DISTRIBUTED APPROACH to WEB PAGE CATEGORIZATION USING MAP-REDUCE PROGRAMMING MODEL

P.Malarvizhi [#1] , Ramachandra V.Pujeri [*2]

# Research Scholor, Karpagam university , Coimbatore, Tamilnadu
* Vice Principal, KGISL Institute of Technoloy, Coimbatore, Tamilnadu
1 malarvizhi_s@yahoo.co.in
2 sriramu_psg@yahoo.com

ABSTRACT

The web is a large repository of information and to facilitate the search and retrieval of pages from it, categorization of web documents is essential. An effective means to handle the complexity of information retrieval from the internet is through automatic classification of web pages. Although lots of automatic classification algorithms and systems have been presented, most of the existing approaches are computationally challenging. In order to overcome this challenge, we have proposed a parallel algorithm, known as MapReduce programming model to automatically categorize the web pages. This approach incorporates three concepts. They are web crawler, MapReduce programming model and the proposed web page categorization approach. Initially, we have utilized web crawler to mine the World Wide Web and the crawled web pages are then directly given as input to the MapReduce programming model. Here the MapReduce programming model adapted to our proposed web page categorization approach finds the appropriate category of the web page according to its content. The experimental results show that our proposed parallel web page categorization approach achieves satisfactory results in finding the right category for any given web page.

Keywords: World Wide Web, Web page categorization, Web crawler, MapReduce programming model, Relevancy measure.

I. INTRODUCTION

A wide area network (WAN) and a client server protocol are employed by the information environment WWW which consists of a huge distributed database of heterogeneous documents. This environment has a graph structure, where nodes (web pages) are joined by edges (hyperlinks). Navigating cross documents through hyperlinks, retrieving the information of interest along the way is the commonly adopted way to access the information found on the WWW [1]. Categorizing the web documents is necessary for the Web which is a huge repository of information to facilitate the indexing, searching and retrieval of pages [2]. The common first step of mining the Web namely, categorizing the Web pages of an exciting class makes Web page categorization/classification as one of the vital techniques for Web mining [3]. The mining of interesting and potentially valuable patterns and implicit information from artifacts or activity related to the World Wide Web is known as Web mining [4]. The process of assigning a web page to one or several predefined category labels is known as Web page classification or web page categorization [5].

The general problem of web page classification can be categorized into multiple sub-problems, subject classification, functional classification, sentiment classification, and other types of classification. For instance, functional classification is the one which chooses a page to be a “personal homepage”, “course page” or “admission page”. The opinion that is presented in a web page is focused by sentiment classification, that is, the author’s approach regarding some specific topic. Other types of classification comprise the genre classification [6], search engine spam classification (e.g., [7], [8]) and so on [5]. Several people employ web page categorization techniques that categorize the data that are retrieved and extracted from the Web content on the basis of keyword categorization. More computation task is required for this type of web page categorization [9], because they process large amount of keywords at a time. The best choice for speeding up the process of web page categorization is parallel computing. Parallel computing, is a form of computation in which many calculations are carried out at the same time.[10] It operates on the principle that huge problems can be usually split up into smaller ones that can be resolved concomitantly ("in parallel").

Various programming models are available for simulating the parallel computing process. Map-Reduce programming model is one of the widely accepted programming paradigms on data-center-scale computer systems [11], [12] and such data analysis applications can be successfully supported by the MapReduce framework [13] popularized by Google which is very attractive specifically for parallel processing of arbitrary data**.** MapReduce creates smaller tasks that run in parallel on multiple machines by dividing a computational task and scales easily to huge clusters of low-cost commodity computers [14]. MapReduce has become a popular means for harnessing the power of large clusters of computers. MapReduce permits programmers to think in a data-centric fashion: they allow the details of distributed execution, network communication, coordination and fault tolerance to be managed by the MapReduce framework and concentrate on applying transformations to sets of data records [15]. To form processing primitives, MapReduce provides an abstraction for programmer-defined “mappers" (that specify the per-record computation) and “reducers" (that specify the result aggregation) and both operate in parallel on key-value pairs by taking inspiration from higher-order functions in functional programming. An arbitrary number of intermediate key-value pairs are generated by applying the mapper to every input key-value pair. An arbitrary number of final key-value pairs are generated as output by applying the reducer to all values associated with the same intermediate key [16].

In this paper, we have proposed an efficient approach for web page categorization. The proposed approach is designed based on the parallel computing MapReduce programming model. At first, the web pages are mined by the crawler from the web. Then, the crawled web pages are given to the MapReduce framework which is a programming model used by Google for performing distributed computation. MapReduce programming model contains two important operations such as, Map function and Reduce function. In Map function, the web pages are mapped into key value pairs, where key refers to the keyword and value refers to the frequency of the keyword. In reduce function, the relevancy measure is calculated based on the frequency of the web pages. The relevancy measure is designed based on the frequency of keyword and weights associated with the predefined domain keywords. Finally, the relevancy measures computed for all reduce functions are combined and from it the appropriate category of the web page is identified.

The structure of the paper is organized as follows: A brief review of the related research is given in Section II. The basic concepts related to the proposed approach are described in section III. The proposed approach for web page categorization is given in Section IV. The experimental results of the proposed approach are presented in Section V. Finally, the conclusions are given in Section VI.

II. RELATED RESEARCH

Numerous works have been carried out in presenting MapReduce as an effective programming model for parallel computing systems, thus making the process of tedious computations to look simple. In the proposed approach, we make use of MapReduce programming model for web page categorization. Here, we present some of the works that are related to MapReduce programming model and web page categorization.

Jost Berthold *et al*. [36] have presented two parallel implementations for Google map-reduce skeleton one consistent with the previous work, and the other optimized version, in the parallel Haskell extension Eden. The efficient execution of the complex coordination structure of that skeleton has been supported by Eden’s precise characteristics, like lazy stream processing, dynamic reply channels, and nondeterministic stream merging. They have delivered runtime analyses for example applications by comparing the usage and performance of the Google map-reduce skeleton implementations. Though supple, the Google map-reduce skeleton is generally too common, and a better runtime behavior has been revealed by typical examples employing alternative skeletons. Petr Krajca and Vilem Vychodil [27] have used the map-reduce approach to data processing in the scalable distributed algorithm they have introduced for computing maximal rectangles. Exploring interesting patterns in binary matrices play a vital role in data mining, mainly, in formal concept analysis and related disciplines. Their approach has overcome the computational complexity which is the major drawback of several algorithms presented for computing specific patterns represented by maximal rectangles in binary matrices that limits their applicability to comparatively small datasets.

