International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 5 Issue: 5 550 – 555

Context Based Indexing On Synonym System Using Hierarchical Clustering In Web Mining

Sunaina
M. Tech Scholar of
Computer Science and Engineering
Rawal Institution of Engineering And Technology, MDU
Faridabad, India
soni.sunaina@gmail.com

Dr. N.N. Das
Associate Professor at
Computer Science and Engineering Department
Rawal Institution of Engineering And Technology, MDU
Faridabad, India
nripendradas@gmail.com

Abstract—Now a days, the World Wide Web is the collection of large amount of information which is increasing day by day. For this increasing amount of information, there is a need for efficient and effective indexing structure. Indexing in search engines has become the major issue for improving the performance of Web search engines, so that the most relevant web documents are retrieved in minimum possible time. For this a new indexing mechanism in search engine is proposed which is based on indexing the synonym terms of the web documents, a synonym term which have multiple context with same meaning of the web documents. The indexing is performed on the bases of hierarchical clustering method which clustered the similar term documents into the same cluster and these clusters are clubbed together to form mega cluster on the basis of synonym term. With the similarity of clusters, it will optimize the search process by forming the different levels of hierarchy. Finally, it will give fast and relevant retrieval of web documents to the user.

Keywords-*Context, Synonym; Hierarchical Clustering; Indexing; Ontology Repository.*

*****

I. INTRODUCTION

With the fast growing internet, the World Wide Web (WWW) has become one of the most important resources for obtaining the information. There are huge amounts of documents existing in the World Wide Web and finding information from WWW according the user interest becomes a critical task. The main aim of the search engine is to provide most relevant documents to the users in minimum possible time. To improve the search results according to the user's query indexing is performed on the web pages. And the most relevant result is found according to the user. The proposed approach is used in which context based hierarchical clustering [9] of web documents is performed. The purpose is to cluster or clubbed the web documents based on the context similarity of the documents. The aim of clustering based indexing is to assign the context similar documents within the same cluster. Further the hierarchical clustering is applied in which similar clusters are clubbed to form a mega cluster and similar mega cluster are combined to form super cluster on basis of Synonym similarity. This will provide the user the best possible matching results in minimum possible time by directing the search process to a specific path form higher levels of clustering to the lower levels. In this paper, the indexing of documents is performed on the basis of context of the documents rather than on the basis of terms. The context of the documents is extracted from the ontology repository. The architecture for the proposed work is represented in Figure 3.1 According to the architecture, firstly the web pages are gathered by crawler and are stored in repository of web pages. After this the preprocessing on documents is performed by indexer for extracting the keywords with their frequency count in documents with their corresponding doc Ids. Now, the keywords whose frequency count is greater than threshold value are extracted and the different contexts of terms are gathered from thesaurus, which are maintained in ontology repository for deducing the context of the documents. Finally, clustering is performed with the extracted keywords.

II. RELATED WORK

In this section, a review of the previous work on indexing is given. In this field of index organization and maintenance, many algorithms and techniques have already been proposed but they seem to be less efficient in accessing the index.

A. Hao liang, Wanli Zuo, Fei Ren, Chong Son (1)

In this paper authors proposed an attribute search-driven mechanism, in this paper the most important factor is the attributes and semantic relations between them. In this paper try to extract abundant attributes, which describe the concept, and the relationships between the set of attributes of same search form and even different forms. The most efficient and effective technique of detecting the semantic relation between words is the WordNet [3]. We extend each attribute into a concept set which is used for matching attributes. The framework takes source query form and target query form as inputs and output a query for target query. During the transaction, we first extract attributes from query forms and find the semantic relation between attributes, and then compose attributes according to the web semantic restriction, finally rewrite the query for target form.

B. N. Chauhan and A. K. Sharma (2)

In this paper authors proposed, the context driven focused crawler (CDFC) that searches and downloads only highly relevant web pages, thus, reducing the network traffic. A category tree has been used, which provides flexibility to the user for interacting with the system showing the broad categories of the topics on the web. The proposed design significantly reduces the storage space at the search engine side.

