International Journal of Hybrid Information Technology
Vol.6, No.6 (2013), pp.187-194
http://dx.doi.org/10.14257/ijhit.2013.6.6.16

Focused Crawler Research for Business Intelligence Acquisition

Peng Xin and Qin Qiuli

School of Economics and Management, Beijing Jiaotong University
No.3 Shang Yuan Cun, Hai Dian District, Beijing 100044, Peoples R China

12120607@bjtu.edu.cn, qlqin@bjtu.edu.cn

Abstract

The internet has become indispensable part of people's life. For enterprises, there are mass of valuable information in the internet. It not only includes competitor information, but also includes customer's evaluation of products. These information is an important source of business intelligence. This paper aims to build a focused crawler to filter business intelligence from vast amounts of information in the internet. The crawler takes a certain number of web pages as seed. Then extract URLs in these pages, and parse main text of every URL. After that, the crawler calculates relevancy between every main text and the crawler's topic based on VSM (vector space model) and TF-IDF (Term Frequency-Inverse Document Frequency). If a web page is relevant, it will be saved; otherwise, it will be discarded. At last, an experiment is done to test the performance of crawler. It can be seen that the recall rate and accuracy of the crawler is very high though the result of this experiment.

1. Introduction

Recent years, especially from 2000, information technology has made great development. Internet is widely used in people's life. According a report published by Pingdom on January 16th 2013, which is company in Sweden, there are 2.4 billion internet users worldwide, and there are 634 million websites. So there is massive information in the internet (Pingdom, 2013). Massive amounts of business intelligence are hidden in the internet. If an enterprise makes good use of it, it must bring great profit (M. Castellanos *et al.*, 2013). How to filter out useful information challenges the enterprises. Traditional search engines such as Baidu and Google, can partly meet the demand of retrieving information from internet, yet there still lies some drawbacks (Zhao, 2010):

(1) Traditional search engines try to index the entire internet, so the retrieval results contain various types of web pages. However, the profession background and search intentions of web users are different in many ways, thus fails to meet field or topic oriented needs from advanced users;

(2) Traditional search engines are based on keyword search, thus cannot be implied from the keywords of semantic retrieval result, or provide users with what they really want;

ISSN: 1738-9968 IJHIT

(3) Current search engines are unable to meet the personalized requirements of users. Meanwhile, due to the lack of comprehensive treatment of the retrieved results, it's laborious to find information they want from a large number of results.

In order to solve these problems, focused crawler is proposed. Focused crawler is a web crawler which only fetches web pages related scheduled topics or fields. When fetching a web page, it determines whether the page is related to the topic pre-determined. If true, it will be saved; otherwise discarded. The domain knowledge base can be as simple as a theme or a set of keywords, can also be a collection of field information. Compared with the traditional web crawlers which pursue volume and the completeness of features, focused crawler pursue precision. Besides, it aims at a certain classes or some kinds of theme to grasp the target information quickly and accurately. Also it increases the recall rate as well as guarantees the accuracy (Diligenti *et al.*, 2000) (Zhou *et al.*, 2005).

This paper researches on the focused crawler for business intelligence acquisition. Compared to other focused crawler, it's different from two aspects. The first is the crawler attaches weight to every URL according to website attributes. This paper proposes a method to calculate a suitable weight for a website, so the crawler can download valuable intelligence in time. The second is that the crawler's correlation analysis method is based on vector space model of TF - IDF text relevancy calculation method. This method can filter out irrelevant web pages efficiently. The first part of this paper investigate on the basic flow of the topic focused crawler; the second part presents the implementation of the focused crawler in detail; the third part shows the experimental analysis, verifying the accuracy and recall rate of this method through the crawler experiment; the fourth part concludes this paper.

2. The Process of Focused Crawler

Web crawler is a web robot, grabbing public web pages from internet, and it's an important part of the search engines. Traditional web crawler finds web pages by links in web pages. It starts from a page of website (usually the home page) and gets the content, then go on to find other links in it. Though these links it find more links. This procedure proceeds onto complete web scraping on this site. If regard the internet as a web site, network spiders can use this principle to crawl down all web pages on the internet (Zhu *et al.*, 2012)(Wei *et al.*, 2011).

For users in the enterprises, focused crawler can help them to get information from internet quickly. This part focuses on the process of the topic focused crawler, which is shown in Figure 1.