Jeffrey Dean and Sanjay Ghemawat [37] have described an implementation of the MapReduce interface tailored towards their cluster-based computing environment. MapReduce is a programming model and an associated implementation which can be applied to a broad variety of real world tasks for processing and generating large datasets. Once the computation is specified by the users in terms of a map and a reduce function, the computation across large-scale clusters of machines are automatically parallelized by their underlined runtime system, which also makes competent use of the network and disks by handling machine failures, and scheduled inter-machine communication. Programmers have found the system easy to use: over the past four years more than ten thousand distinctive MapReduce programs have been executed internally at Google,

and every day an average of one lakh MapReduce jobs have been implemented on Google's clusters, processing over twenty petabytes of data each day.

Tamer Elsayed *et al.* [38] have proposed a MapReduce algorithm for computing pairwise document similarity in large document collections. MapReduce is an appealing framework since it enables the separation of the inner products engaged in computing document resemblance into separate multiplication and summation stages in such a way to well compete with the competent disk access patterns across quite a few machines**.** Their algorithm has shown linear growth in running time and space in terms of the number of documents on a collection comprising of roughly 900,000 newswire articles. Nuanwan Soonthornphisaj and Boonserm Kijsirikul [39] have presented an approach called Iterative Cross-Training (ICT). Web page categorization has been done on three data sets by applying their algorithm. Classifying Web documents into a definite number of predefined categories is the goal of Web page categorization. The supervised learning algorithms, Co-Training and Expectation Maximization have been used to assess and analyze the functioning of ICT. They have discovered ICT as an effectual approach for the Web page categorization task.

Jebari Chaker and Ounelli Habib [40] have presented a supple approach for document genre categorization. A combination of contextual and structural classifiers which are homogenous classifiers has been used as the basis for the proposed approach. While the URL has been used by the contextual classifier, the structural classifier has utilized the document structure. Contextual and structural classifiers are both centroid-based classifiers. Compared to other categorization approaches, a superior micro-averaged break- even point (BEP) of more than 85% has been obtained by the proposed system in the experimental results.

Jane E. Mason *et al.* [41] have presented part of a larger project on genre based classification of Web pages. Genre based classification has been a powerful tool for filtering online searches. Two sets of experiments have been described by them for examining the automatic classification of Web pages by their genres. In these experiments, their approach has used profiles composed of fixed-length byte n-grams to represent the Web pages. In their study, the influence of the three feature selection measures on the preciseness of Web page categorization has been examined by the first set of experiments whereas a comparison of the classification accuracy of the three classification methods which employ n-gram representations of the Web pages has been made by the second set of experiments.

III. PRELIMINARIES

A. Web Crawling

In making the Web simpler to use for millions of people, a fundamental role is played by the first complete full-text search engine for World-Wide Web known as WebCrawler [17], [18]. The process by which the WebCrawler collects pages from the Web is known as Crawling. Collection of Web pages at a central location is the end result of crawling. WebCrawler begins with a single URL, downloads that page, retrieves the links from that page to others, and the process is repeated with each of those pages [18]. The graph structure of the Web is used by programs called Web crawlers to move from one page to another and to download them. Words that are quite suggestive of Web imagery such as wanderers, robots, spiders, fish, and worms are used to name **s**uch programs in their infancy [19]. Automated software agents (called crawlers) used by crawler based search engines, visit a Web site, read the information on the actual site, read the meta tags of the site and also does indexing on all linked Web sites by following the links that are connected by the site. All that information is returned to a central warehouse by the crawler, which the data is indexed. To examine whether any of the information has changed the crawler periodically returns to the site. The administrators of the search engine determine the frequency with which this happens [20]. The overall architecture of the web crawler is given in figure 1.

B. Web page Categorization

One of the fundamental problems in web information recovery is Web document categorization. The dimension and dynamism of the web generally rules out the possibility of Manual categorization. The normal alternative is to allot one of some predefined category labels to each document (classification) by employing a variety of supervised or unsupervised learning algorithms or create groups of related documents (clustering) [21]. In numerous information management and retrieval tasks, classification plays an important role. Focused crawling, assisted development of web directories, topic-specific web link analysis, and analysis of the topical structure of the Web necessitate classification of Web page content. The quality of web search can also be improved by the Web page classification [5]. Recently to classify web pages web directories such as Yahoo! And LookSmart are employed [22]. The goal over here is to assign a Web page to one or more predefined classes within a classification scheme. Chakrabarti *et al.* [23] used the predicted classes of pages in the neighborhood graph of a given web page to classify it. Attardi *et al*. [24] presented a technique that classified Web pages on the basis of the context of their URLs. They utilized a series of context strings obtained by exploiting the HTML structure for categorization.

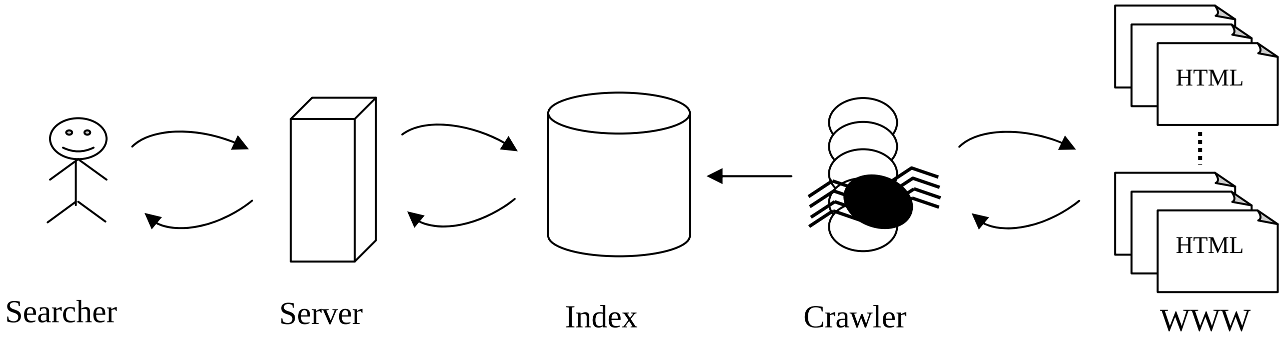

Fig. 1. The overall architecture of WebCrawler

C. MapReduce Framework

MapReduce, originally designed and implemented by Google is a distributed programming model for processing and generating large data sets [11], [25]. MapReduce is extremely appropriate for huge data searching and processing operations. For traditional clusters, the model has shown excellent I/O features, which is apparent from its successful application in large-scale search applications by Google [11], [26]. Data in the MapReduce framework [27, 28] are usually delineated in the form of key-value pairs <*key, value*>. In the primary step of the computation, the framework reads input data and optionally changes it into proper key-value pairs. In the second step which is the *map phase*, on each pair <*k, v*> a function *f* which returns a multiset of new key-value pairs is applied. i.e.,