C. Mike Perkowitz and Oren Etzioni (3)

In this paper authors formalize index page synthesis as a conceptual clustering problem and introduce a novel approach which we call conceptual cluster mining: In this paper authors search for a small number of cohesive clusters that correspond

ISSN: 2321-8169
550 – 555

to concepts in a given concept description language L. Next, we present SGML, an algorithm schema that combines a statistical clustering algorithm with a concept learning algorithm. The clustering algorithm is used to generate seed clusters, and the concept learning algorithm to describe these seed clusters using expressions in L.

D. Yitong Wang and Masaru Kitsuregawa (4)

In this paper, authors proposed a new approach to cluster search results returned from Web search engine using link analysis. Unlike document clustering algorithms in IR that based on common words/phrases shared between documents, this approach is based on common links shared by pages using co-citation and coupling analysis. In this paper authors also extend standard clustering algorithm K-means to make it more natural to handle noises and apply it to web search results.

III. PROPOSED WORK

This paper proposes an algorithm for indexing the web documents on the basis of synonym terms and context of the documents using hierarchical clustering in data mining. The proposed indexing mechanism will group the similar documents into the one clusters and further these clusters are clubbed into mega clusters on the basis of synonym similarity. The proposed architecture of indexing of documents is shown in Fig1.

A. COMPONENTS OF PROPOSED ARCHITECTURE

The proposed architecture of indexing in data mining consists of the following functional components.

*1) Crawler:*Web crawler is software for downloading pages from the Web automatically. It is an important component of web search engines, where they are used to collect the corpus of web pages indexed by the search engine.

*2) Web Page Repository:*This is the collection of web documents that have been collected by the crawler from the WWW. It is a database which stores the web pages that are gathered by the crawler from WWW in order to provide web documents for indexing purpose.

*3) Preprocessing of Documents:*Pre-processing of input document is an integral part of Tokenization, which involves the preprocessing of documents and generates its respective tokens. It involves stemming as well as removal of stop words.

4) Frequency Count: On the completion of stemming process, next step is to count the frequency of each word. Frequency count denotes the number that how many times the particular term is repeated in the documents.

*5) Keyword Extraction Greater than Threshold Value:*This step allows to removing or eliminating the indiscriminate terms from the documents to improve the document clustering accuracy and reduce the computational complexity. Algorithm for keyword selection according to the threshold value is given in Figure 4.4.

*6) Thesaurus:*It is a dictionary of words available on the World Wide Web from thesaurus.com which contains the words as well as their multiple meanings.

7) Ontology Repository: After the extraction of the keywords from the documents, and extracting the multiple context of the keywords from the thesaurus, this task is further extended by forming their structural framework which would represent the relationship and thus the semantic meaning of the document, and such representations are referred as 'Ontologies'. Ontology repository contains various concepts with their relationships.