For this focused crawler, there is a URL queue waiting for fetched. The initial of this queue is the seed URLs. When crawler starts, it pulls a URL from this queue, and downloads the corresponding web page. Then parse the web page and main elements, such as title, content, pubtime, etc. Based on these elements, the crawler analysis whether it's relevant or not. If true, it will be saved in the database of relevant page; otherwise it will be discarded. For pages links, the crawler query everyone from the data table of existed URL, and push new URLs into the URL queue. This is a whole process dealing with a web page. The crawler loops execution, until finish processing all URLs in the collection queue.

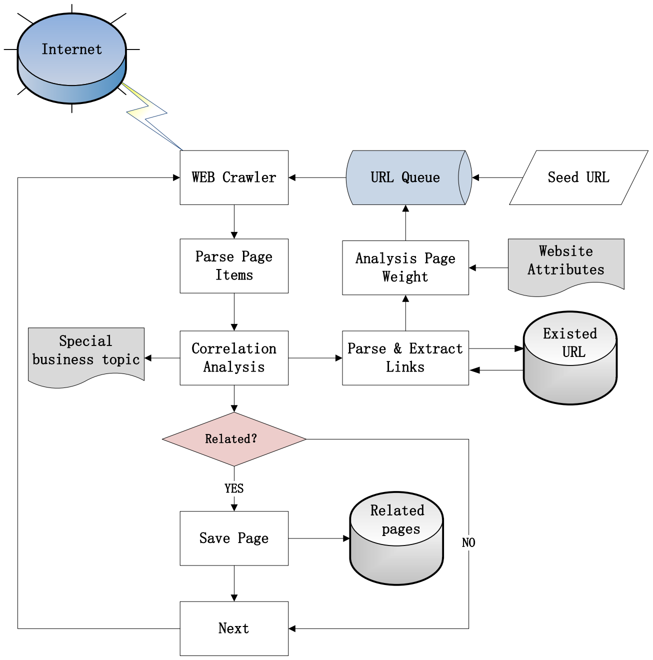

Figure 1. The process of focused crawler

When pushing new URLs into the queue, the crawler attaches weight to every URL according to website attributes, such as Alexa and PR (PageRank). The reason is that if a website's Alexa is top, or its PageRank is large, that usually means this website has more users than other website and it has great influence. If a information is published by a large website and a small website at the same time, the former's audience is certainly more than the latter's. For this reason, enterprise should give priority to acquire intelligence from large website.

3. Key Technologies

The key technologies of this crawler are as follows.

3.1 Web page pretreatment

The purpose of web page pretreatment is to clean HTML content and create page DOM tree. Web page cleaning includes several parts, including:

(1) Eliminating script codes which may influence analysis, such as the content of the script node and the action script in normal web page nodes;

(2) Removing style codes in the page, such as the content of the style node and style attribute of normal web page nodes;

(3) Eliminating commented code that is comments in the web page;

(4) Rejecting other irrelevant code, such as banner ads, copyright statement and so on.

After these works, crawler builds the DOM (DOM is the Document Object Model) tree, which is recommended as the extensible markup language standard programming interface by the W3C. Figure 2 is a DOM tree which is built from a simple web page.

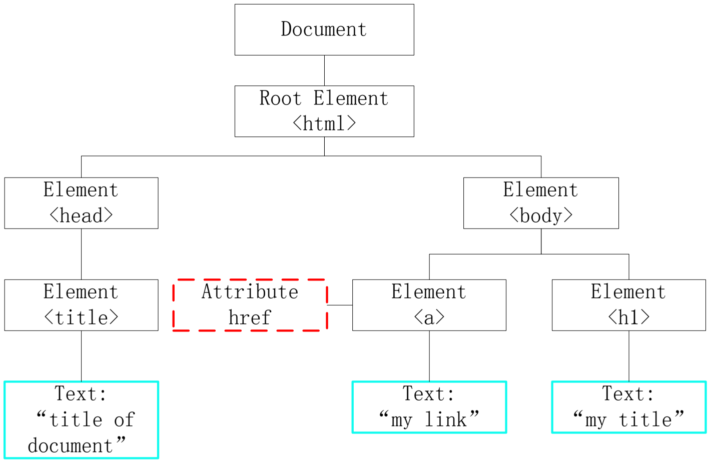

Figure 2. A demo of DOM tree

From the DOM tree, it is easy to find out the link nodes, text nodes and their superior nodes (Wang *et al.*, 2004). This facilitates our further analysis. This paper uses the DOMParser to parse HTML. Considering the non-standard nodes that may exist in some pages (such as lack of common ending tag), DOMParser does refines this kind of label. This guarantees the DOM tree is complete and specification.