$$f(<k,v>) = \{<k_1, v_1>, <k_2, v_2>, \cdots, <k_n, v_n>\}$$

Unlike the usual map, any number of results may be retuned by the function *f* and they are gathered at the time of the *map phase*. Then, in the *reduce phase*, all pairs that are generated in the preceding step are grouped according to their keys and their values are aggregated (reduced) using a function *g*:

$$g(\{<k_1, v_1>, <k_2, v_2>, \cdots, <k_n, v_n>\}) = <k,v>$$

To illustrate the MapReduce, we consider an example which counts the frequency of word lengths. The example process is shown in Figure 2. The input data contains a list of words with varying word lengths. At first, Map function obtains this input data and generates key value pairs. Here, key refers to the word length and value refers to the keyword. So, for the word “child”, a map function generates a key/value pair of “5/child”. Then, the key/value pairs with the same key are grouped and given to the reduce function. The reduce function obtains all the pairs with same key and counts the number of pairs. If a reduce function obtains a pair

with key “5”, it counts the number of the words that have a length of “5”. For example, {5,3} means that there are “3” words of word length “5”.

IV. DISTRIBUTED APPROACH FOR WEB PAGE CATEGORIZATION USING MAP-REDUCE PROGRAMMING MODEL

Web page classification or web page categorization is the process of associating a web page with one or several predefined category labels [5]. Web pages classification, allows web visitors to traverse a web site quickly and competently [34]. At present, two types of search engines are generally used by web users, they are *directory-style* search engines for example Yahoo! JAPAN [29] and ISIZE [30], and *robotstyle* search engines for example goo [31], excite [32] and altavista [33]. The web-pages that contain input keywords are listed by *robotstyle* search engines without considering the themes that characterize the respective Web-pages. Because of this, these search engines are liable to give misdirected Web-pages. Contrary to this, Web-pages stored in a database are classified with hierarchical categories and are ordered consistent with their themes by the directory-style search engines. This facilitates not only to follow input keywords but also to traverse hyperlinks that classifies Web-pages into categories in systematic order in order to get the Web-pages that contain the information that meets our need. However, present directory-style search engines employ man power for categorizing the large number of Web-pages into appropriate categories with their themes. Therefore, considerable time and care are required for this task. This shows that categorizing the continuously increasing number of Web-pages is an increasingly difficult task [35].

In order to reduce the computation task incurred due to the processing of huge number of web pages in directory-style search engines, we have used the successful distributed model, namely the MapReduce programming model. By taking advantage of MapReduce programming model, we an significantly reduce the processing time for such a computation-intensive operation. For web page categorization, web crawler can be initially used to collect the web pages from the web. Then, the web pages that are crawled by the web crawler can be given to the proposed approach for web page categorization. The proposed web page categorization approach is executed in the distributed environment using MapReduce programming model and the computation time incurred can be reduced considerably. The proposed web categorization approach is described in sub-section A. and the incorporation of web categorization approach in the MapReduce programming model is described in sub-section B.

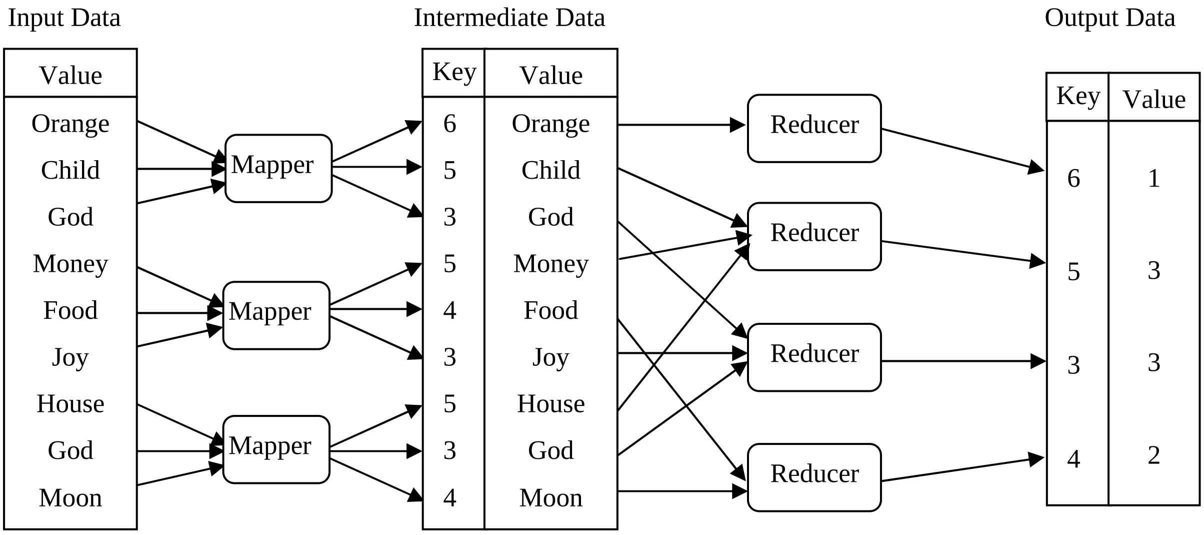

Fig. 2. Examples of MapReduce process

A. Proposed approach for web page categorization

To facilitate web page categorization, the crawled web pages are arranged into their relevant category based on the designed procedure. This section describes the concept involved in the proposed approach. Let w be the crawled web page which, is represented as two sets namely, V_1 and V_2 .

$$V_1 = \{ k_i \in w \; ; \; 1 \le i \le n \; \}; \; f(k_i) \ge T_f$$

$V_2 = \{ k_i \in w \ ; \ 1 \le i \le m \ \}; \ f(k_i) < T_f$

Both sets V_1 and V_2 contains a set of keywords that that are obtained from the the web page w. The set V_1 signifies the keywords k_i that satisfy the frequency threshold T_f given by the user within the web page w and set V_2 represents the set of keywords k_i that are not satisfying the frequency threshold (i.e., remaining keywords in the web page). Both these vectors are used as a representative of the web page w. Then, these two vectors are used to find the relevant category of the web page w. To find the relevant category (c_i), we compute the relevancy measure R_M in between these two vectors and the category vector C. The category vector contains 'k' number of sets and each set contains a set of domain keywords D_j that are relevant to their category c_i.