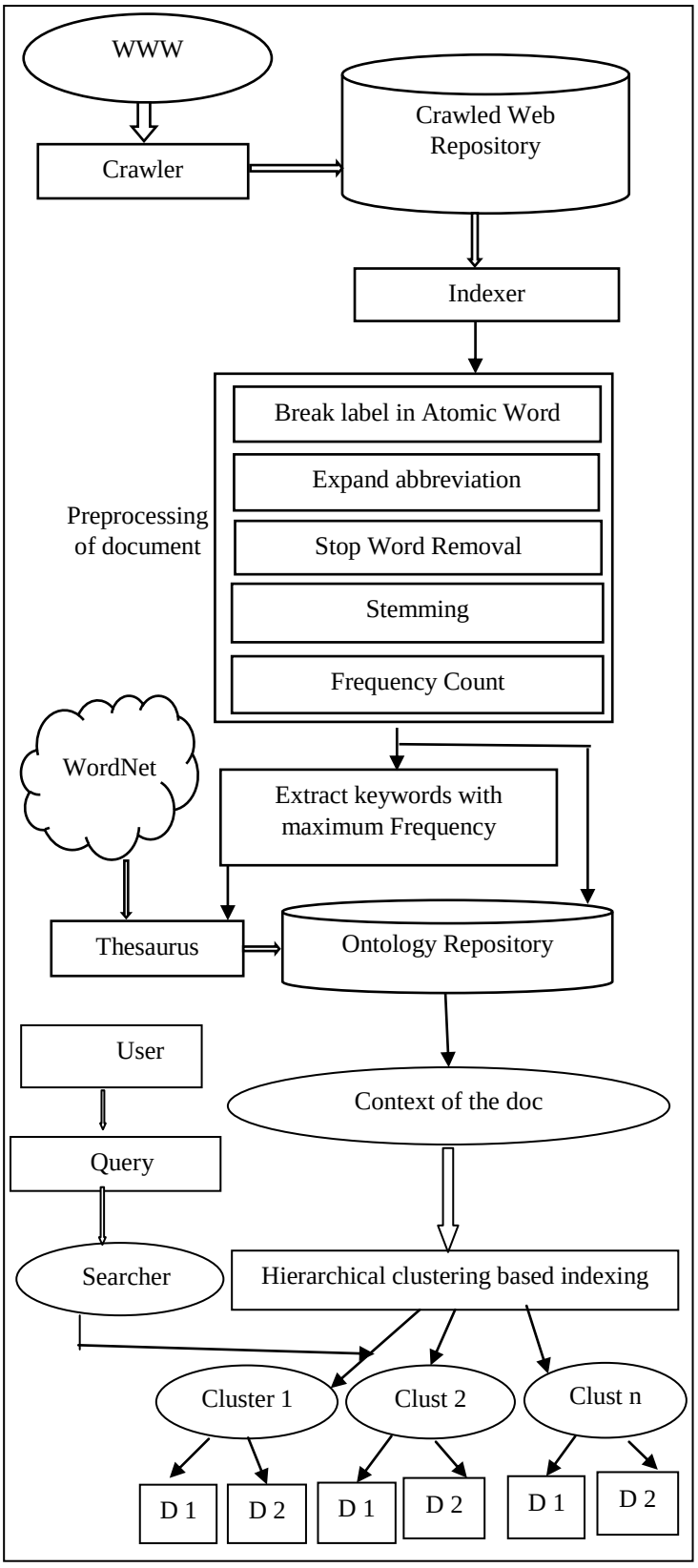

Figure 1. Architecture of Hierarchical Clustering.

8) Context of the Document: The context of the document deduce from ontology represent the semantic or theme of the document. At this, level the different documents retrieved for the same term are categorized according to the context. The document context has been extracted using thesaurus and ontology repository.

9) Hierarchical Clustering based Indexing: As the context of the document is determined, the clustering of the documents is performed on the basis of context similarity of documents. In hierarchical clustering based indexing, initially the clustering of documents is done on the basis of context similarity and keyword similarity of the documents, which is calculated using the similarity measure. Further the mega clusters out of the similar clusters are generated on the basis of similarity between the hyponyms terms of the clusters.

*10) Searcher:*It is that module of the search engine that receives user queries via the user interface and hence after searching the results in the index provides them to the user.

*11) Query Interface:*It is that user interface through which user types the query.

a) The web pages are crawled by the web crawler from WWW and are collected into the web page repository.

b) In the preprocessing of documents indexing is performed by which it includes the fetching of documents from repository with their corresponding doc ids and perform keyword extraction phase, stop word removal, stemming and frequency count.

c) When the document preprocessing is complete, the keywords with frequency count greater than threshold value are extracted.

d) The keywords that are selected in previous step are searched in the thesaurus for extracting the multiple meanings of the terms. This step will help in generating multiple context of the keywords.

e) Now the multiple contexts and the terms of the document are compared with the ontology repository. The context of the document is extracted by matching the keywords of the documents and the multiple contexts with the concepts and the relationship terms in the ontology repository.

f) Now the hierarchical clustering based indexing of documents is performed on the basis of context similarity of the documents and synonym similarity of clusters by calling following algorithms.

