International Journal of u-and e-Service, Science and Technology
Vol.7, No.5 (2014), pp.117-128
http://dx.doi.org/10.14257/ijunnesst.2014.7.5.11

Research and Implementation of a Vertical Search Engine in the Financial Domain

Yue Hou-guang[1],ZhangLing[2], Meng Fan-jun[3] and Songhong-hao[1]

[1]*School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan 250014, China*
[2]*Library, Shandong University of Finance and Economics, Jinan 250014, China*
[3]*Computer& Information Engineering College, Inner Mongolia Normal University, Hohhot010022, China*
yuehg71@163.com

Abstract

The exponential growth of the data on the web makes it difficult for web users to locate and obtain the interesting information. The general web search engines, such as Google, Baidu, reduce the degree of difficulty to some extent. The rise of vertical search engine provides a promising approach to address the big challenge. A Chinese vertical search engine is urgently needed to provide professional financial information retrieval service for related financial institutions and government agencies. Therefore, in this paper we investigate, design and implement a vertical search engine in the financial domain. Based onNutch plug-in mechanism, we provide and implement the details of a vertical search engine in the financial domain, especially in the finance domain focused crawler. Experimental results show that, the finance domain focused crawler based on Nutch plug-in mechanism has good performance and can satisfy practical requirements of medium search application, and the system runs well.

Keywords: *financial focused crawler; nutch plug-in; vertical search engine; Hadoop platform*

1. Introduction

Along with social and mobile Internet wave, the data on the network shows explosive growth, the big data era has come. How to deal with these data is not only a challenge faced by search engines, but also an opportunity for search engines [1]. Vertical search engine is a new web retrieval service mode proposed in this context. Vertical search engine is the professional search engine fora particular area or industry, which is an extension and refinement to general search engine. It is characterized by "specialized, refined, deep" and has great color of domains and industries [2].

The vertical search engine technology is very similar to general search engines, but there are differences in emphasis and search strategies. In recent years, the vertical search engine has become a hot research point of search technology and a number of vertical search sites that have practical significance were produced:

(1)Kosmix: founded in 2005 and provides applications search services. It collects web pages which is from more than 10,000 sources and extracts applications information to provide the classification result of applications for users.

(2)Zillow: founded in 2006 and provides real estate information search service. Zillow users can use it to search real estate information of the location they are interested in.

ISSN: 2005-4246 IJUNESST

(3)Pluggd: founded in 2007 and provides audio and video vertical search services. Pluggd users search the video or audio files by entering the keywords. But be different with other sites which tag videos and audios manually, Pluggd describes the contents of videos and audios through the text recognized from videos and audios by voice recognition technology.

(4)Retrevo: One of the world's most successful vertical search engines. Retrevo can collect the articles which relate to product reviews in each web site for analyzing and indexing. Retrevo users can search products by product brand or model, then centralized obtain the evaluations of other consumers for the products as a reference and formed the comprehensive objective evaluation of the product.

With China's rapid economic development and further opening up, a Chinese vertical search engine is needed to provide professional financial information retrieval service for related financial institutions and government agencies. Therefore a Chinese vertical search engine in the financial domain was implemented in this paper and the financial focused crawler is the focus of the study.

Building vertical search engine must first build a high-performance focused crawler. Currently the research of focused craw lera broad is almost all concentrated on focused algorithm, but does not take into account the difficulties of practical applications about focused crawler technology, such as development difficulty, costly hardware for medium applications. The appearance of nutch and Hadoop framework can change the above status quo. Therefore, in this paper we use the plug-in mechanism of nut chopen source framework on Hadoop plat form to build the focused crawler oriented to medium vertical search applications. We also develop an improved do main keyword algorithm for resolving problems in building focused crawler. Throughout this paper we take the financial vertical search engine as example to detail the development process.

2. The Characteristics of Nutch and Hadoop

Nutch is an excellent open source search framework based on Lucene [3], which belongs to the top-level projects of Apache Software Foundation (ASF). Nutch bottom uses HadoopDistributed File System (HDFS) as the data storage platform to implement the distributed data storage [4]. It uses the Map /Reduce distributed programming framework of Hadoop for data processing. Hadoop has an outstanding feature of running in ordinary and in expensive commodity hardware, which solves the problem of high hardware requirement of focused crawler. This design makes nutch have capabilities of processing and storing massive data, which is easier to build distributed focused crawler [5]. Nutch architecture is shown in Figure 1.