The category vector is represented as, $C = [c_1 \ \mathrm{c}_2 \cdots \mathrm{c}_\mathrm{k}]$ and the elements in the vector are represented as, $c_i = \{D_j, \alpha_j\}; \quad 1 \le i \le k \ ; \ 1 \le j \le l$, where D_j signifies a list of keywords and α_j specifies the weightage of relevant domain keyword D_j. The words in the category vector are predefined words (D_j) that are more relevant to the corresponding category (c_i) and each word in the category vector has weightage value based on its importance. This weighting method is very efficient for characterizing and distinguishing the most important keyword from others, and it provides better results for the application of web page categorization in the information retrieval system.

To identify the category label of web page w, we extract the keywords from the web page and based on their frequency, the sets V_1 and V_2 are formed. Then, we compute the relevancy between these two vectors with the predefined category vector C. Keyword-based relevancy measure relies on the idea that the content of a web page can be characterized by a set of keywords that is a set of words expressing the most significant concepts in the web page. The relevancy measure devised contains two parts, where the first part is based on frequent keywords and the second part is based on the remaining keywords of the web page w.

1) Frequent keyword-based similarity: The motivation behind this approach is that the most significant words are likely to be referred repeatedly, or, at least, more frequently than unimportant words. In practice, the words that are frequently occurring in a web page have more expressive power in the web page and also in the domain. Based on this, we have designed a frequent keyword-based similarity measure that gives more importance to the frequent keywords rather than infrequent keyword. For computing the *frequent keyword-based similarity* $S_{f_k}^{(V_1)}$, the frequent keywords in the set V_1 are matched with the category vector C, where for each category a list of keywords is presented with its weightage. The matched keywords are used to compute frequent keyword-based similarity measure of the web page.

$$S_{f_k}^{(V_1)} = \sum_{j=1}^{l} \alpha_j \, M_{D_j}^{(V_1)}$$

Where,

$$M_{D_j}^{(V_1)} = \left\{ \begin{matrix} 1 & ; & \textit{if the domain keywords } D_j \textit{ is matched with the keywords in set } \mathrm{V}_1 \\ 0 & ; & \textit{otherwise} \end{matrix} \right\}$$

l → Number of domain keywords in the category vector c_i

α_j → Weightage of domain keyword D_j

2) Keyword-based similarity: The relevancy measure R_M is not a best measure if it is only based on frequent keywords to categorize a web page. To overcome such a situation, we also incorporate keywords other than the frequent words for finding their suitable category. The importance of this *keyword-based similarity* measure $S_k^{(V_2)}$ is relatively less compared with the frequency based similarity

measure. The formulae used for computing the *keyword-based similarity* is given by,

$$S_k^{(V_2)} = \frac{1}{2} \sum_{j=1}^{l} \alpha_j \ M_{D_j}^{(V_2)}$$

where,

$$M_{D_j}^{(V_2)} = \begin{Bmatrix} 1 \ ; & if\ the\ domain\ keywords\ D_j\ is\ matched\ with\ the\ keywords\ in\ set\ \mathrm{V}_2 \\ 0 \ ; & otherwise \end{Bmatrix}$$

Based on the above two similarity measure, the overall relevancy measure R_M for matching the web page W with the individual category vector set c_i is given by,

$$R_M = S_{f_k}^{(V_1)} + S_k^{(V_2)}$$

$$R_M = \sum_{j=1}^{l} \alpha_j \ M_{D_j}^{(V_1)} + \frac{1}{2} \sum_{j=1}^{l} \alpha_j \ M_{D_j}^{(V_2)}$$

B. Adaptation of the proposed web page categorization approach to map-reduce programming model

. In this section, we present an overview of parallel approach proposed for web page categorization. Initially, we mine a set of web pages from the web using web crawler and then, the mined web pages are applied to the proposed approach for web page categorization. Here, we have used MapReduce programming model that was a patented software framework introduced by Google to support distributed computing on large data sets on clusters of computers. Using the MapReduce programming model, the proposed web page categorization approach is adapted to the distributed environment. The distributed approach of web page categorization using MapReduce framework is given in figure 3.

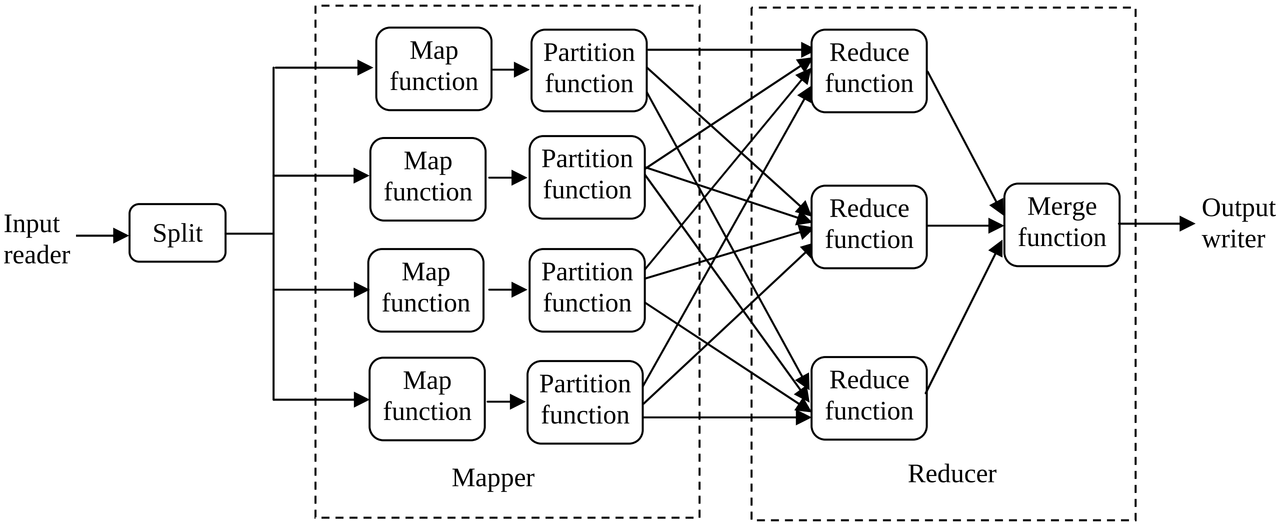

Fig. 3. The distributed approach of web page categorization using MapReduce programming model