- Call algorithm Document_Similarity.
- Call algorithm Document_Clustering.
- Call algorithm Mega_Clustering.

g) The synonym term along with the mega cluster Id are indexed into the posting list by the previous step. The posting list consists of three columns, one containing the synonym term, second containing mega cluster Id and third containing cluster Id. At this step final indexing is performed on the basis of both context as well as the synonym term.

h) When the user fire a query, then the index is being searched for the synonym term.

i) After matching the synonym term the mega cluster Id is fetched this gives the corresponding cluster Id of the clusters containing the relevant documents according to the different context of the terms.

j) Now the user can fetch clusters according to the context in which the user desires.

Figure 2. Algorithm for Constructing Hierarchical Clustering.

B. ALGORITHM FOR COMPUTING SIMILARITY MATRIX:

Let D= {D1, D2,……Dn) be the collection of N number of textual documents being crawled to which consecutive integers document identifiers 1…n are assigned. Each document Di can be represented by a corresponding set Ki such that Ki is a set of all the keywords extracted from Di. Let us denote that set by D* such that D*= {K1,K2,……. Kn}. The similarity of any two documents Ki and Kj can be computed using the similarity measure.

- Similarity_measure (Ki, Kj) = |Ki ∩ Kj | / |Ki U Kj |

If the context of the document Di and Dj are same then the similarity measure is calculated as:

- Similarity_measure(Ki, Kj) = |Ki ∩ Kj | / |Ki U Kj | +1

The algorithm constructs the document similarity matrix. The algorithm calculates the similarity of each document with every other document using the similarity_measure given above is given in Fig 3. The number of calculations performed only leads to the formation of the upper triangular matrix. The rest of the values in the similarity matrix are assigned as similarity_measure(i,j)= similarity_measure(j,i).

Algorithm 3: Document Similarity

Input: The set D*={ K1,K2 ,…….. Kn } where Ki is a set of all the keywords of Similarity Matrix of order N*N.

```
for i=1 to n
begin
      sim[i][j]=0;
for j=i+1 to n
begin
       if(context.Di= =context.Dj)
then
sim[i][j]=similarity_measure(Ki,Kj) +1;
sim[j][i]=sim[i][j];
else
      sim[i][j]=similarity_measure(Ki,Kj);
      sim[j][i]=sim[i][j];
end for
end for
```

Figure 3. Algorithm For Computing Similarity.

C. ALGORITHM FOR DOCUMENT CLUSTERING AND MEGA CLUSTERING.

The clustering algorithm which clusters together the similar documents and the hierarchical clustering algorithm aims at forming the mega clusters out of the similar clusters.

Algorithm 3: Document Clustering

```
i=1
for f=1 to c                    //for number of clusters
begin
cs=0                            // initially cluster is empty
for e=1 to n/c                  // n is number of document
  begin
    for j=1 to n
          Select max from sim[i][j]
          if(context.Di= =context.Dj)
                cs = cs U Ki
                D*=D*-Ki
            for p=1 to n
                begin
                      sim[i][p]=0;
                      sim[p][i]=0;
          end
         i=j
end
end
```

Figure 4. Algorithm for Document Clustering.

Algorithm 4: Mega_Clustering

Input: CS={C_1,C_2,C_3,…..C_3)
Output: MC={MC_1,MC_2,MC_3,……MC_m)

```
i=1
for f=1 to m        //number of mega cluster
begin
          MC=0
for e=1 to n/m
begin
for j=2 to n
          MC=Ci
          CS=CS-Ci
                if (synonym.Ci= =synonym.Cj)
                MC=MC U Cj
                CS=CS-Cj
          else
          Cj is not added to the cluster;
end
          i=j
end
```

Figure 5. Algorithm for Mega Clustering.

