo pageInfo ) {
    URL[] links = pageInfo.getLinks();
    // compute relevance

    // if page was not downloaded yet, then download
    if( !site.isLoaded( pageInfo.getUrl().toString() ) ) {
      String filename = "/tmp/cache/" + pageInfo.getTitle() + ".html";
      site.markAsLoaded( pageInfo.getUrl().toString(), filename );
      downloadPage( pageInfo, filename );
    }
        // store metadata about page
  }
  protected void handleBadLink(URL url,URL url1,PageInfo pageInfo) {
  }
  protected void handleNonHTMLlink(URL url, URL url1,PageInfo pageInfo) {
  }
  protected void handleExternalLink(URL url,URL url1) {
  }
  protected void handleBadIO( URL url, URL url1 ) {
  }
}
```

Several basic functionalities must be implemented in focused crawler:

- Recognize already visited pages
- Detect the page change and make updates
- Page absence detection and invalidate the content of downloaded page
- Calculate the depth of crawling

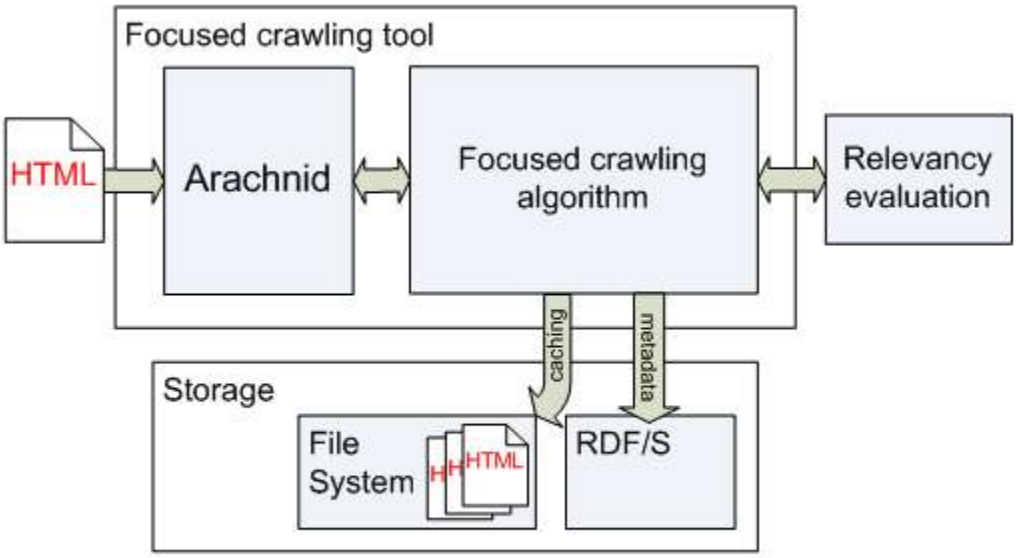

Figure 1: Block scheme of focused crawler

The proposed focused crawler (Figure 1) writes copy of relevant page into the cache and writes metadata about successfully downloaded pages into an RDF database. It includes page URI, date and time of download, pathname in the cache, etc.

Relevance evaluation component

As long as, we interested in information processing for the domain of job offers, the Internet offers too many irrelevant information. Furthermore, there is no rule for web page formatting and layout. To deal with such variety of sources, it must be designed tool for page content processing that takes into account characteristic keywords relevant for given domain of use and its synonyms.

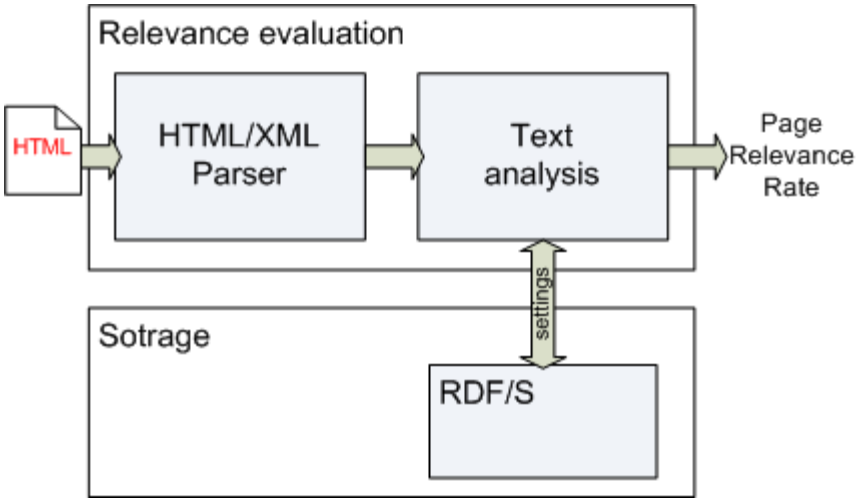

Figure 2: Block schema of relevance evaluation component

The component (Figure 2) receives the page content for relevancy evaluation. It parses the content and tries to match the defined set of keywords. A single keyword is defined as a regular expression that describe format that identify (with some probability), that the page falls into the investigated domain.

Every keyword is coupled with the weight, which comes into the relevancy evaluation process of entire page.

