Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 54 (2015) 371 – 379

Procedia Computer Science

Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Semantic Retrieval of Relevant Sources for Large Scale Virtual Documents

R. Priyadarshini*, Latha Tamilselvan, T. Khuthbudin, S. Saravanan and S. Satish

Department of Information Technology, B. S. AbdurRahman University, India

Abstract

The term big data has come into use in recent years. It is used to refer to the ever-increasing amount of data that organizations are storing, processing and analyzing. An Interesting fact with bigdata is that it differ in Volume, Variety, Velocity characteristics which makes it difficult to process using the conventional Database Management System. Hence there is a need of schema less Management Systems even this will never be complete solution to bigdata analysis since the processing has no focus on the semantic information as they consider only the structural information. Content Management System like Wikipedia stores and links huge amount of documents and files. There is lack of semantic linking and analysis in such systems even though this kind of CMS uses clusters and distributed framework for storing big data. The retrieved references for a particular article are random and enormous. In order to reduce the number of references for a selected content there is a need for semantic matching. In this paper we propose framework which make use of the distributed parallel processing capability of Hadoop Distributed File System (HDFS) to perform semantic analysis over the volume of documents (bigdata) to find the best matched source document from the collection source documents for the same virtual document.

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Keywords: Virtual documents (VD); Source document; Hadoop file System (HDFS); DW Ranking algorithm; Top *K* algorithm.

1. Introduction

Big data is an evolving term that describes any voluminous amount of structured, semi – structured and unstructured data that has the potential to be mined for information. Big data can be characterized by 3Vs: the extreme volume of data, the wide variety of types of data and the velocity at which the data must be processed. Although big data doesn't refer to any specific quantity, the term is often used when speaking about terabytes and petabytes of data, much of which cannot be integrated easily traditional processing frameworks. Because big data takes too much time and costs too much money to load into a traditional relational database for analysis, new approaches for storing and analysing data have emerged that rely less on data schema and data quality. Instead, raw data with extended metadata is aggregated for which machine learning and artificial intelligence (AI) programs are used to search for repeated patterns. Big data analytics is often associated with cloud computing because the analysis of large data sets in real – time requires a platform like Hadoop to store large data sets across a distributed cluster. MapReduce is required to coordinate, combine and process data from multiple sources. Although the demand for big data analytics is high, there

*Corresponding author. Tel. +91 98 40352 046.
E-mail address: rpriyadarshini@bsauniv.ac.in

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)
doi:10.1016/j.procs.2015.06.043

is currently a shortage of data scientists and other analysts who have experience working with big data in a distributed, open source environment. There is a huge requirement of human resources to solve more of problems and research issues in Big Data. To sound the same we propose the new concept based on relevancy score and concept matching within ontology. In traditional DW ranking centrality score (Cs) is calculated based on number of forward link and backward link to that ontology. Then, the authority score (As) of a concept is computed which depends on reused entities between ontologies and the weight of these relationships based on the authority of the source ontology. The assumption behind this is that ontologies that are reused by other ontologies are more authoritative than others. Then in the query processing phase a candidate set for a *top-k* concept is selected from the ranked list of ontologies and then filtered based on frequency of the entities which are repeated in the source and virtual documents. To improve the efficiency of the retrieval we have proposed a new method "Modified Dual Walking Algorithm" based on Dual walking algorithm proposed by (Manuel Lama *et al.* 2012). In our work the centrality score is calculated based on threshold value of the new entity compared with the central entity in the ontology and authoritative score by reusability of neighbour ontologies in the ontology corpus. This algorithm shows significant changes in the metrics measured.

2. Overview

2.1 Motivation

The size of available data in internet is growing at an increasing rate. There are number of source links provided for an article identified in internet. These sources are mostly repeated or irrelevant content. Wikipedia is a Content Management System which consist of number of references and source documents. The references are categorized to Primary sources, Secondary sources and Tertiary sources in different clusters. Primary sources are nothing but the peer reviewed journals and conference articles and official white papers. These primary sources will be either in document format or in weblog format and some of these will also have digital object identifier. There are number of primary sources cited for a single article in wikipedia. The content in the article is cited with external citation with the primary sources. The user has to download each paper from URL and sites mentioned. The Secondary sources are in the form of official weblogs and E-magazines. Secondary sources are accounts at least one step removed from an event or body of primary – source material and may include an interpretation, analysis, or synthetic claims about the subject. Secondary sources may draw on primary sources and other secondary sources to create a general overview; or to make analytic or synthetic claims. Tertiary sources are publications such as encyclopedias or other compendia that sum up secondary and primary sources. For example, Wikipedia itself is a tertiary source. Many introductory textbooks may also be considered tertiary to the extent that they sum up multiple primary and secondary sources. So if the researchers or users want to know about any particular domain they might get confused because of lot of sources for a single virtual document. In Wikipedia the average sources for a document is 50–60 so the job of the user is very difficult to choose correct and more relevant source document. Hence we have made an attempt to reduce the source document by content management system. In this system the retrieval of source document is more accurate. On the other hand growing data in size affects the retrieval speed, so for that reason we have implement this system in apache hadoop for parallel processing of large data set.

