ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 5, October 2014

Design and Development of a Domain Specific Focused Crawler Using Support Vector Learning Strategy

M. Selvakumar, Dr. A.Vijaya

Research scholar, Department of Computer Science, Periyar University, Salem, India

Assistant Professor, Department of Computer Applications, Government Arts College, Salem, India

ABSTRACT: The usage of internet is immense on a large scale for the past many years. Especially more than 90% of people are using search engine. People are using search engine largely for their keywords. Search engine results gives irrelevant data too. We should divide it an focused crawler. The relevant web pages, we have to enter our keywords Focused crawler using SVM. The SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes. Margin means the maximal width of the slab parallel to the hyperplane that has no interior data points. Moreover, we get SVM minute is used more and more, on the other hand, we get relevant web pages. The difficulty of bounding the offset is overcome. Naïve bayes and SVM are used to demonstrate the main results.

KEYWORDS*:* Search Engin, Focused Crawler,SVM, Naïve bayes.

I. INTRODUCTION

A web search engine is a software system that is designed to search for information on the World Wide Web. The search results are generally presented in a line of results often referred to as search engine results pages (SERPs). The information may be a specialist in web pages, images, information and other types of files. Some search engines also mine data available in databases or open directories. Unlike web directories, which are maintained only by human editors, search engines also maintain real-time information by running an algorithm on a web crawler. A web crawler (also known as a robot or a spider) is a system, a program that traverses the web for the purpose of bulk downloading of web pages in an automated manner[1]. A web crawler is a program that, given one or more seed URLs, downloads the web pages associated with these URLs, extracts any hyperlinks contained in them, and recursively continues to download the web pages identified by these hyperlinks. Web crawlers are an important component of web search engines, where they are used to collect the corpus of web pages indexed by the search engine[2]. The types of crawler includes horizontal (BFS), vertical (DFS), periodic, parallel, topical, domain specific, social network, semantic, and dynamic. Challenges of web crawler are web coverage, dynamic web, hidden web, deep web and freshness.

II. DESIGN ISSUES OF FOCUSED CRAWLER

The crawlers can be classified in to two types based upon the application. General Crawler and Focused (Topical) Crawler. The General Crawler serves as an entry point to web pages. It strives for coverage that is as broad as possible, where as the Focused Crawler is built to retrieve the pages within a certain topic. In this case, the task of crawling could be constrained by programmer[12].A focused crawler or topical crawler is a web crawler that attempts to download only web pages that are relevant to a pre-defined topic or set of topics.

2.1 Definition

A focused crawler may be described as a crawler which returns relevant web pages on a given topic in traversing the web. It takes as input one or several related web pages and attempts to find similar pages on the web, typically by recursively following links in a best first manner. Ideally, the focused crawler should retrieve all similar pages while retrieving the fewest possible number of irrelevant documents. The goal of a focused crawler is to selectively seek out

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 5, October 2014

pages that are relevant to a pre-defined set of topics. The topics are specified not using keywords, but using exemplary documents. Rather than collecting and indexing all accessible web documents to be able to answer all possible queries, a focused crawler analyzes its crawl boundary to find the links that are likely to be most relevant for the crawl, and avoids irrelevant regions of the web[14].

2.2 General Architecture

A focused crawler has the following main components: (a) A way to determine if a particular web page is relevant to the given topic, and (b) a way to determine how to proceed from a known set of pages. An early search engine which deployed the focused crawling strategy was proposed in [1] based on the intuition that relevant pages often contain relevant links. It searches deeper when relevant pages are found, and stops searching at pages not as relevant to the topic. Unfortunately, the above crawlers show an important drawback when the pages about a topic are not directly connected in which case the crawling might stop pre- maturely. A topical crawler ideally would like to download only web pages that are relevant to a particular topic and avoid downloading all others. Therefore a topical crawler probability that a link to a particular page is relevant before actually downloading the page. A possible predictor is the anchor text of links; this was the approach taken by Pinkerton [4] in a crawler developed in the early days of the Web. In a review of topical crawling algorithms, Menczer et al. [5] show that such simple strategies are very effective for short crawls, propose to use the complete content of the pages already visited to infer the similarity between the driving query and the pages that have not been visited yet. Guan et al[8].