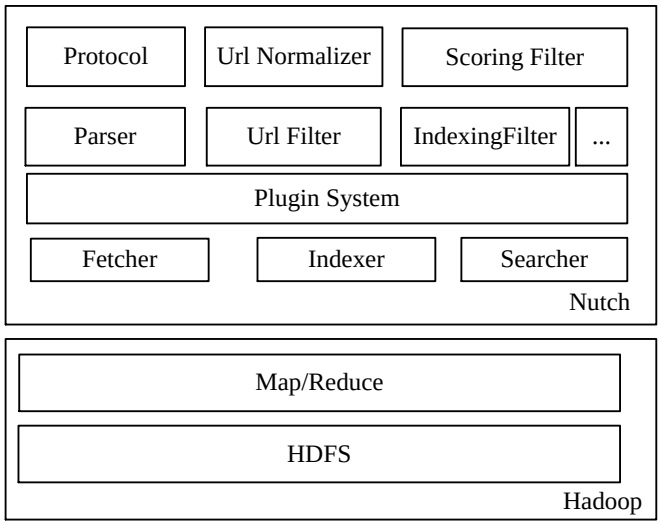


Figure 1. Nutcharchitecture

For Nutch, its development language is fixed and the core module is determined and it does not involve security, transaction, desktop applications, so huge perfect plugin framework is not suitable [6]. Nutch plugin mechanism has the following three design concepts:

-Scalability: Nutch developers can implement given interfaces to extend Nutch functions.

- Flexibility: Nutch administrators can select appropriate plug-ins according to specific needs to customize their search engine.

- Maintainability: developers only need to focus on their own areas without having to consider other things. Kernel developers only need to write Nutch core engine and provide the interfaces description for plug-in developers. Plug-in developers only need to focus on the functions of the plug-in they developed. This makes the Nutch code structure more simple and makes Nutch easier to maintain and more robust.

Nutch plugin mechanism gives Nutch features of easy expansion, easy to develop, easy maintenance. Nutch plug-in mechanism enables developers to focus on crawling, indexing and query strategy, reducing the development difficulty.

3. Overall Design of Vertical Search Engine in Financial Domain

Similar to general search engines, vertical search engine system consists of three subsystems:web pages collecting and processing subsystem,indexing subsystem, retrieval subsystem. It’s working principle shown in Figure 2.