3.2 Web page analysis

Web page analysis aims to extract the text and links in the web page. The text will be used to analyze the relevant, while the links can be used as the URL seeds in the next grasping phase of the crawler. The following parts describe these in more detail.

3.2.1 Extracting text from web pages

It means that extracting the main text from web pages such as the news content in the news web pages and the blog contents form the blog post. Here we take Sina as an example. By comparing home page with a detail page of news, we can easily find that the main text exists in the news page (what we should note here is that the text refers to the non-link text which should be separated from the link text, the link node - the text corresponding to the A node is referred as anchor text), while in the home page the whole text does not or seldom exists. From the DOM tree, there are many long text paragraphs nodes in news page, such as the ‘P’ node, ‘SPAN’ node and others; while in the Home page, there are many link nodes – ‘A’ node.

The process of extracting text is as follows: first, we traverse up the DOM tree and calculate the length of text in all general text (except the anchor text) in each node, we take it as *TL*; second, we calculate the ratio between the length of text in each node with the length of total text in web page by formula (1), and take the rate as *NR*. If NR is greater than a certain threshold (the threshold is generally set to 0.3), then it will be identified as a possible text node. Meanwhile, we calculate another ratio *UKR* on number of nodes by formula (2). This ratio is between the numbers of sub-nodes which has normal text in each node, we take it as *UNN*, with the total number of node which has normal text. By this ratio, we can get the direct parent node of the main text, and the threshold is set to 0.5.

$$NR = TL / \sum_i TL_i \tag{1}$$

$$UKR = UNN / \sum_j UNN_j \tag{2}$$

Calculation in the first step is to get nodes which are possible text node. Based on the results of first step, next step is to confirm whether the node get in the first step is the text node by calculating the UKR.

3.2.2 Extract the links in the web page

Comparing with extracting text, extracting links in web page is relatively simple. First, we traverse up the DOM tree to get all ‘A’ node whose ‘HREF attribute isn’t null and isn’t JavaScript. Then we get the anchor text and ‘HREF attributes of each node, and make unified conversion to the link addresses. Finally, we rule out navigation links and advertising links according to the content and length of anchor text.

3.3 The algorithm of calculate website’s weight

This algorithm calculates weight of websites according to the following attributes:

(1) Alexa ranking: it's a web traffic report provided by Alexa Internet Inc a subsidiary company of Amazon.com. This company's website ranking system tracks over 30 million websites worldwide. Millions of people from across the globe visit Alexa.com each month to access their web analytics and other services. A large site always has great traffic, so it has a top rank.

(2) PageRank: it's is a link analysis algorithm and is used by the Google web search engine. A PageRank results from a mathematical algorithm based on the webgraph, created by all World Wide Web pages as nodes and hyperlinks as edges. The rank value indicates an importance of a particular site. Based on this algorithm, Google calculate the PageRank for every website. The value is an integer from 0 to 10. The website’s PageRank is larger while it’s more important.

(3) User-defined: there are many special websites for every enterprise or industry. So we define five classes for these sites. Fifth class is most important or concerned websites.

Considering these three properties, we calculate weight of websites by the formula (3):

$$W_s = \frac{1}{\left\lceil {AR_s}/{1000} \right\rceil} \lambda_1 \times 100\% + \frac{PR_s}{10} \lambda_2 + \times 100\% + \frac{U_s}{5} \lambda_3 \times 100\% \tag{3}$$

In this formula, W_s is the weight of website s, and AR_s is the Alexa Ranking of s, and PR_s is PageRank, and U_s is user-defines class of s, while λ_1, λ_2 and λ_3 is weight of every attribute. Every attribute is converted to 100 in order to unify accounting unit. Considering to that Alexa Ranking aims at millions of websites, if take the reciprocal directly, the result will be will small with great probability. So we make it divided by 1000, and round up the fare. Otherwise, λ_1, λ_2 and λ_3 are used to distinguish between the importance of these attributes, and sum of them is 1.