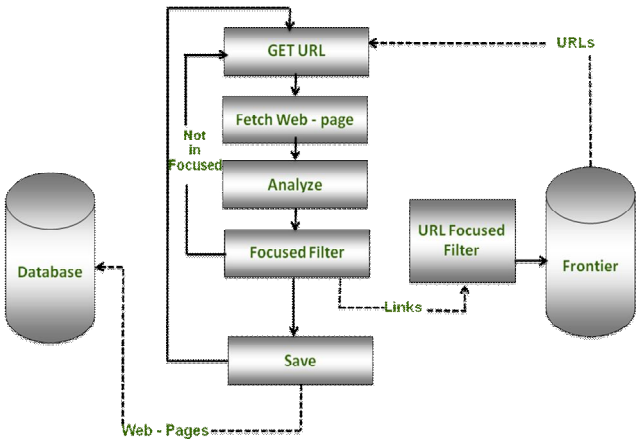

Figure 2.1: Architecture of Focused Crawler.

In Fig 2.1, its implementation just saves the web-pages fits with the topic dentition. The web-pages are always saved, it will be the algorithm that decides which out-links or not will go to the queue and with which score. The scheduling is modified in order to have a shorter queue, before it takes all the ready pages. Out-links have to be scored. When the Crawler starts to run, it has to be differen tiated the normal Crawler with the seeds (not scoring) from the score Crawler where the queue is filled with the greatest-scored pages. A new initialization was needed. The analysis had to be changed. Some other small changes were necessary to manage the databases[15]. The length of the queue is an important issue: On the one hand if it is too short, the crawler has to stop too often to recalculate/fill the queue and it is completely stopped (the frontier has to be sorted). On the other hand, if it is too long it stores many pages with low score that may not be crawled otherwise. In this light the value has to be selected carefully, in both [17] and [10] test beds were done.

III .LITERATURE SURVEY

The term focused crawler was first introduced by Chakrabarti and their colleagues [9]. They described the focused crawler in which a crawler seeks, acquires, indexes, and maintains pages on a specific set of topics that represent a relatively narrow segment of the web [9]. The focused web crawlers are designed for retrieving web pages based on the rules that identify relevant pages or/and priority criterions to sequence the web pages to be crawled and add them to

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 5, October 2014

the local database [1]. Focused crawlers are not designed only for downloading documents to be indexed for a domain specific search engine. But they also are designed to downloaded documents to use as a source for data mining [10].

Focused crawling is a hopeful approach for improving the precision and recall of expert search on the Web. As said before, the focused crawling motivation comes from the poor performance of general purpose search engines, which depend on the results of generic web crawlers [8]. The focused crawlers aim to search and retrieve Web pages from the World Wide Web, which are related to a specific domain. Instead of visiting all Web pages, a focused crawler visits only the region of the web that contains relevant pages, trying to skip irrelevant regions. This leads to significant savings in both computation and communication resources [2].

The way focused crawlers exploit hyper-textual information is one of the features that characterize them. Traditional crawlers convert a Web page into plain text extracting the contained links, which will be used to crawl other pages. Focused crawlers exploit additional information from Web pages, such as anchors or text surrounding the links. This information is used to predict the benefit of downloading a given page [11]. Essential issues of focused crawler is how to identify links and pages that are relevant to the specific domain, and to order the URLs in the URL queue [6]. So, a successful focused crawler has to predict precisely a web pages relevance before downloading it [12]. Early focused crawlers relay on using the domain keywords to determine if the page is relevant or not after downloading it, like [9]. Trying to enhance the focused crawler, some use ontology to detect the relevant score for links before downloading. They order the previously download documents using ontology by computing the page relevance [2] [13] [14] and [15]. Pahal and his colleagues [16] present a focused crawling that uses the concept with its context and context information for downloading web documents. Filter the downloaded document is still used, Luong and his colleagues [15] use the concepts to crawl web documents. After that they filter the retrieved documents by applying SVM classification.