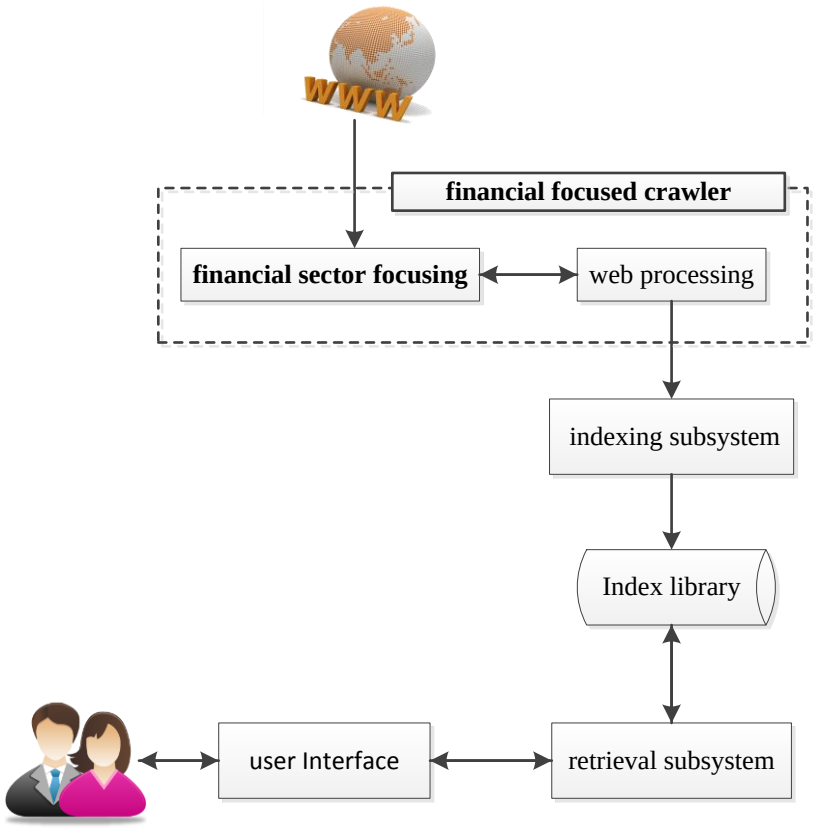


Figure 2. The Working Principle of Vertical Search Engine in Financial Domain

(1) Web pages collecting and processing subsystem:also known as web crawlers, is responsible for collecting web pages on the Internet. The financial focused crawler is also responsible for focusing financial domain web pages and extracting texts of web pages. In this paper, we use the financial focused plug-in which is based on Nutch plug-in mechanism to achieve the function of financial focused crawler. The financial focused plug-in plays an important role in the design of vertical search engine.

(2) Indexing subsystem:processes the texts extracted by web processing model. It indexes web texts and stores the index file to the index database for the use of retrieval.In this paper,we use the default indexing system of Nutch as the indexing subsystem.

(3)Retrieval subsystem:receives the search requests from users and retrieves the data which meets the users'query from index database to return to users.In this paper,we use the default retrieval system of Nutch as theretrieval subsystem.

4. The Design of Financial Focused Plug-in

Nutch plug-in mechanism involves three concepts: Nutch kernel, extension points and extension. Extension is the class that implements predefined functions of the extension point.Extension is matched with extension point. Developers can enhance the functionality defined by extension point. For an extension which implements an extension point, it must implement the interface defined by the extension point and return the value of predefined type. An extension point can have multiple extensions and each extension implements an

interface defined by the extension points. The relations between Nutch kernel, extension points and extension are shown in Figure 3.

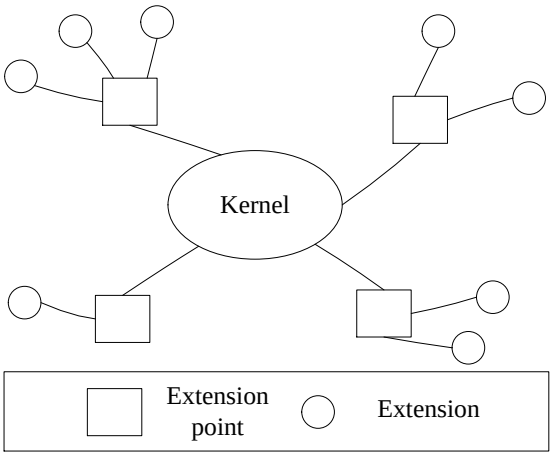


Figure 3. The Relations between Nutch Kernel, Extension Points and Extension

All core functions, including page analysis and scoring, URL filtering and normalization, indexing and query strategy, are implemented by extension in Nutch. To extend Nutch, it reserves a lot of extension points and provides basic implementations of these extension points, such as "Parser", "IndexingFilter", "URLFilter", "ScoringFilter" and so on.

Nutch kernel can select the appropriate plug-in based on the file type to parse the content of crawled file. A focused plug-in that implemented "Parser" extension point was designed in this paper.It replaces thedefaultHtmlParserplug-in to parse html file so that Nutch can focus on financial domain web page.

The design of financial focused plug-in mainly consists of three steps:

First, extract the texts of the web pages. The web pages not only include the content that we are interested in, but also include other noise content, such as advertising and licensing information. In html parsing process, the noise information must be filtered and the text must be reserved in maximize degree. The NekoHtml tool was used to build the DOM tree [7] of the input html document and this article extract the web text according the following rules:

(1) Remove the labels that cannot contain text in the DOM tree, such as FROM, SELECT, IFRAME, INPUT, STYLE, SCRIPT nodes and so on.

(2) Extract the texts of the H1, H2, H3 stressed labels in the DOM tree. Generally,these labels contain the important information of the web content.

(3) Generally, the labels that have the ancestor relations with H1, H2, H3 and other similar labels contain most information of the web content. Select these ancestor labels in a certain threshold and extract their text.

(4) Extract the texts of meta properties which describe the web pages, such as keywords, title. These properties always contain important attributes of the web pages.

Then, segment the extracted web texts to words. This article used IK-analyzer for Chinese word segmentations and in order to improve the accuracy of segmentation, over 2300 Chinese financial domain words selected by financial experts was added to the extension lexicon of IK-analyzer. By testing segmentation efficiency on 200 financial domain web documents, obtained this method could validly identify financial domain words in the web pages.