1) Input reader: It reads web pages provided by the web crawler and splits them into subsets of data which is given as input to the map function. Let W_D be a set of web pages crawled by the web crawler in certain duration, $W_D = \{w_i; \quad 1 \le i \le N\}$. These web pages are split into 'q' number of subsets, where each subset contains the web pages,
$W_D{}^{(j)} = \{w_i; \quad 1 \le i \le N/q, \, j = 1,2,,\cdots q\}$.

2) Map function: Each map function (M) obtains a set of web pages (N/q) from the input reader and it extract keywords from the web pages and converts the web page w into <*key, value*> pair where, key refers to the keyword and value refers to the numerical value one. Then, each <*key, value*> pair is transformed into <*unique key, value*> pairs, where unique key refers to the unique keyword of the web page w and value refers to the frequency count of the keyword. In such a manner, for each web page, the map function returns a set of keyword and their frequency value. Here, we have used 'm' number of map function so that the computation time will depend on this number of map functions. If the value of 'm' is large, the computation time incurred by this parallel approach will be reduced significantly.

3) Partition function: The <*unique key, value*> pair of web page w obtained from the Map function is given to the partition function that forwards this <*unique key, value*> pair of web page w to all reduce functions present in the reduce network. Likewise, all <*unique key, value*> pair corresponding to the 'N' web pages are fed to the reduce function to find the relevancy measure of the web page with respect to all predefined categories.

4) Reduce function: Reduce function (R) obtains <*unique key, value*> pair of web page w from the map function through partition function. Here, for web page w, this function build two sets V_1 and V_2 where, V_1 contains the keywords that satisfy the predefined frequency threshold and V_2 contains a set of keywords which does not satisfy the frequency threshold. These two sets are constructed for web page w to find whether this web page w is appropriate for the predefined category c or not. The reduce function R contains one category set c that contains a set of predefined keywords. In order to find suitability, reduce function R computes the relevancy measure of the web page w based on the frequent keyword-based similarity and keyword-based similarity. The formula to be used for finding relevance measure is given in sub-section A. Finally, reduce function (R) returns a <*web page, relevancy_measure*> pair represented as, $\{w, R_M^{(w)}(R)\}$.

Similarly, for web page w, <*web page, relevancy_measure*> is computed with respect to all predefined categories in different reduce function present in the reduce network. So, every reduce function outputs one relevancy measure for web page w with respect to one predefined category. This procedure is done for all web pages in W_D. The designed programming model contains 'l' number of reduce function which is equivalent to the number of predefined categories present in the category vector.

5) Merge function: The merge function combines the <*web page, relevancy_measure*> given by all reduce function and generates <*web page, {relevancy_ measure1, relevancy_ measure2,…, relevancy_ measure l}* > pair for all web pages given by the web crawler. For example, the merge function output for a web page w is represented as,

$< w,\{R_M^{(w)}(R_1), R_M^{(w)}(R_2),\cdots, R_M^{(w)}(R_l)\} >$ where, w denotes the web page and $R_M^{(w)}(R_1)$ signifies the relevance measure of web page w with respect to the category set c_1 present in the reduce function R_1.

6) Output writer: The output writer obtains a set of <*web page, {relevancy_ measure1, relevancy_ measure2,* … *, relevancy_ measure l}* > pair for all web pages from the merge function. These pairs are then utilized by the output writer to find the category of the web pages. From the pair, the category of the web page w is identified by comparing all the relevancy measures of the web page w. The category which has the highest relevancy measure is the appropriate category label
for the web page w. Similarly, an appropriate category is identified for all the web pages.

The example process of the proposed approach is described in figure 4. In this figure, as an example, we have used two map functions and three reduce functions .Map function converts the input web pages into {keyword, frequency} pair and the reduce function obtains these pairs and generates a {page, relevancy} pair. Finally, the merge function combines all the outputs and finds the suitable category of the web page.