Some researches analysis the structure hyperlinks to evaluate the relevance. Huang and his colleges [14] present a focused crawler approach that evaluates the pages content relevance using ontology and hyperlinks analysis. Jamali and his colleges [8] use the link structure analysis with the similarity of the page context to determine the download pages priority. while Xu and Zuo [12] use the hyperlinks to discover the relationships between the web pages. We can categorize the focused crawler approaches according to their dependency on determining the relevant pages to: ontology based focused crawler, structure based focused crawler, and others focused crawler approaches.Structure base focused crawlers take in accounting the web pages structure when evaluating the page relevance. Jamali and his colleges [8] and Huang and his colleges [14] analysis the hyperlinks between the candidate crawled page and the domain web pages and to determine if it relevant to the domain or not. Others use all HTML elements to determine the relevant of the web pages [12] [5]. Xu and Zuo [12] crawl the web using rules which learned from the structure of the relevant pages. Patel and Schmidt [5] use the anchor text html structure to order the candidate crawled page.

Bazarganigilan and his colleges [18] present a focused crawler that use similarity function to determine the page relevant. They use genetic programming to discover the best combination for estimation the similarity evaluation among pages. Their crawler download the web pages pointed to by the starting URLs. For each downloaded web page, the similarity function will be used by a classifier to determine if this is a computing-related Web page. If yes, this web page will save into the download collection. The outgoing links of the download relevant web pages will be collected and put into the crawling queue. They apply a decay concept to each page. The page would stop of crawling if it does not comply with predefined threshold. Zhang and Lu [19] use Q learning with semi-supervised learning to select the most topic relevant URL to crawl based on the scores of the URLs in the unvisited list. They calculate these scores based on the fuzzy class memberships and the Q values of the unlabelled URLs. As others, they use Keywords and selected relevant web pages to describe the semantics of a topic and to guide the first crawling according to the Q values of the seed URLs. As more relevant web pages have been crawled, they update the topic keywords by modifying the weights of word update the occurrence frequencies of the word features and change the word features according to their occurrence frequencies. The hub and authority score of a web page is calculated using information of link structure among web pages. The set of the download web pages changes online and the immediate action reward should also be updated online. After a hyperlink has been traversed, its corresponding document is classified into different classes with fuzzy memberships. The unvisited hyperlinks list will be evaluated again and ranked according to their Q values. The average Q value of a class is re-calculated based on the topic relevance of the newly crawled web page, and this process continues as the crawl progresses.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 5, October 2014

IV .PROPOSED SVM BASED FOCUSED CRAWLER

Support Vector Machines is a statistical based learning algorithm [20]. This algorithm addresses the general problem of learning to discriminate between positive and negative members of a given class of n dimensional vectors. The SVM need both positive and negative training set which are uncommon for other classification methods. The performance of the SVM classification remains unchanged even if documents that do not belong to the support vectors are removed from the set of training data; this is one of its major advantages. Merits[21][22] as it is able to manage large spaces of features and high generalization ability. Demerits: But this makes SVM algorithm relatively more complex which in turn demands high time and memory consumptions during training stage and classification stage. Support Vector Machines Classifier is better than another if it generalizes better, i.e. shows better performance on documents outside of the training set. It turns out that the generalization quality of the plane is related to the distance between the plane and the data points that lay on the boundary of the two data classes. These data points are called "support vectors" and the SVM algorithm determines the plane that is as far from all support vectors as possible. In other words, SVM finds the separator with a maximum margin and is often called a "maximum margin classifier. In particular the linear SVM represents a state –of-the- art method for text classification. Given a set of labeled data $D=\{(x_1y_1), (x_2y_2),\ldots,(x_m,y_m)\}$, where $x_i \in X$ and $y_i \in \{-1,+1\}$, a SVM is represented by a hyper plane

$$f(x) = (\sum_{i=1}^{m} \alpha_i K(x_i, x)) + b = 0$$

Where $K(u,v)$ is a kernel function satisfying Mercer's condition. The hyperplane defined above can be interpreted as a decision boundary and thus the sign f(x) of gives the predicated label of input x. For the following discussion it is important to note examples far away from the decision boundary can be classified with a high confidence while the correct classes for examples close to the hyperplane or within the margin are uncertain. In Fig2.2.H_1 does not separate the classes. H_2 does, but only with a small margin. Fig 2.3.H_3 separates them with the maximum margin Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