In today's era there is greater incredibility of data in any sector of the business which may increase the possibility of business and retard the quality of retrieval of that information on future especially for growing data (bigdata). To overcome this issue the semantic analysis comes in to picture which make use of the metadata to learn about the large volume of data to be operated (Searched). Which intern helps the searching process to qualitatively identify outcome of the search. Hence this type of searching will never be the blind searching technique and it's a meaningful search called semantic analysis. The same concept applies for our problem the volume of source document to be searched may grow which retard the quality of retrieval. Hence the system should be provided with the knowledge of concept to be searching from the virtual document and the concept exists in each and every source document of that particular virtual document.

2.2 Contribution

The proposed system consist of a virtual document that is created with already available sources on internet. These sources URLs and source contents are stored in Hadoop. As number of users increases and number of documents

increases based on user requirement. This data which grows dynamically cannot be handled by traditional systems. The huge amount of data with source URLs and source contents will make searching and retrieval as difficult tasks. When this large volume of data is stored and analyzed the source links and references retrieved will be huge in number which again creates a major issue. The huge number of references in Wikipedia is also a drawback in encyclopedia like content management system. The references are also not semantically matched. The proposed system overcomes the above mentioned issue.

3. Related Work

Lee *et al.,*[1] in this paper they used HadUp tailored hadoop architecture which uses duplication based snapshot differential algorithm which measures the duplication ration between the old and new version of data sets. The experiment conclude that that the computation is not required on whole large scale data rather it can be done only on the updated data(snaps) alone (i.e.) eg: Wikipedia English article-2000 and 2012, found duplication ratio of 90% which means that that 10% is updated in Wikipedia and that can only be analysed for comparison. Aidan *et al.,*[2] this paper includes semantic algorithm in which entity consolidation is taken thus Using based line approach for entity consolidation in which data inconsistency occurred while extension of entities done. Zhang[3] in his study he systematically introduced main features, principles, properties in MVC (Modal, View, Controller) in a CMS which is exclusively design for the native people in china to use. It provides an efficient tool for the web page maker. This CMS separates forms from the web page content, so that it becomes very easy for the search engine to index. This CMS is applicable only for the native china mainland and advanced application is left for research. The database used was not scalable. Ismail and Joy[4] proposed an application to identify the similarities in a domain using semantic approaches. This approach focused on the existing approach for describing CMS and how metadata capture the pedagogic information which can be applied to the semantic information stored in CMS. Novel approach to capture relationship between tagged metadata for learning object stored from repository has been clearly described. But origin of the source content is not mentioned. Butt *et al.,*[5] this paper they introduce DWRank, a two-staged bi-directional graph walk ranking algorithm for concepts in ontologies. DWRank characterizes two features of a concept in an ontology to determine its rank in a corpus, the centrality of the concept to the ontology within which it is defined (HubScore) and the authoritativeness of the ontology where it is defined (AuthorityScore). It then uses a Learning to Rank approach to learn the feature for the two ranking strategies in DWRank. They compare DWRank with state-of-the-art ontology ranking models and traditional information retrieval algorithms. This evaluation shows that DWRank significantly outperforms the best ranking models on a benchmark ontology collection for the majority of the sample queries defined in the benchmark. In addition, they compare the effectiveness of the HubScore part of our algorithm with the state-of-the-art ranking model to determine a concept centrality and show the improved performance of DWRank in this aspect. Finally, they evaluate the effectiveness of the FindRel part of the AuthorityScore method in DWRank to find missing inter – ontology links and present a graph – based analysis of the ontology corpus that shows the increased connectivity of the ontology corpus after extraction of the implicit Inter – ontology links with FindRel. Shaban[6], in this paper, semantic understanding based approach to cluster documents is presented. The approach is based on semantic notions to represent text, and to measure similarity between text documents. The representation scheme reflects existing relations between concepts and facilitates accurate similarity measurements that result in better mining performance. They tested the system against different standard clustering techniques and different data sets. The semantic approach has enabled more effective document clustering than what conventional techniques would provide. Latha[7], in this study is proposed to reinvent the document based CMS in cloud. The proposed document based CMS varies from these traditional CMS in architecture in storage and control flow. Source URLs and content markings are indexed and mirrored. In this study the URLs and contend marked in the source websites are stored in the database. The source content and URLs are stored separately, the marked content in the user blog is highlighted and semantically processed with a NLP tool the marked content is semantically matched with concept matching and named entity recognition technique. The semantically annotated content are located in the stored websites and matched with the original source websites. This paper comprised the initial work of implementing annotation technique and location of the marked content in the document. The results were evaluated with metrics like recall and precision techniques. From the above findings of existing work, it is clear that semantic analysis of any document or system yields effective