Finally, focus the domain of web pages. Select the words that have obvious characteristics of the financial domain from 2300 collected Chinese words, such as finance, stock, equity

market, securities, bonds, funds, wealth management, investment, banking, insurance, trust, futures, foreign exchange, gold, Hong Kong stocks, as the Important Domain Keywords Library (IFKL). Furthermore build the Common Domain Keywords Library (CFKL) using remaining words. In order to improve configurability, IFKL and CFKL are stored in the configuration file and will be loaded into memory when starting plug-ins. Considering IFKL and CFKL are loaded only once and all operations on them are query operations, IFKL and CFKL were stored using hash tables in memoryto improve query efficiency.The financialdomain keywordslibrary used in this paperis shown in Table 1.

Table 1. The Financial Domain Keywords Library

Category	IFKL(Chinese Version)	CFKL(Chinese Version)
Stock	Finance|stock|limit |stockindex| component Index|hold | plates|	Stock|limit|cowhide city| composite Index | component Index|limit| opening|stockindex|lighten|margin call|GEM|smallplates|... (491 words)
Fund	Fund|raised|private| ETF |subscribe|LOF | |Recruitment	industry index| sector funds| fund risk|fund net income| companies fund | contractual fund | currency fund |… (254 words)
Futures	Futures|delivery|positions futures | futures margin | totalhand	futures contracts | margin | settlement |warehouse | brokered transactions | arbitrage |warehouse explosion | Total positions | hands Transactions |…(226 words)
Exchange	forex | hedge | exchange | currency translation| Rate Index | intermediate	haven currency | margin forex | forward foreign standard settlement date | reserved spot | hedge |export verification |… (266 words)
Gold	gold | gold reserves | paper gold | gold Investment |	gold | gold options trading | e-gold | international gold market | gold reserves | gold spot trading | non-monetary gold | gold spot trading |… (33 words)
Bond	treasury | bonds |debt financing | premium | conversion period	standard coupon discount rates | creditors |financing bills |repurchase transactions | Maturity risk premium | real estate mortgage bonds |… (240 words)
Bank	bank | excess reserves | reserve | deposit | loans | money supply	excess reserves | performing asset | hedge supplement rates | insurance against positions | thrift institutions |interbank placements | subordinated |… (311 words)
Insurance	insurance | premiums | coverage |Insurance regulation | Loss ratio |	subject matter of insurance | insurance certificates | policyholder dividends | policyholder dividends | insurance… (386 words)
Trust	trust | trust trading | trust business | trust list | trust beneficiary	agency collection and payment | guarantee witness services | risk-sharing | residential construction trust | housing trust | mutual funds |… (84 words)

Considering the same content in different type of labels has the different importancedegree for the whole page, the importance degree of the label content was divided into several grades in this paper. The texts extracted from title/keywords properties or H1-H6/B/U/I labels were called the important texts. The whole texts extracted from the web page were called common texts. Thus the traditional term weight calculation method was improved in this paper. That is sufficient to consider the importance degree of labels in the web page. Weights of different types of labels are not same. By total number of 1,000 web documents set, which included 200 pages of the financial domain, tested with the focus algorithm proposed in this paper, the reasonable weight values of different types of labels is shown in Table 2.

Table 2. Labelweights

Keywords	title	H1-H3	H4-H6	B/U/I	others
2.5	2.3	2	1.8	1.2	1

The web relevance is defined as:

$$sim(D) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} f(t_{ij})w(j)}{T(D)} \quad (1)$$

Where D is the document that contains n words, *i.e.*, $D = \{t_1, t_2, t_3 \ldots t_n\}$. $f(t_{ij})$ is the frequency of wordt_i appearing in type j .$w(j)$ is the weight of type j. $T(D)$ is the total number of words in document D. Bytotal number of1,000 webdocuments set, which included 200pagesofthe financial domain, tested with the focus algorithmproposedin this paper, when $sim(D)$ is greater than 0.035, can obtain a better focusing effect.

Algorithm 1 describes the improved domain focusing algorithm based on the web keywords. IFCL and CFKL are static member variables of the focusing class, which are built during class initialization, and are used directly in the algorithm.

Algorithm 1: the improved domain focusing algorithm based on the web keywords.

Input: the text of keywords meta attribute *kd_t*, the text of title label *tl_t*, the text of H1-H3 labels *h1_3_t*, the text of H4-H6 labels *h4_6_t* the text of B/U/I labels *bui_t*, the whole web content *text*.