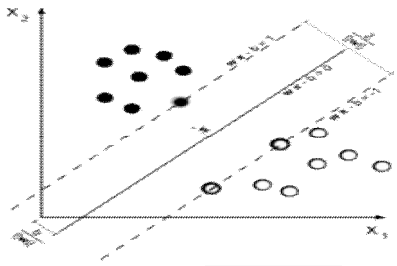

Figure 2.2: SVM Small Margin.

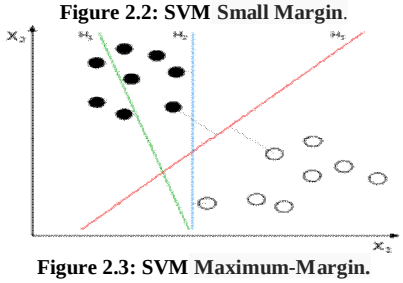

Figure 2.3: SVM Maximum-Margin.

4.1 ARCHITECTURE

A main idea behind any crawler design Fig 2.4, its implementation just saves the web-pages fits with the topic dentition. The web-pages are always saved, it will be the algorithm that decides which out-links or not will go to the queue and with which score. The scheduling is modified in order to have a shorter queue, before it takes all the ready pages. Out-links have to be scored. When the Crawler starts to run, it has to be differen tiated the normal Crawler with the seeds (not scoring) from the score Crawler where the queue is filled with the greatest-scored pages. A new initialization was needed. The analysis had to be changed. Some other small changes were necessary to manage the databasesis to leave the actual Spider doing as little processing as possible, thus leaving it free to download documents

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 5, October 2014

faster. Feature vectors are equivalent to the vectors of explanatory variables used in statistical procedures such as linear regression. Feature vectors are often combined with weights using a dot product in order to construct a linear predictor function that is used to determine a score for making a prediction. The concepts of support vectors, kernels and slack variables can be easily adapted in identify Facture Vector . Most importantly, all the parameters we need to estimate for identify facture vector are outside of the kernel functions, ensuring the convexity of the solution space, which is the same as in SVM. a feature vector is an n-dimensional vector of numerical features that represent some object. when representing texts perhaps to term occurrence frequencies.

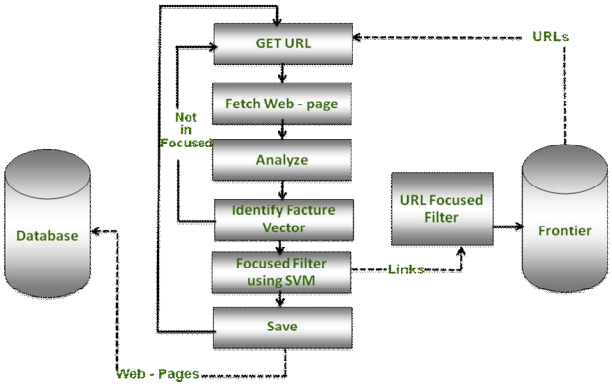

Figure 2.4: Architecture of Focused Crawler using SVM.

4.2 DESIGN AND DEVELOPMENT

Simple linear classifiers, being unable to deal with non-linearly separable data or noisy data, are not always sufficient data classifiers. If necessary, the more complex Support Vector Machine (SVM) learning algorithms are there to fill the gap. Support Vector Machines, though exceedingly complex to implement, offer a solution to the above limitations; by mapping data into a richer feature space including non-linear features, it is then possible to classify the data in a simple linear fashion, something that seemedim possible just moments before. Support Vector Machines are based on Mercer's theorem, which states that any continuous, symmetric, positive semi-definite kernel function can be expressed as a dot product in a high-dimensional space. As a side effect of this theorem, the resulting problems have no local minima, meaning that all have the property of convexity. If a machine learning algorithm can be rewritten into any higher dimensional space so that they so that the mathematical computations it uses are based solely on inner (dot) products between the data features, a Support Vector Machine can be created by replacing every dot-product with the chosen SVM kernel, a computational short cut called the Kernel Trick.