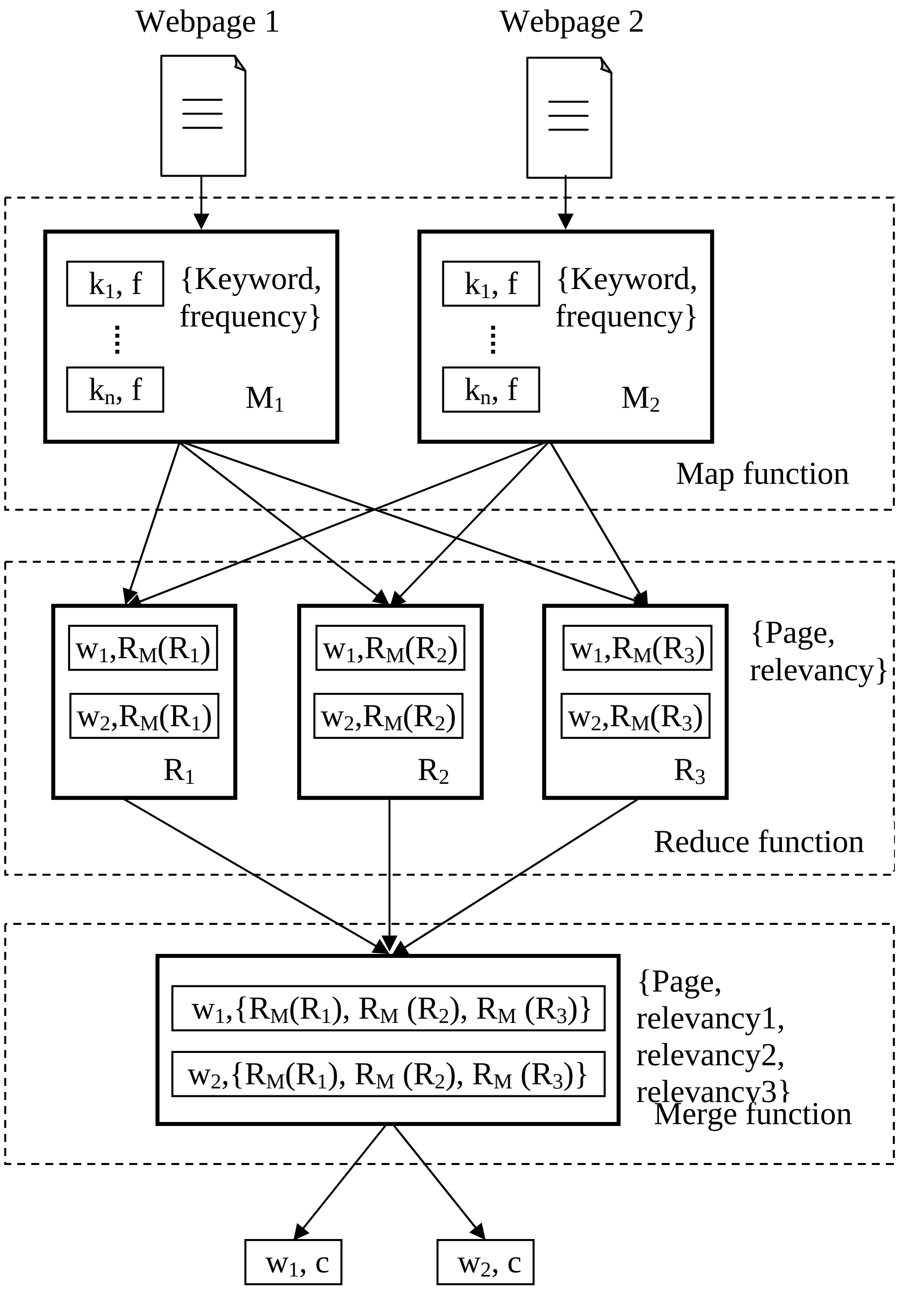

Fig. 4. The example process of the proposed distributed approach for web page categorization using map-reduce programming model

V. EXPERIMENTATION AND PERFORMANCE EVALUATION

The proposed approach for web page categorization was implemented using Java (jdk 1.6). The sample results of the proposed approach are given in sub-section A. and the performance of the proposed approach is given in sub-section B. The proposed approach is evaluated on test data using Precision, Recall and F-measure. *Test data:* This dataset is collected using the web crawler , websphinx [42, 43]. Websphinx is a personalized web crawler that mines the personalized web pages from the web. In our case, we mine the web pages related to biometrics, image processing management and networking. For every category, we mine a set of web pages and totally, 125 web pages are collected for classifying the web pages.

A. Experimental Results

The sample experimental results of the proposed approach are presented in this section. For sample results, we take 25 web pages from biometrics (w1 to w7), image processing (w8 to w14), management (w15 to w21) and networking (w22 to w25) and these web pages are given to the proposed approach for web page categorization. At first, these web pages are given to the map functions M1, M2, M3, M4 and M5. Here, we make use of thread parallelism to run these map functions in a parallel manner. The map functions compute the <keyword, frequency> pair of each web page after performing the stop word removal and stemming techniques. Then, these pairs related to each web page is given to the all reduce function for computing the relevancy measure. Here, we have used four reduce functions, which execute in a parallel manner using four threads. Each reduce function contains distinct predefined keywords from any one of the domains, image processing (R1), management (R2), biometrics (R3) or networking (R4). From every reduce function, we obtain one relevancy measure that illustrates the suitability of web pages with respect to the domain keyword. Finally, the merge function combines all relevancy measures of each web page obtained from all reduce functions and outputs their corresponding category (biometrics→ c1, image processing → c2, management→ c3, and networking→ c4). Table I shows the intermediate results obtained by the proposed approach while finding the suitable category for the web pages.

Table I . Sample Results obtained by the proposed approach while finding a suitable category for the web pages

Web Page	Map Function	$\{w,R_M(R_1)\}$	$\{w,R_M(R_2)\}$	$\{w,R_M(R_3)\}$	$\{w,R_M(R_4)\}$	category
w_1	M_1	0.7	0.25	4.5	0.25	c1
w_2		1.15	3.1997	5.3	0.7	c1
w_3		0.7	3	4.8	0.7	c1
w_4		0	0.25	1.9	0	c1
w_5		0.45	1.25	2.3	0.9	c1
w_6	M_2	1.6	0.5	3.85	0.45	c1
w_7		0.7	2.55	5.5	0.7	c1
w_8		4.999	0.25	0	0	c2
w_9		2	0.25	0	0	c2
w_{10}		1.75	0.25	0	0	c2
w_{11}	M_3	0.9	0.25	0	0	c2
w_{12}		2.1	0.25	0	0	c2
w_{13}		0.45	0.25	0	0	c2
w_{14}		0.9	0	0	0	c2
w_{15}		0.8	3.05	0	0.45	c3
w_{16}	M_4	0	3.5	0	0.45	c3
w_{17}		0	3.5	0	0.45	c3
w_{18}		0	3.05	0	0.45	c3
w_{19}		0.8	2.25	0	0.45	c3
w_{20}		0	3.05	0	0.45	c3
w_{21}	M_5	0	1.25	0.3	0.9	c3
w_{22}		0.3	1.8	0.3	2.95	c4
w_{23}		0	1.5	0.3	4.35	c4
w_{24}		0	2.15	0.3	3.2	c4
w_{25}		0.75	0.7	2	2.9	c4

B. Performance Evaluation

The performance of the proposed approach is evaluated on *test data* using the evaluation metrics namely, Precision, Recall and F-measure. The test data obtained from the web crawler is given to the Map Reduce framework that provides the appropriate category label for each web page. Thus, the results obtained are then used to compute the following evaluation metrics which are used to find whether the categorization is done accurately or not. The definition of the evaluation metrics is given as,

$$\text{Recall}\quad(\alpha,\beta) = C_{\alpha\beta} / C_{\alpha}$$

$$\text{Precision }(\alpha,\beta) = C_{\alpha\beta} / C_{\beta}$$

$$F(\alpha,\beta) = \frac{2 * \text{Recall}(\alpha,\beta) * \text{Precision }(\alpha,\beta)}{\text{Precision }(\alpha,\beta) + \text{Recall}(\alpha,\beta)}$$

where $C_{\alpha\beta}$ is the number of members of topic α in category β, C_{β} is the number of members of category β and C_{α} is the number of members of topic α.