The hierarchical clustering algorithm creates mega clusters out of the similar clusters. This algorithm, the first mega cluster is considered which is initially empty. The first cluster from the collection is considered and put in the first mega cluster. Now, using the synonym term similarity of the cluster, the next cluster is included in the first mega cluster, this will repeat until all the synonym similar cluster are clubbed into single mega cluster. If the cluster are not synonym similar then it will put into second cluster and it will takes the role of first cluster and the same procedure is repeat until all similar clusters are put into second mega cluster and so on.

TABLE I. EXTRACTED KEYWORDS FROM DOCUMENTS AFTER PREPROCESSING

KEYWORD	FREQUENCY COUNT	DOC ID
Age	6	1,2,3,4,6,10
Period	4	1,2,4,6
Lifetime	1	1
Era	1	1
Stock	2	1,4
Formation	2	1,4
Duration	1	2
Season	1	2
Internal	1	2
Space	1	2
Term	1	2
Eldership	2	3,10
Priority	2	3,10
Superiority	2	3,10
Value	6	4,5,6,7,10,11
Reason	1	4
Statement	1	5
Relation	3	5,6,11
Profit	3	5,6,11
Story	4	5,6,7,11
Chief	3	7,9,11
Director	2	7,9
Guide	2	7,9
Head	1	7
Mass	2	8,9
Whole	2	8,9
Assemblage	1	8
Substance	1	8
Collection	1	8
Country	1	12
District	1	12
Part	1	12
Portion	1	12

TABLE II. EXTRACTED CONTEXT AND SYNONYM TERM FOR DOCUMENTS

KEYWORDS SET	DOC ID	CONTEXT	SYNO -NYM
{Age, Period, Lifetime, Era, Stock, Formation}	1	Generation	Age
{Age, Period, Duration, Season, Interval, Space, Term}	2	Date	Age
{Age, Eldership, Priority, Superiority}	3	Seniority	Age
{Age, Period, Stock, Formation, Value, Reason}	4	Generation	Age
Statement, Relation, Value, Profit, Story}	5	Amount	Value
{Age, Period, Profit, Relation, Value, Story}	6	Amount	Value
{Chief, Director, Guide, Head, Value, Story}	7	Leader	Value
{Mass, whole, Assemblage, Substance, Collection}	8	Body	Value
{Mass, Whole, Chief, Director, Guide}	9	Leader	Value
{Eldership, priority, Superiority, Value, Age}	10	Seniority	Age
{Relation, Value, Profit, Story, Chief, Direction}	11	Leader	Value
{Country, District, Part, Portion}	12	Region	Value

TABLE III. CLUSTERING OF DOCUMENTS

CONTEXT	SYNONYM	CLUSTER ID	DOCUMENT ID
Generation	Age	C1	1,4
Date	Age	C2	2
Seniority	Age	C3	3,10
Amount	Value	C4	5,6
Leader	Value	C5	7,9,11
Body	Value	C6	8
Region	Value	C7	12

Extract the keywords which have frequency count greater than threshold value=3. Only the keyword “Age” and “Value” has frequency count greater than 3. Both keywords will select and their multiple contexts and all other information will be extracted from thesaurus and maintained in the ontology repository in the form frames. After extracting the context of the documents using ontology with the synonym term, following table will generated as shown in Table II. The similarity among the documents is computed on the basis of context similarity measure, and similarity matrix is constructed. According to the clustering algorithm documents are clustered to form clusters as shown in Fig 6.

- 1st cluster will have document 1 and 4 (all the documents have the same context Generation).
- 2nd cluster will have document 2 (all the documents have the same context Date).
- 3rd cluster will have document 3 and 10 (all the documents have the same context Seniority).
- 4th cluster will have document 5 and 6 (all the documents have the same context Amount).
- 5th cluster will have document 7, 9 and 11 (all the documents have the same context Leader).
- 6th cluster will have document 8 (all the documents have the same context Body).
- 7th cluster will have document 12 (all the documents have the same context Region).