The higher-dimensional non-linear algorithm is equivalent to the original lower dimension linear algorithm; the new higher-dimensional representation is simply operating in a feature space mapped from the original. Because of this kernel trick, the new feature space function - which has the potential of being highly complex - is never explicitly computed. This is highly desirable, as it makes this calculation solvable in merely polynomial - rather than exponential - computation time. Merely finding one hyperplane that separates the training data is not enough to result in an accurate learning machine; many such hyper planes will exist, and - as it is extremely easy to overfit in high dimensional spaces - merely using any random hyper plane will likely result in poor accuracy against the test data set. In order to choose the best possible hyper plane and minimize the risk of overfitting, it is necessary to find the one with the maximal margin between the classes of items being identified. The transformation from Primal to Dual form is therefore done by means of a Lagrangian, whose constraints each place an upper bound on the linear combination of the Lagrangian variables and thus limit the amount it is possible to fit the training data within the hyperspace created by that Lagrangian. Additionally, if the kernel and it's parameters were chosen carefully, the margin will be larger and better generalization can be achieved. Once this maximum margin is found, only the points nearest to the hyperplane will be given a positive weight, resulting in sparsely weighted features within the within the higher-dimensional feature space. These points are rather appropriately titled support vectors.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 5, October 2014

V. RESULT & DISCUSSION

we will like to give brief description of machine learning techniques used by us (SVM and NB), for our experiments evaluation. The Support Vector Machines is a classifier that finds best hyper plane between two classes of data, by separating positive and negative examples through solid line in the middle called decision line. In following figure gap between solid and dashed line reflects the margin of movement of decision line left or right without miss-classification of document. Naive Bayes classifier is basically a probabilistic classifier based on hypothesis. On the basis of assumption and training document; Bayesian learning is to find most appropriate assumption based on prior hypothesis and initial knowledge. Main assumption is that terms in test document have no relation among them and probability is calculated that document belong to category. it was found that the Basic Naive Bayes classification algorithm gave the best prediction results on our testing set, coming in at a respectable 97.8%. However, assuming accurate parameter tuning (when relevant), all Support Vector Machines and Naive Bayes algorithms that were designed to utilize binary imput data resulted in an accuracy well above the 90% range. As mentioned when the Naive Bayes algorithms were first described above, the Multivariate Gauss Naive Bayes algorithm is not designed to work with binary data it instead expected counts of the occurrences of each feature in the email. It's far lower accuracy of 70%, while still far better than random guessing, would likely improve greatly if it received data of the correct form.

Table.1.Naive Bayes Accuracy

NB Algorithm	Accuracy #	Correct in Testing Set
Basic Naive Bayes	97.8%	489/500
Multinomial Naive Bayes	95.8%	479/500
Multivariate Naive Bayes	93.8%	469/500
Multivariate Gauss Naïve Bayes	70.2%	351/500

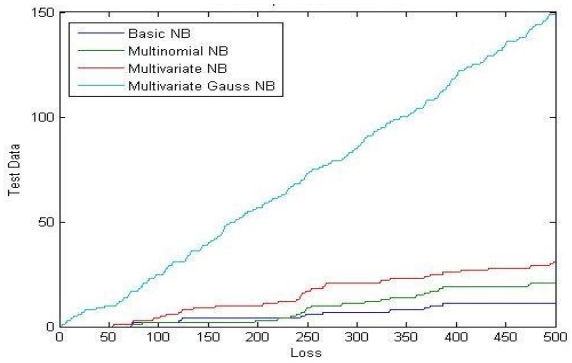

Fig 3.1.Naive Bayes Comparison of Test Loss

The Support Vector Machine is a classifier that finds best hyper plane between two classes of data, by separating positive and negative examples through solid line in the middle called decision line. In following figure gap between solid and dashed line reflects the margin of movement of decision line left or right without miss-classification of document. Support Vector Machines classifier is basically a probabilistic classifier based on hypothesis. On the basis of assumption and training document; Support Vector Machine learning is to find most appropriate assumption based on prior hypothesis and initial knowledge. Main assumption is that terms in test document have no relation among them and probability is calculated that document belong to category. it was found that the Support Vector Machine classification gave the best prediction results on our testing set, coming in at a respectable 98.6%. However, assuming accurate parameter tuning (when relevant), all Support Vector Machine that were designed to utilize binary input data resulted in an accuracy well above the 90% range. As mentioned when the Support Vector Machine are classifiers