For *test data*, we make use of five map functions ($m = 5$) and four reduce functions ($l = 4$) to execute the proposed approach in parallel manner. The output is used to calculate the precision, recall and F-measure of the resultant categories given in Table II. Finally, the results are plotted as graphs and are shown in figure 5. From these graphs, it is evident that the precision, recall and F-measure obtained by the proposed approach are highly accurate for all categories.

Table II. Precision, Recall and F-measure of test data

Category	No. of web pages in topic C_{α}	No. of web pages in the category C_{β}	No. of web pages in category related to topic $C_{\alpha\beta}$	Precision	Recall	F-measure $F(\alpha,\beta)$
C_1 (Biometrics)	35	37	33	0.89	0.94	0.91
C_2 (Image processing)	30	28	26	0.92	0.86	0.89
C_3 (Management)	30	40	28	0.7	0.93	0.8
C_4 (Networking)	30	20	18	0.9	0.6	0.72

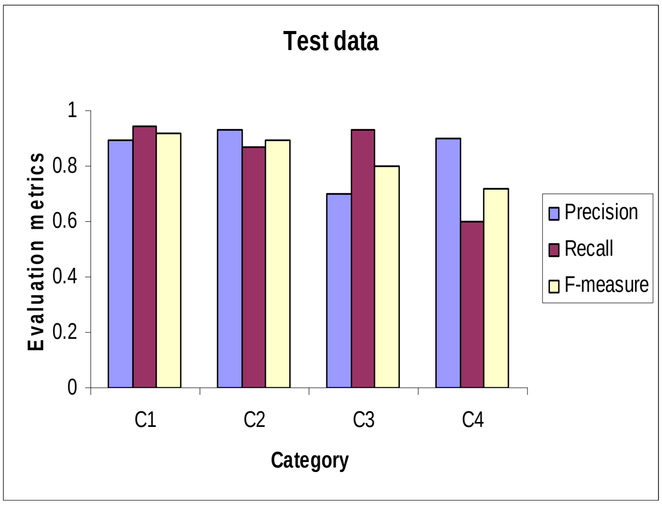

Fig.5. Performance of the proposed approach on test data

VI. CONCLUSION

The numbers of web pages have increased and identifying the required information effortlessly and instantly from the thousands of web pages retrieved by a search engine is a difficult task. For effectively addressing this difficulty of retrieving information from the Internet and to solve this problem, web page classification techniques has been proposed by several researchers. With this intention, we have developed an efficient web page categorization approach based on the parallel approach. At first, the web crawler was used to mine the World Wide Web and the web pages were categorized using the proposed parallel approach, where the web page categorization technique was incorporated into the MapReduce programming model. The proposed approach was used to identify a suitable category, based on the relevancy measure that was designed based on the frequency of keywords and the weights associated with the predefined keywords specified in the category. The experimentation ensured that the proposed parallel approach effectively classifies the web documents and in addition, the computation task was reduced significantly.

REFERENCES

[1] M. H. Marghny and A. F. Ali, "*Web Mining Based On Genetic Algorithm*", In Proceedings of ICGST International Conference on Articial Intelligence and Machine Learning(AIML- 05), 19-21 December, Cairo, Egypt, 2005.

[2] Arul Prakash Asirvatham and Kiranthi Kumar Ravi, "*Web Page Categorization based on Document structure*", Technical report, International Institute of Information Technology, Hyderabad, India, 2001.

[3] Xin Jin, Rongyan Li, Xian Shen, Rongfang Bie, "*Automatic web pages categorization With Relief and Hidden Naïve Bayes*", Proceedings of the 2007 ACM symposium on Applied computing, Seoul, Korea, pp. 617-621, 2007.

[4] Rajni Pamnan, Pramila Chawan, "*Web Usage Mining: A Research Area in Web Mining*", In Proceedings of International Symposium on Compute Engineering & Technology (ISCET), pp.73-106, 2010.

[5] Xiaoguang Qi and Brian D.Davison, "*Web Page Classification : Features and Algorithms*", ACM Computing Surveys (CSUR), Vol. 41, No.2, 2009.

[6] Sven Meyer zu Eissen and Benno Stein, "*Genre classification of web pages*",Lecture Notes in Computer Science, Springer,Berlin ,Vol.3238, pp.256-269, 2004.

[7] Z. Gyongyi and H. Garcia-Molina, "*Web spam taxonomy*", In Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 05), Bethlehem, PA, pp. 39–47,2005.

[8] Carlos Castillo, Debora Donato, Aristides Gionis, Vanessa Murdock , Fabrizio Silvestri, "*Know your neighbors: web spam Detection using the web topology*", In Proceedings of the 30th annual International ACM SIGIR conference on Research and developmentt in information retrieval, Amsterdam, The Netherlands, pp. 423 - 430, 2007. .

[9] Shakirah Mohd Taib, Soon-Ja Yeom, Byeong-Ho Kang, "*Elimination of Redundant Information for Web Data Mining*", In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), Vol. 1, pp. 200 - 205, 2005.

[10] Almasi, G.S. and A. Gottlieb, "*Highly Parallel Computing*", Benjamin Cummings publishers, Redwood City, CA,1989.

[11] Jeffrey Dean and Sanjay Ghemawat, "*MapReduce: Simplified Data Processingon Large Clusters*", In Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation , Vol. 6 , San Francisco, CA, December, 2004.

[12] Shimin Chen, Steven W. Schlosser, *Map- Reduce Meets Wider Varieties of Applications*, Intel Research Pittsburgh Technical Report, IRP-TR-08-05, May, 2008.

[13] Bingsheng He Wenbin Fang Naga K. Govindaraju, Qiong Luo Tuyong Wang, "*Mars: A MapReduc e Framework on Graphics Processors*", pp. 260-269, 2008.