3.4 The TF - IDF text relevancy analysis based on the VSM

VSM (Vector Space Model) is an algebraic Model applied in the information filtering, information retrieval, index and relevance analysis. Since it was proposed by Salton and others in the 1960s, it has been successfully applied to the famous SMART text retrieval system (David *et al.*, 2010). VSM takes each document as a vector, and obtain the similarity between different documents by calculating the angle between vectors. The model is proposed based on the following contemplation: the semantics of documents is expressed by words. If you take each word in the document as a vector, then you can compare the documents or the query keywords to determine their similarity. Here, in this paper, the documents or keywords on special filed are regarded as a kind of query. Then we build query vectors, and analyze whether the document is related to this field by calculating the cosine of the angle between a document and the query vector. Figure 3 represents the basic idea of the VSM applied in this document, including mapping the field of thematic as a basic vector and four sample document vectors. Through individually calculating the vector spaces between the vector S with vector D1, D2, D3 and D4 to get whether the document belongs to this field.

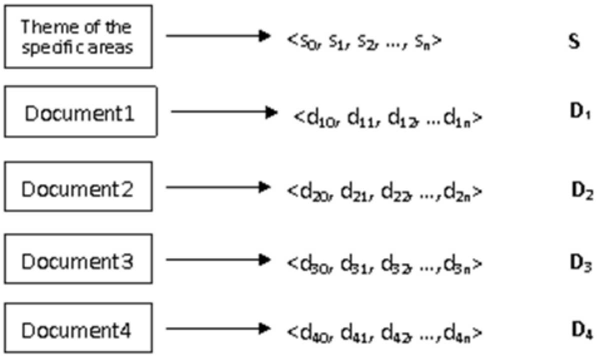

Figure 3. VSM

For every different word in the document collection, VSM records it as a component in the vector. For example, word ‘a’ appears twice in document D1, then the corresponding value of vector component of word ‘a’ is 2. Each word in the document is treated fairly here, and the importance of each word in expressing the theme of document has not been taken into consideration. In fact, some words are indeed more important than others, so we put a weight to each word. The specific weight of each word is determined by the frequency of the data set. For a given word, its weight is calculated by using the IDF (Inverse Document Frequency). We define the following variables:

t: number of different words in the document collection

tf_{ij}: times of word ‘t_j’ appears in Document D_i, that’s term frequency

df_j: number of documents which contain the word ‘t_j’

idf_j: $\lg \frac{d}{df_j}$, ‘d’ is the number of all documents

In every document, the weight of each word is determined by the frequency it appears in the whole document collection or in a specific document, and then it can be calculated and get the value. While the calculation, the main consideration is the value of TF and IDF. Since there are lots of specific calculation formulas, this paper uses a common formula.

$$w_{ij} = \frac{(\lg tf_{ij} + 1.0) \times idf_j}{\sum_{j=1}^{t} [(\lg tf_{ij} + 1.0) \times idf_j]^2} \tag{4}$$

This formula can effectively avoid the consequence of the high frequency matching words overwhelming other matching words. Using lg(tf)+1.0, it can narrow the scope of the word frequency, and with that we can accurately calculate the appropriate weights to words in different degree of importance.

As a result, this paper uses these two kinds of methods (TF-IDF and VSM) to initialize the field vocabulary database and calculate the initial semantic similarity between words in vocabulary database; it computes the initial semantic similarity between texts based on the initial semantic similarity. According to the initial semantic similarity between words and text, it does not stop alternating iterative calculating between semantic similarity of each text and the semantic similarity of words until convergence. After that, it constructs the ultimate meaning of similar matrix of all vocabularies according to the convergence results of iterative calculation; According to described ultimate meaning similarity matrix, it transforms the original text's word frequency vector into a new word frequency vector, and finally calculates the centered text relevancy of the described text.