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 5, October 2014

because they usually achieve good error rates and can handle unusual types of data. It's far lower accuracy of 80.4%, while still far better than Naïve Bayes, would likely improve greatly if it received data of the correct form web pages

Table.2.Support Vector Machine Accuracy

Algorithm	Accuracy #	Correct in Testing Set
SVM (High+)	98.6%	493/500
SVM (Low-)	80.4%	402/500

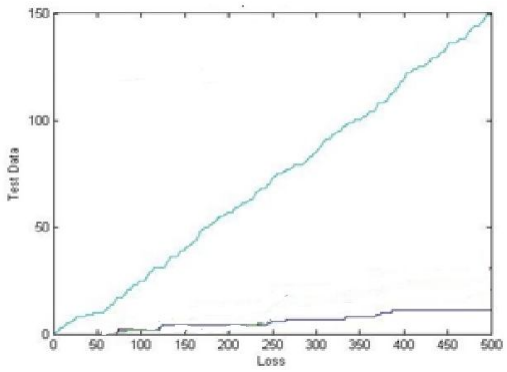

Fig 3.2 Support Vector Machine Comparison of Test Loss

VI. CONCLUSION

This result in the discovery of some high-quality information resources that might have otherwise been overlooked. The focused crawler also gives better user experience as it tries to give results which are more relevant to user's information needs, thus leading to user satisfaction which is one of the parameters measuring success of Focused Crawler System. The quantity of data on web is increasing exponentially so it is critical for Focused Crawler system to be very efficient while handling user queries which normally require high precision, thus focused crawler plays a crucial role over here. We showed that SVM-based focused crawler with feature selection is more suitable for our approach. The SVM focused crawler performed better than the Naïve Bayes classifier, based on the evaluation results shown above, and better than the baseline approach. Focused Crawler was used as a pre-processor for dimensionality reduction followed by the SVM method for text classification. There is a need to experiment with more such hybrid techniques in order to derive the maximum benefits from machine learning algorithms and to achieve better classification results.

REFERENCES

[1] A. Thukral, V. Mendiratta, A. Behl, H. Banati and P. "Bedi, FCHC: A Social Semantic Focused Crawler", in Communications in Computer and Information Science, Vol. 191, Part 5, pp. 273-283, 2011.
[2] M. Kumar and R. Vig, "Design of CORE: context ontology rule enhanced focused web crawler", International Conference on Advances in Computing, Communication and Control (ICAC3˝09) pp. 494-497, 2009.
[3] A. Chandramouli, S. Gauch, and J. Eno, "A Cooperative Approach to Web Crawler URL Ordering", iIn Human Computer Systems Interaction, AISC 98, Part I, pp. 343–357, 2012
[4] P. Gupta, A. Sharma, J. P. Gupta, and K. Bhatia, „A Novel Framework for Context Based Distributed Focused Crawler (CBDFC), Int. J.CCT, Vol. 1 , No. 1 , pp.13-26. 2009
[5] A. Patel, and N. Schmid, "Application of structured document parsing to focused web crawling", in Computer Standards & Interfaces 33 (2011) .
[6] A. Pirkola and T. Talvensaari, "Effects of Start URLs in Focused Web Crawling", in INFORUM 2009: 15th Conference on Professional Information Resources Prague, May 27-29, 2009.
[7] S. Yang and C. Hsu, "An Ontology-Supported Web Focused- Crawler for Java Programs", Proc. of 2010 International Workshop on Mobile Systems, E-commerce, and Agent Technology, Jinhua, China, Jul. 5-6, 2010.
[8] M. Jamali , H. Sayyadi , B. B. Hariri, and H. Abolhassani, "A Method for Focused Crawling Using Combination of Link Structure and Content Similarity", Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp.753-756, December 18-22, 2006.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol.2, Special Issue 5, October 2014