Output: focused result which is boolean variable.

Focus(*kd_t*,*tl_t*,*h1_3_t*,*h4_6_t*,*bui_t*,*text*)

Begin

(1) Initialize *simd = 0.0* and *WordFrequency = Ø;*/**simd* is web relevance and *WordFrequency* is a hash table whose type is *HashTable<String,int[6]>*. Its keys are words and Its values are frequencies of words which appear in different types of labels.*/

(2) Segment*kt_t* to words and query each word of segmentation result whether appears in *IFKL*. If there is, then add it to *WordFrequency*and the variable whose corresponding variable subscript is 0 plus 1;

(3) For *tl_t*, *h1_3_t*, *h4_6_t* and *bui _t*, take the same approach as step (2), but when update word frequencies, the corresponding subscripts are 1,2,3 and 4;

(4) Apply variable *freq = 0.0* and get all records in *WordFrequency*, then store in the temporary Iterator*tmp*;

(5) While*tmp*.hasnext do

(6) Get a record and obtain its value, then store in the temporary array *a[6]*. *freq*+=*(a[0]**2.5+ *a[1]**2.3+ *a[2]**2.0+ *a[3]**1.8+*a[4]**1.2);

(7) End While

(8) Segment *kd_t+tl_t+h1_3_t+h4_6_t+bui_t* to words and get the number of important text words *total* from the segmentation result.

(9) *simd=freq/total*;

(10) If *simd>Threshold* Then return *true*;

(11) Else

(12) *WordFrequency* is set to *null* and *freq* is set to *0.0*;

(13) Segment*kt_t* to words and query each word of segmentation result whether appears in *CFKL*. If there is, then add it to *WordFrequency*and the variable whose corresponding variable subscript is 0 plus 1;

(14) For *tl_t*, *h1_3_t*, *h4_6_t*,*bui _t* and *text*, take the same approach as step (13), but when update word frequencies, the corresponding subscripts are 1,2,3,4 and 5;
(15) Get all records in *WordFrequency*, then store in the temporary Iterator*tmp*;
(16) While *tmp.hasnext*do
(17) Get a record and obtain its value, then store in the temporary array *a[6]*. *freq*+=*(a[0]**2.5+ *a[1]**2.3+ *a[2]**2.0+ *a[3]**1.8+*a[4]**1.2+*a[5]*);
(18) End While
(19) Segment *kd_t*+*tl_t*+*h1_3_t*+*h4_6_t*+*bui_t*+*text* to words and get the number of common text words *total* from the segmentation result.
(20) *simd*=*freq*/*total*;
(21) If *simd*> *Threshold* Then return *true*;
(22) Else return *false*;
(23) End If
(24) End If
(25) End

In order to evaluate the performance of the improved domain focusing algorithm based on the web keywords, this paper also implements the traditional domain focusing algorithm based on the web keywords.

Algorithm 2: the traditional domain focusing algorithm based on the web keywords.

Input: the text of keywords meta attribute *kd_t*, the text of title label *tl_t*, the whole web content *text*.

Output: focused result which is boolean variable.

Focus(*kd_t*,*tl_t* ,*text*)

Begin

(1) Initialize*simd* = 0.0 and *num* = 0.0;/**simd* is web relevance and *num* is the number of financial domain keywords.*/
(2) Segment *kd_t*+*tl_t*+*text* to words and get the number of the whole content words *total* from the segmentation result.
(3) Query each word of segmentation result whether appears in *CFKL*. If there is, then *num*++;
(4) *simd=num/total*;
(5) If simd>Threshold then return *true*;
(6) Else return *false*;
(7) End if
(8) End

Compared with the traditional algorithm, the web page keywords' locations are incorporated into determination information in the improved algorithm used by this paper. When focusing, the system will be given priority to the use ofthe important texts which can largely reflect the domain of a web page to judge the domain of web page. Only when focusing failed, it uses the common texts which contain more specific information and reflect the domain of web page implicitly to judge the domain of web page. These can significantly reduce the amount of calculation and improve the crawling speed of focused crawler.

The flow chart of focused plug-in is shown in Figure 4. As shown in Figure 4, kernel just provides the crawling web documents for focused plug-in, rather thanconcern with thespecificallyinternal process of the focused plug-in; Focused plug-in just analyzes the data from kernel and return the analysis result to kernel in the format required by the kernel, rather thanconcerns with how kernel crawl and store the web pages.