Method of relevance evaluation

Relevance evaluation of downloaded web page is the most important role in the process of focused crawling. The time of web site search is highly influenced by the speed and quality of relevancy evaluation of a web page and therefore it is the critical part of web crawler. Method that evaluates the relevance of downloaded page comprises the context identification, evaluation of particular weight for given keyword and calculation of page relevance.

Context identification

The quality of relevance evaluation is proportional to the size of dictionary of relevant keywords. Therefore is important to handle large dictionary of the most relevant keywords for given domain.

For example, some keywords can be identified for the domain of job offers:

1. *at least [\d]* years experience* - statement identified as an requirement for an applicant
2. *salary: [\d\.]** - statement for an offered salary

Moreover, the synonyms for specific keywords (e.g. salary, position, etc.) have to be defined and put into the page processing to process variety of pages.

The dictionary of relevant words could change from time to time. Therefore, the useful tool for automatic synonym update can enhance relevance evaluation. For example, it could be created with the support of WordNet [WordNet] dictionary located at Cognitive Science Laboratory of Princeton University. Such tool should be executed in initialization phase or whenever the dictionary of relevant words is changed.

Weight of keyword

As it was mentioned earlier the rate of relevancy is computed as weighted sum of each occurrence of keyword in the content of page. Therefore, the component for relevancy evaluation must maintain the number of occurrences for particular keyword in all relevant pages and in all irrelevant pages. The weight value, that expresses the relevancy rate, could be simply evaluated as:

$W(kw) = N_r(kw)/P_r - N_i(kw)/P_i$

, where

W(kw) is relevancy weight for the keyword “kw”, $N_r(kw)$ is number of keyword “kw” occurrences in pages evaluated as relevant, $N_i(kw)$ is number of keyword “kw” occurrences in pages evaluated as irrelevant, Pr is number of relevant pages and Pi is number of irrelevant pages.

The weight W(kw) express the difference between the rate of keyword “kw” found in relevant and irrelevant pages. The W(kw) shouldn't be greater than 1 provided that the keyword “kw” can be found in the content of relevant page only once. This method enables dynamically identify the relevant,

neutral and irrelevant keywords. The keywords identified as the most irrelevant should be excluded from keyword set and substituted with others. The values of variables that come into W(kw) evaluation must be updated after every calculation of page relevance.

Calculation of page relevance

The rate of relevancy for particular page can be treated as weighted sum of context rate for the largest relevant group of keywords. It can be calculated according to the following formula:

$R(page) = Sum(W(kw))/N_{kw}(page)$

, where

R(page) is the relevancy rate of page for given domain characterised by set of keywords, W(kw) is weight of keyword “kw” and the $N_{kw}(page)$ is number of keywords found in the content of page. The sum is evaluated for every keyword “kw” found in the content of page.

The value of predefined threshold states whether the page is relevant or not. However, the value of W (kw) depends on number of pages evaluated as relevant and irrelevant, the convenient value of threshold and the set of keywords could separate the miscellaneous pages into groups of relevant and irrelevant for given domain.

Other relevance evaluation mechanisms, such as pagerank, backlinks, evaluation of location (in hierarchy), could be included into the calculation of page relevance, but the relevance evaluation based on keywords from particular domain could separate relevant pages with greater probability.

Crawling strategy

The relevance evaluation is used, particularly, to make decision whether to download the page or not, but it can be used to influence the depth of crawling. The crawling algorithm walks through the links found in the content of the page, identifies pages that weren't yet put into crawling process and calculate the relevance for particular page. If most of the pages were recognised as irrelevant, the algorithm could stop the further crawling for that site. However, depth of crawling can be defined implicitly, but the described crawling strategy is able to exclude irrelevant sites and process relevant sites into much greater depth.

Conclusion

The described focused crawling mechanism is able to process web sites and identify the relevant page according to its content text analysis. Moreover, the crawling strategy could stop crawling process for irrelevant sites. The research, implementation and testing of crawling and page relevance evaluation algorithm is performed within the scope of NAZOU project. The further enhancements of proposed focused crawler could be realized during the testing phase.

References

[Koster 1999] "The Web Robots Pages", M. Koster. 1999.

http://info.webcrawler.com/mak/projects/robots/robots.html

[Chakrabarti et al. 1999] "Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery", S. Chakrabarti, M. van den Berg and B. Dom. In Proceedings of the 8th International WWW Conference, Toronto, Canada, May 1999.

http://www8.org/w8-papers/5a-search-query/crawling/index.html

[Cho et al. 1998] "Efficient Crawling Through URL Ordering", J. Cho, H. Garcia-Molina, L. Page. In Proceedings of the 7th International WWW Conference, Brisbane, Australia, April 1998.

http://www7.scu.edu.au/programme/fullpapers/1919/com1919.htm

[Crimmins 2001] Francis Crimmins, 10 September 2001

http://dev.funnelback.com/focused-crawler-review.html

[WordNet] Web page of WordNet project at Cognitive Science Laboratory of Princeton University

http://wordnet.princeton.edu/obtain