4. Experiment

4.1 Evaluation index

In traditional information retrieval, the basic evaluation index of a system including: recall and accuracy. Recall is ratio of the number of relevant documents in retrieve results and the number of all relevant documents in the document library; Precision is ratio of the number of relevant documents in retrieve results and the total numbers of documents in retrieve results. Based on these indicators, we get the evaluation index of this experiment: recall ratio and accuracy ratio. Recall ratio is calculated by formula (5). n_{releInDB} is the number of web pages which are really relevant in the database of relevant page, while n_{total} is the number of relevant web pages within the scope of collection. Accuracy ratio is calculated by formula (6). n_{DB} is the number of web pages in the database of relevant page.

$$recall = \frac{\mathrm{n_{releInDB}}}{\mathrm{n_{total}}} \tag{5}$$

$$accuracy = \frac{n_{releInDB}}{\mathrm{n_{DB}}} \tag{6}$$

4.2 Results analysis

This experiment chooses the electronic commerce field as the target. We retrieve Chinese keywords '电子商务' in BAIDU news, and take the 20 document at the first page as seed URLs. The depth of the crawler is set up to 3. Finally we collect 1147 related web pages. Finally, through artificial filter, we get 868 web pages which are really relevant. Within the scope of collection, there are 944 relevant web pages. So the n_{releInDB} is 868, and n_{total} is 944, and n_{DB} is 1147.

According to the experimental formula of index, we can get the recall ratio is 91.95%, and the accuracy ratio is 75.68%. Based on the experimental results, it can be seen that the crawler can collect most of the targeting web page; while its accuracy rate is 75.68%, which means that about 75% of the crawler determined relevant pages is proved to be correct. After us analysis data created by this experiment, we find that there is 909 irrelevant URLS in 1023 URLS in the second loop. So the filtration rate is 88.9%. In the third loop, the filtration rate is 95.0%. The filtration rate of this experiment is 94.6%. So we can get that this crawler can help us to filter relevant information from lots of irrelevant, and it can provide high relevant data for our further study.

5. Conclusion

Internet is an intelligence source of enterprise that can't be ignored. But there are amounts of data in the internet, so enterprise should try every means to filter out data which they don't need. This article proposes a focused crawler for business intelligence acquisition. While the crawler acquires data, it attaches weight for the URL by the special algorithm. Meanwhile, it analysis whether the web page is relevant based on VSM and TF-IDF. Finally, it can be seen the focused crawler's performance on recall and accuracy is very well, and it can fetch relevant web pages efficiently.

Massive amounts of business intelligence are hidden in the internet. If an enterprise makes good use of it, it must bring great profit. At the same time, it's a series of technology challenges how to help users make use of these data. The focused crawler is only one of these technologies. It can improve much better.

References

[1] D. A. Grossman and O. Frieder, "Information Retrieval: Algorithms and Heuristics (2nd Edition)", Beijing, China, **(2010)**.

[2] Pingdom: Internet 2012 in numbers, http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/.

[3] Q. Q. Jiang, Z. Gong and Y. Xin, "Design and Implementation of BBS Information Extraction System Based on HTML Parser", Techniques of Automation and Applications, vol. 1, **(2012)**, pp. 32-37.

[4] M. Castellanos, F. Daniel and I. Garrigós, "Business Intelligence and the Web", Information Systems, vol. 15, **(2013)**, pp. 307–309.

[5] M. Diligenti, F.M. Coetzee, S. Lawrence, C. L. Giles and M. Gori, "Focused Crawling using Context Graphs", 26th International Conference on Very Large Databases, VLDB 2000, Cairo, Egypt, **(2000)**, pp. 527–534.

[6] Q. Wang, S. W. Tang, D. Q. Yang and T. J. Wang, "DOM-Based Automatic Extraction of Topical Information from Web Pages", Journal of Computer Research and Development, vol. 10, **(2004)**, pp. 1786-1792.

[7] J. J. Wei, D. D. Yang and X. W. Liao, "Focused Crawler Based on Improved Algorithm of Web Content Similarity", Computer and Modernization, vol. 9, **(2011)**, pp. 1-4.

[8] L. Z. Zhou and L. Lin, "Survey on the research of focused crawling technique", Journal of Computer Applications, vol. 9, **(2005)**, pp. 1965-1969.

[9] Y. P. Zhao, "Comparative Study of Services of Common and Semantic Search Engine", Journal of Information Science, vol. 2, **(2010)**, pp. 255-270.

[10] Z. Minand L. Shengxian, "Research of a Focused Crawler to Specific Topic Based on Heritrix", Computer Technology and Development, vol. 2, **(2012)**, pp. 65-68.