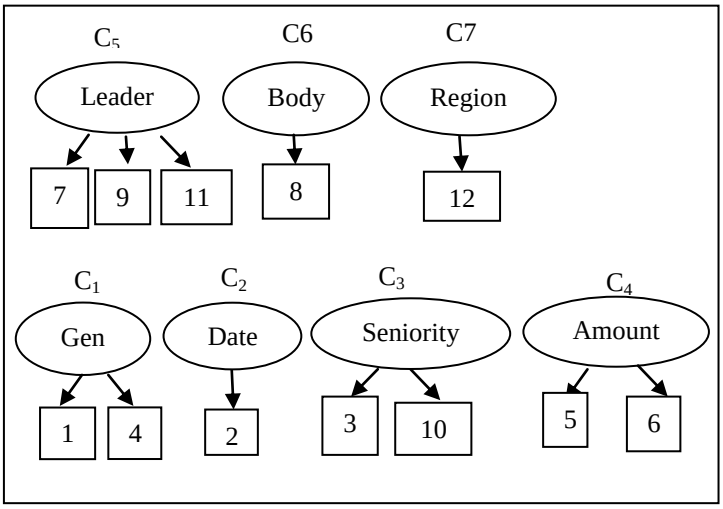

Figure 6. Cluster Formations of Documents.

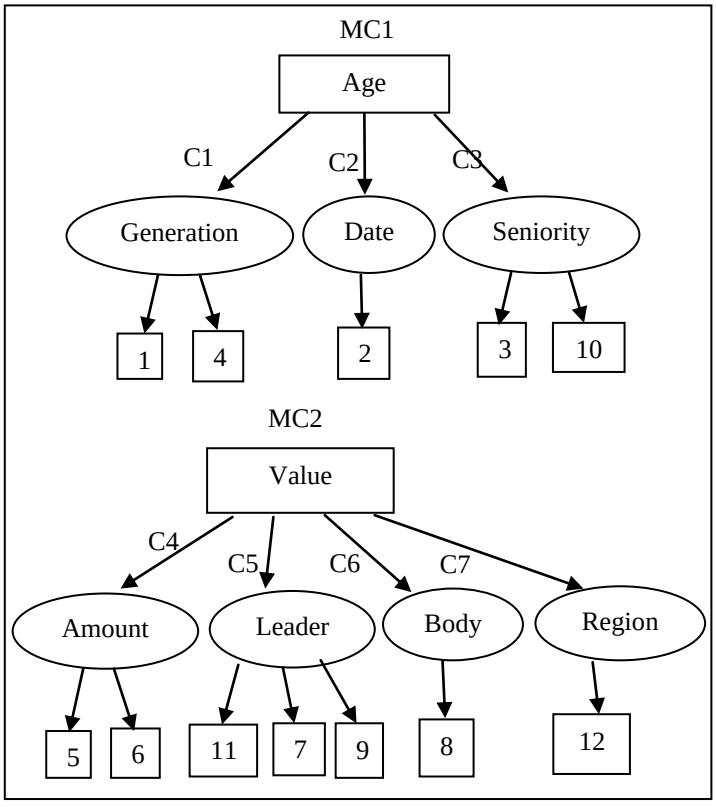

Figure 7. Mega Cluster Formations of Clusters.

- Formation of mega clusters by clubbing the clusters on the synonym similarity by using the mega clustering algorithm.
- We have cluster set CS= {C1, C2, C3, C4, C5, C6, C7}.
- Now algorithm will check the synonym similarity of the clusters and combine them into same mega clusters if they are synonym similar.
- 1st mega cluster will have C1, then C2,and last C3(having same synonym term Age).
- 2nd mega cluster will have C4, C5, C6, and C7(having same synonym term Value).