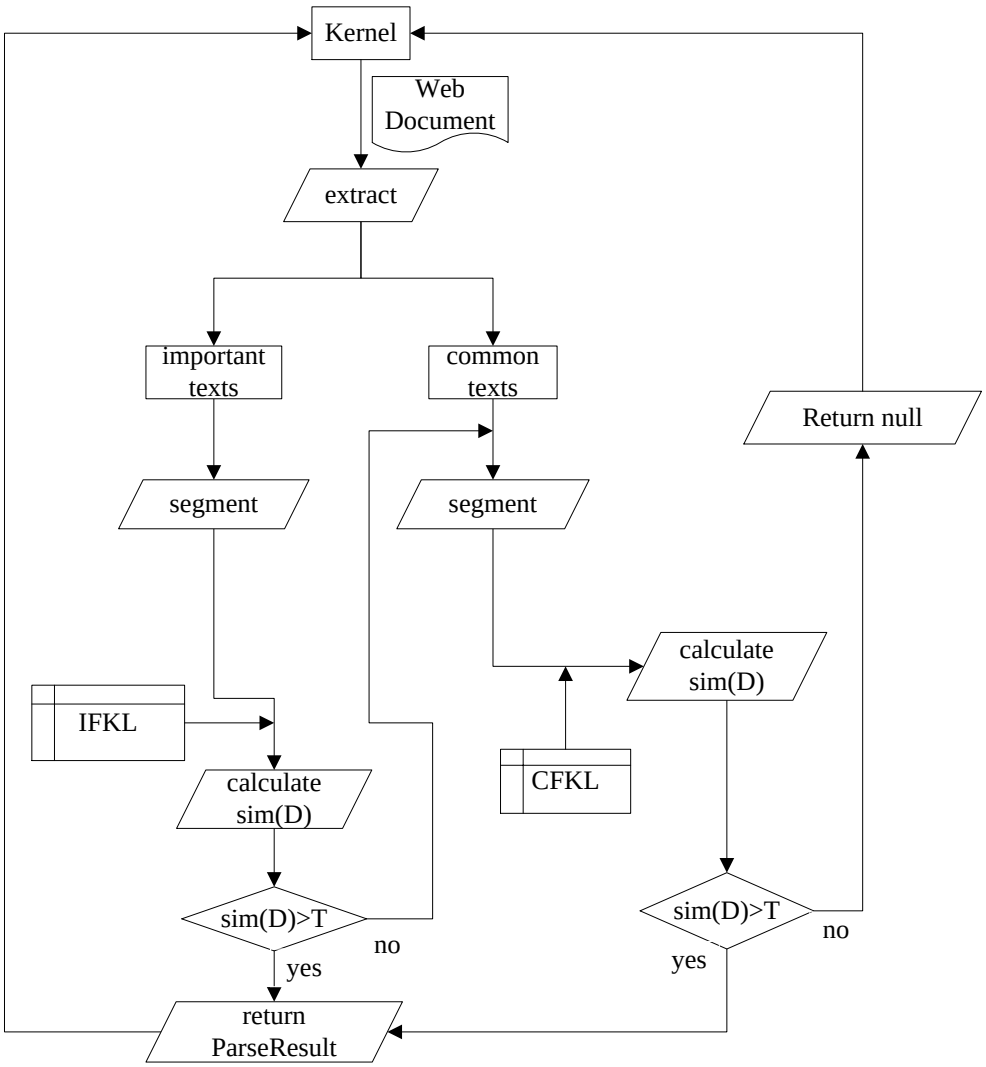


Figure 4. The Flow Chart of Focused Plug-in

5. Experiment

To prevent the focused crawler failed caused by the URL seeds which are focused unsuccessfully, the URL seeds that have high degree of topic similarity must be selected. So a number of authority financial domain web sites were selected as the initial URL seeds of focused crawler. Selected URL seeds are shown in Table 3 (data from the rank of the financial category websites at 2013 February 10th on *ChinaZ.com*).

Table 3. URL Seeds

Site	URL
jrj.com	http://www.jrj.com.cn/
Eastmoney	http://www.eastmoney.com/
Hexun	http://www.hexun.com/
Ifeng	http://finance.ifeng.com/

In order to evaluate the effects of focused crawler, this paper used Crawling Speed (CS) and Harvest Rate (HR) to measure the performance of focused crawler.