[9] S. Chakrabarti, M. v. d. Berg, and B. Domc, "Focused crawling: a new approach to topic-specific Web resource discovery", Computer Networks, 31(11–16):1623–1640. 1999 [10] A. Pirkola, "Focused Crawling: A Means to Acquire Biological Data from the Web", in VLDB '07, September 23-28, 2007.
[11] A. Micarelli and F. Gasparetti, "Adaptive Focused Crawling", in The Adaptive Web, LNCS 4321, pp. 231–262, 2007.
[12] Q. Xu and W. Zuo, "First-order Focused Crawling", pp. 1159-1160,WWW 2007.
[13] C. Su, Y. Gao, J. Yang, and B. Luo, "An efficient adaptive focused crawler based on ontology learning", Hybrid Intelligent Systems, 2005. HIS apos;05. Fifth International Conference on 6-9 Nov. 2005.
[14] W. Huang, L. Zhang, J. Zhang, M. Zhu, "Focused Crawling for Retrieving E-commerce Information Based on Learnable Ontology and Link Prediction" ieec, International Symposium on Information Engineering and Electronic Commerce, pp.574- 579, 2009.
[15] H. P. Luong, S. Gauch, and Q. Wang, "Ontology-Based Focused Crawling", Information, Process, and Knowledge Management, 2009 (eKNOW '09) ,pp. 123-128 1-7 Feb. 2009. [16] N. Pahal, N. Chauhan, and A.K. Sharma, "Context-Ontology Driven Focused Crawling of Web Documents", A.K. Wireless Communication and Sensor Networks, 2007. WCSN apos:07. Third International Conference, pp.121-124 , 13-15 Dec. 2007.
[17] H. Dong, F. K. Hussain, and E. Chang, "State of the art in semantic focused crawlers" in 2009 IEEE International Conference on Industrial Technology (ICIT 2009),
[18] M. Bazarganigilani, A. Syed and S. Burki, "Focused web crawling using decay concept and genetic programming", In International Journal of Data Mining & Knowledge Management Process (IJDKP) pp:1-12, 2011,Vol.1,.2010
[19] H. Zhang and J. Lu, "SCTWC: An online semi-supervised clustering approach to topical web crawlers", in Applied Soft ComputingVol. 10, No. 2, pp. 490-495, 2010.
[20] A.Khan,B.Baharudin,Lan Hong Lee. A Review of Machine Learning Algorithms for Text- Documents Classification. Journal Of Advances in Information Technology, Vol. 1 , No. 1, Feb.2010.
[21] Z. Wang, X. Sun, D. Zhang. An optimal Text categorization algorithm based on SVM.
[22] Automatic Text Classification: A Technical Review International Journal of Computer Applications (0975 – 8887) Volume 28– No.2, August 2011 Mita K. Dalal Sarvajanik College of Engineering & Technology, Surat, India, Mukesh A. Zaveri Sardar Vallabhbhai National Institute of Technology, Surat, India.

BIOGRAPHY

M. SELVAKUMAR, is currently doing Ph.D, in Computer Science, in Periyar University Salem-11. He has 5 years of teaching experience. He has published several research publications at National, International conferences and journals. His areas of interests are Networking, web mining, search engines, web communities, social network mining.

Dr. A. VIJAYA KATHIRAVAN is working as an Assistant Professor in Computer Applications in PG and Research Department of Computer Science, Govt. Arts College (Autonomous), Salem-07, TamilNadu, INDIA. She received her M.Phil. in Computer Science from Bharathiar University, Coimbatore and she awarded her doctoral degree in Computer Applications from University of Madras, Chennai. She has published 6 Books, 3 papers in National Journal, 30 papers in International Journal, 35 Papers in National Conference Proceedings and 27 Papers in International Conference Proceedings. Her research interests include data structures and algorithms, data/text/web mining, search engines, web communities, social network mining, machine learning, Natural Language Processing, Organizational leadership and human resource management.