[14] Matei Zaharia, Andy Konwinski, Anthony D. Joseph, Randy Katz, Ion Stoica, "*Improving MapReduce Performance in Heterogeneous Environments*", Technical Report, No. UCB/EECS-2008-99.

[15] Tyson Condie, Neil Conway, Peter Alvaro, Joseph M. Hellerstein , John Gerth, Justin Talbot, Khaled Elmeleegy, Russell Sears, *Online aggregation and continuous query Support in MapReduce*", Proceedings of the International conference on Management of data, pp. 1115-1118 ,Indianapolis, Indiana, USA, 2010.

[16] Jimmy Lin, Donald Metzler, Tamer Elsayed and Lidan Wang, "Of Ivory and Smurfs: Loxodontan MapReduce Experiments for Web Search ", In Proceedings of the Eighteenth Text Retrieval Conference (TREC 2009), Gaithersburg, Maryland, November 17-20, 2009.

[17] Gautam Pant and Padmini Srinivasan, "*Link Contexts in Classifier-Guided Topical Crawlers*", IEEE transactions on knowledge and data engineering, vol.18, no. 1, January 2006.

[18] Brian Pinkerton, "*WebCrawler : Finding What people want*", Technical Report, University of Washington, November 2000.

[19] Gautam Pant, "*Learning to Crawl: Classifier-Guided Topical Crawlers*", Technical Report, The University of Iowa, July 2004.

[20] Sandeep Sharma, "*Web-Crawling Approaches in Search Engines*", Technical Report, Thapar University, Patiala, June 2008.

[21] Zoltan Gyongyi , Hector arcia-Molina, Jan Pedersen, "*Web content categorization using link information*", Technical report, Stanford University ,2006.

[22] Susan Dumais, Hao Chen, "*Hierarchical Classification of Web Content*", In Proceedings of the 23rd annual international ACM SIGIR conference on Research and Development in information retrieval, pp. 256 - 263, Athens, Greece, 2000.

[23] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan, "*Automatic Resource List Compilation by Analyzing Hyperlink Structure and Associated Text,*" in Proceeding of Seventh International conference on World Wide Web, 1998.

[24] G. Attardi, A. Gullı´, and F. Sebastiani, "*Automatic Web Page Categorization by Link and Context Analysis,*" Proc. THAI- 99, First European Symp. Telematics, Hypermedia, and Artificial Intelligence, 1999.

[25] Jacopo Urbani, "*RDFS/OWL reasoning Using the MapReduce framework*", Technical Report, Vrije Universiteit, July 2, 2009.

[26] M. Mustafa Rafique, Benjamin Rose, Ali R Butt, Dimitrios S. Nikolopoulos, "*CellMR: A Framework for supporting mapreduce on asymmetric cell-based clusters*", In Proceedings of the IEEE International Symposium on Parallel & Distributed Processing, pp. 1-12, 2009.

[27] Petr Krajca and Vilem Vychodil, "*Distributed Algorithm for Computing Formal Concepts Using Map-Reduce Framework*", Lecture Notes in Computer Science; Vol. 5772, Springer- Verlag , 2009.

[28] Weiyi Shang, Zhen Ming Jiang, Bram Adams, Ahmed E. Hassan, "*MapReduce as a General Framework to Support Research in Mining Software Repositories (MSR)*", in proceedings of 6th IEEE International Working Conference on Mining Software Repositories, pp. 21 – 30, Vancouver, BC, 2009.

[29] Yahoo Japan from "*http://www.yahoo.co.jp/*".

[30] ISIZE from "*http://www.isize.com/*".

[31] Goo from "*http://www.goo.ne.jp*/".

[32] Excite from "*http://www.excite.co.jp*/".

[33] AltaVista from "*http://www.altavista.com*/".

[34] Lee Zhi Sam, Mohd Aizaini Maarof and Ali Selamat, "*Automated Web Pages Classification with Independent Component Analysis*", in Proceeding of The 2nd Postgraduate Annual Research Seminar, PARS'06, Faculty of Computer Science & Information Systems, Universiti Teknologi Malaysia, 24 - 25 May, 2006.

[35] Makoto Tsukada, Takashi Washio and Hiroshi Motoda, "*Automatic Web- Page Classification by Using Machine Learning Methods"*, Lecture Notes In Computer Science, Vol. 2198, pp. 303-313, 2001.

[36] Jost Berthold, Mischa Dieterle and Rita Loogen, "*Implementing Parallel Google Map- Reduce in Eden*", Lecture Notes in Computer Science, Springer ,Vol. 5704,pp.990- 1002, 2009.

[37] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: simplified data processing on large clusters", Communications of the ACM, Vol.51, No.1, pp. 107-113, January 2008.

[38] Tamer Elsayed, Jimmy Lin , Douglas W. Oard, "*Pairwise document similarity in Large collections with MapReduce*", In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, pp. 265-268 , June 16-17, Columbus, Ohio, 2008.

[39] Nuanwan Soonthornphisaj and Boonserm Kijsirikul, "*Iterative cross-training: An Algorithm for web page categorization"*, Intelligent Data Analysis, Vol. 7, No.3, pp. 233 – 253, 2003.

[40] Jebari Chaker and Ounelli Habib, "*Genre Categorization of Web Pages*", In Proceedings Of the Seventh IEEE International Conference on Data Mining Workshops, pp. 455-464, October 28-31, Omaha, Nebraska, USA ,2007.

[41] Jane E. Mason , Michael Shepherd, Jack Duffy, "*Classifying Web Pages by Genre: An n-Gram Approach*", In Proceedings of the 2009 IEEE/WIC/ACM I nternational Joint Conference on Web Intelligence and Intelligent Agent Technology , Vol.1, pp. 458- 465, Milan, Italy, 2009.

[42] WEBSPHINX: A personal, customizable web crawler (accessed on 04/2008), http://www.cs.cmu.edu/rcm/websphinx/

[43] Manuel Salvadores, Landong Zuo, SM Hazzaz Imtiaz, John Darlington, Nicholas Gibbins, Nigel R Shadbolt and James Dobree, "*Market Blended Insigh t: Modeling Propensity to Buy with the Semantic Web"*, Lecture Notes in Computer Science, Vol.5318, pp. 777-789, 2010.