The crawling speed is defined as:

$$CS = \frac{NPRF}{CT} \quad (2)$$

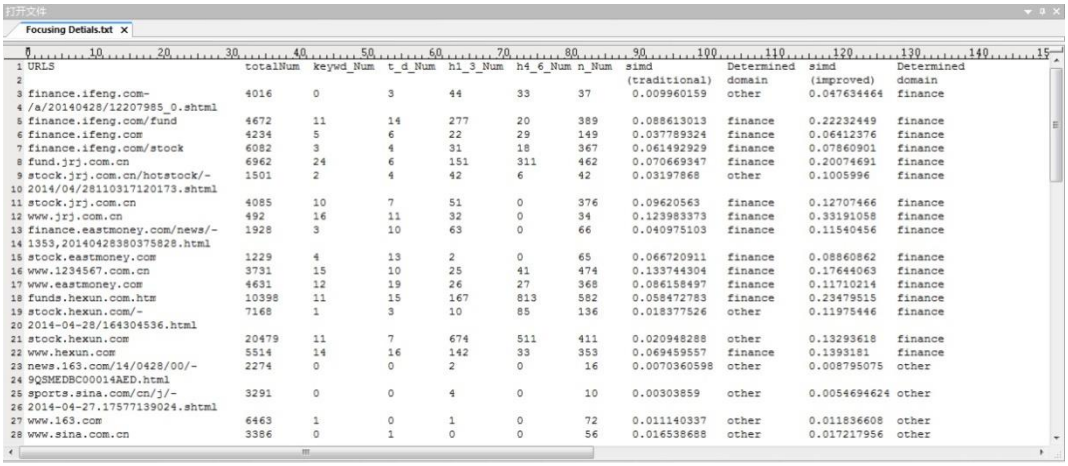
where NPRF is the number of web pages related to specific domain and CT is the crawling time.

The harvest rate is defined as:

$$HR = \frac{NPRF}{TNCP} \quad (3)$$

where NPRF is the number of web pages related to specific domain and TNCP is the total number of crawled pages.

This paper uses three computers whose configuration are intelpentium dual-core 1.6HZ processor and 1GB of main memory to build Hadoop environment and runs financial domain focused crawler which is based on Nutch plug-in mechanism and uses the improved algorithm proposed by this paper on the Hadoop environment. We also uses one computer whose configuration is Intel Core Duo 2.0HZ processor and 2GB of main memory to run the crawlers which uses the traditional algorithm and the improved algorithm as a comparison (we abbreviated as "Hadoop + Improved", "Single + Improved", "Single + Traditional"). The focusing details of some web pages used two algorithms are shown in Figure 5.



Focusing Detials.txt

	URLS	totalNum	keywd_Num	t_d_Num	h1_3_Num	h4_6_Num	n_Num	simd (traditional)	Determined domain	simd (improved)	Determined domain
3	finance.ifeng.com-/a/20140428/12207985_0.shtml	4016	0	3	44	33	37	0.009960159	other	0.047634464	finance
5	finance.ifeng.com/fund	4672	11	14	277	20	389	0.088613013	finance	0.22232449	finance
6	finance.ifeng.com	4234	5	6	22	29	149	0.037789324	finance	0.06412376	finance
7	finance.ifeng.com/stock	6082	3	4	31	18	367	0.061492929	finance	0.07860901	finance
8	fund.jrj.com.cn	6962	24	6	151	311	462	0.070669347	finance	0.20074691	finance
9	stock.jrj.com.cn/hotstock/-2014/04/28110317120173.shtml	1501	2	4	42	6	42	0.03197868	other	0.1005996	finance
11	stock.jrj.com.cn	4085	10	7	51	0	376	0.09620563	finance	0.12707466	finance
12	www.jrj.com.cn	492	16	11	32	0	34	0.123983373	finance	0.33191058	finance
13	finance.eastmoney.com/news/-1353,20140428380375828.html	1928	3	10	63	0	66	0.040975103	finance	0.11540456	finance
15	stock.eastmoney.com	1229	4	13	2	0	65	0.066720911	finance	0.08860862	finance
16	www.1234567.com.cn	3731	15	10	25	41	474	0.133744304	finance	0.17644063	finance
17	www.eastmoney.com	4631	12	19	26	27	368	0.086158497	finance	0.11710214	finance
18	funds.hexun.com.htm	10398	11	15	167	813	582	0.058472783	finance	0.23479515	finance
19	stock.hexun.com/-2014-04-28/164304536.html	7168	1	3	10	85	136	0.018377526	other	0.11975446	finance
21	stock.hexun.com	20479	11	7	674	511	411	0.020948288	other	0.13293618	finance
22	www.hexun.com	5514	14	16	142	33	353	0.069459557	finance	0.1393181	finance
23	news.163.com/14/0428/00/-9QBMEDBC00014AED.html	2274	0	0	2	0	16	0.0070360598	other	0.008795075	other
25	sports.sina.com/cn/j/-2014-04-27.17577139024.shtml	3291	0	0	4	0	10	0.00303859	other	0.0054694624	other
27	www.163.com	6463	1	0	1	0	72	0.011140337	other	0.011836608	other
28	www.sina.com.cn	3386	0	1	0	0	56	0.016538688	other	0.017217956	other