TABLE IV. FINAL INDEX ON THE BASIS OF SYNONYM

SYNONYM	MEGA CLUSTER ID	CLUSTER ID
Age	MC1	C1 ,C2 ,C3
Value	MC2	C4 ,C5, C6, C7

- Finally, the index is constructed on the basis of synonym term and context of the document as shown in Table IV.
- Now when the user fires a query 'Age' the keyword in user's query are matched with the synonym term in the index, if match is found then the search will start from mega cluster MC1 and proceed to cluster C1, C2, C3 which gives different context of the documents now the user can retrieve relevant documents according to the context he/she desires. Thus, the proposed indexing approach will give better results to a great extent and the more relevant Web pages can be presented to the user and provide better performance to the users.

IV. CONCLUSION

In the context based indexing on synonym system using hierarchical clustering in web mining we performed clustering on the bases of synonym system. On the WWW there are huge amount of document stored and then how a user will extract the useful information from these document is a difficult task. So, the system helps us to find out fast results as compare to earlier systems. We make cluster of similar types of items. The document having similar types of data is collect in a separate cluster. After making the separate cluster we make mega cluster on the behalf of these clusters.

The document having same context is placed in separate cluster. The synonym term indexing will help the user to extract all the relevant documents which contain every context, and user can choose the document cluster according to the context he/she wants. In this way, all the different context documents are fetched as a group. So, the problem of getting irrelevant results for the query using the term having multiple context is reduce to some extent.

REFERENCES

[1] Wang Xiaoyu, Cui Xiangyang, Chen Deyun, Jiang Feng "Book Information Retrieval System Based On Deep-Web Data Integration" 2010 First International Conference on Pervasive Computing, Signal Processing and Applications IEEE.

[2] Ying Xie, Wanli Zuo, Fengling He, Ying Wang "Automatic Deep Web Query Results Extraction Based on Tag Trees" 2009 Second International Symposium on Computational Intelligence and Design IEEE.

[3] Hao liang, Wanli Zuo, Fei Ren, Chong Son "Accessing Deep Web Using Automatic Query Translation Technique" Fifth International Conference on Fuzzy Systems and Knowledge Discovery IEEE.

[4] He-Xiang Xu, Xiu-Lan Hao, Shu-Yun Wang, Yun-Fa Hu "A method of deep web classification" Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007 IEEE.

[5] Yiyao Lu, Hai He, Hongkun Zhao, Weiyi Meng "Annotating Structured Data of the Deep Web" State University of New York at Binghamton Binghamton, NY, 13902, U.S.A. 2007 IEEE.

[6] C. Abi Chahine, N. Chaignaud, JPh Kotowicz and JP Pe´cuchet "Context and Keyword Extraction in Plain Text using a Graph Representation" Learning Object Metadata draft standard document 2008 IEEE.

[7] Travis D. Breaux, Joel W. Reed from Department of Computer Science "Using Ontology in Hierarchical Information Clustering" Proceedings of the 38th Hawaii International Conference on System Sciences – 2005 IEEE.

[8] Govind Murari Upadhyay, Kanika Dhingra" Web Content Mining: Its Techniques and Uses" International Journal of Advanced Research in Computer Science and Software Volume 3,Issue 11, November 2013.

[9] Parul Gupta and A.K. Sharma"A Framework for Hierarchical Clustering Based Indexing in Search Engines" BVICAM's International Journal of Information Technology (BIJIT) Bharati Vidyapeeth's Institute of Computer Applications and Management(BVICAM), New Delhi.

[10] M.F., Porter, "An algorithm for suffix stripping", Program, vol. 14, pp. 130-137. 1980.

[11] Dhiraj Khurana, Satish Kumar (2012), "Web Crawler: A Review", International Journal of Computer Science & Management Studies, Vol. 12, Issue 01, ISSN: 2231 –5268.

[12] J.Uma Maheswari, Dr. G.R.Karpagam, "A Conceptual Framework For OntologyBasedInformationRetrieval"InternationalJournalof Engineering ScienceandTechnologyVol. 2(10), 2010, 5679-5688.