Figure 5. The Focusing Details of Some Web Pages of Two Algorithms

Crawling result and statistic information are shown in Table 4.

Table 4. Crawling Result and Statistic Information

crawler type	NPRF	TNCP	CT	HR(%)	CS(page/s)
Hadoop+Improved	14233	18045	27364s	78.88	0.52
Single+Improved	14072	17872	33110s	78.74	0.43
Single+Traditional	9604	15626	33883s	61.46	0.28

By comparing the focusing details of two algorithms, we find that the improved algorithms can distinguish the financial domain web pages and other domain web pages better and has higher accuracy than traditional algorithms.

From Table 4, we also find that the harvest rates of the improved domain focusing algorithm based on the web keywords raises about 17 percent than traditional,and the speed of the focused crawler running at Hadoop environment raises about 25 percent than single environment. Experimental results show that the crawler based on Nutch plug-in running at Hadoop environment is better than the focused crawler running at single environment.

6. Conclusions

This paper intent to investigate, design and implement a vertical search engine in the financial domain.Based onNutch plug-in mechanism, we provide and implement the details of a vertical search engine in the financial domain. To construct the finance domain focused crawler, the key component of the system, we develop an improved focusing algorithm based on our own financialdomain keywordslibrary. Experimental results show that, the finance domain focused crawler based onNutchplug-in mechanism has good performanceand can satisfypracticalrequirements of medium search application, and the system runs well.

Acknowledgments

This research was supported by Scientific Research Project of Higher Education of Inner Mongolia Autonomous Region, China (NJZY13052).

References

[1] L. Zhou and L.Lin, “Survey on the research of focused crawling technique”, Computer Applications, vol., 25, no. 9, **(2005)**,pp.1965-1969.
[2] J.Fang,“Research of Main Technologies of Vertical Search Engine”, MS thesis, Jinan University, **(2010)**.
[3] H. Zhan, Y. Yang and H.Fang,“Research and Optimization of Nutch Distributed Crawler”, Frontiers of Computer Science And Technology, vol. 5, no. 1,**(2005)**, pp.68-74.
[4] Apache,“Welcome to Apache Nutch”, http://nutch.apache.org/, **(2013)**.
[5] T.White,“Hadoop: The Definitive Guide. America”, O’Reilly Media, **(2009)**.
[6] T.Xia, “Analysis of Nutch’s Plug-in Mechanism”, Journal of Guangxi Normal University: Natural Science Edition, vol. 28, no. 1, **(2010)**, pp.105-108.
[7] X. Li and Y.Gu,“DOM-based Information Extraction for the Web Sources”, Chinese Journal of Computers, vol. 25, no. 5,**(2002)**, pp.526-533.

Authors

Yue Houguang, received the BS and MS degrees in computer science fromChina University of Petroleum in 1994 and 2002, and received the Ph.Din computer science from Graduate University of Chinese Academy of Sciences in 2005. Currently, his research interests include date mining, machine learning, and algorithms.

Zhang ling received her MSdegree in software engineering fromUniversity of Electronic Science and Technology of China in2014. Currently, her research interests includesoftware engineering, ming of Massive datasets.

MengFanjun received the BS and MS degrees in computer science from Inner Mongolia Normal University, China, in 1999 and 2007. Currently, his research interests include scheduling techniques and parallel algorithms for clusters, and also multi-core processors and software techniques for I/O-intensive applications.

Song Honghao, received the BS and MS degrees in computer science from Shandong University of Finance and Economics, China, in 2011 and 2014. Currently, his research interests include data and knowledge engineering, and financial information engineering.