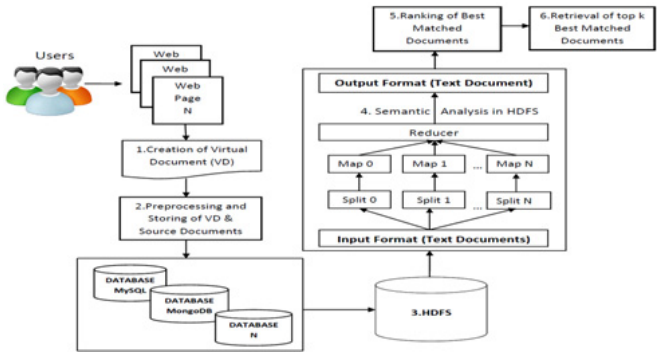

Fig. 1. Rendering system for semantic analysis of virtual documents.

results. There is no technique based on semantic analysis of documents over the distributed parallel environment like hadoop. In our proposed work semantic analysis and ranking of centrality concept of the document will be done by centrality and authoritative scores of document in HDFS. Further results are improvised by top k filtering of ranked documents based on frequency of the relevance entities of the ranked document.

3.1 Background work

As a background work before working with hadoop the similar kind of work is deployed cloud environment. A weblog is created with the features of locating exact source which is retrieved dynamically from web. The dynamic software application is bundled in to image and then deployed in the private cloud environment. In existing system the search was done only based on annotated word but in proposed system the search is done concept wise meaningfully using Machine Learning techniques. The dynamic retrieval will be done using meta heuristic techniques and performance is evaluated. The rendering engine is used to render the semantic based RDF/XML document. The documents in the rendering engine are clustered based on the latest clustering techniques such as agglomerative clustering. In this study the TopK algorithm is used based on Dual walking.

4. System Model

4.1 System overview

The proposed framework consists of following 6 main modules such as 1.Content Marker Creation and Selection of Virtual Documents 2. Pre-processing and Storing of VD and Source Document 3. Storing Documents to HDFS 4. Semantic Analysis in HDFS using Ontology 5. Ranking the Best Matched Documents 6. Retrieval of top k Best Matched Documents. Figure 1 depict the overall system architecture where user first creates virtual document by referring multiple source pages and then the complete virtual document with all of it source web pages are stored in the database and the documents from legacy databases are moved to the HDFS, the multiple database simple implies that it's possible to move data from variety of legacy database to Hadoop file system where semantic analysis is done using DW Ranking algorithm to rank the best matched source documents for the same virtual document. And the resulting documents are filtered using Top K algorithm based on the frequency of the entities in the source document.

4.2 Methodology

4.2.1 Content marker – creation and selection of virtual documents

The Virtual documents are created with the help of the user interface and stored in the legacy databases with the collection of its source documents from different web pages. Virtual documents are created with one or more concepts with the intention of the requirement by the user. The content from any source document can be copied

Table 1. Triple statement for above RDF response document.

Sl. no	Subject	Predicate	Object
1.	http://rdf.alchemyapi.com/rdf/v1/r/response.rdf#d2f6a39ad8 6a49226da7ed20e5da2fb874c73dda3-ge_ 1	http://rdf.alchemyapi.com/rdf/v1/s/aapi-schema# Relevance	"0.840039"
2.	http://rdf.alchemyapi.com/rdf/v1/r/response.rdf#d2f6a39ad8 6a49226da7ed20e5da2fb874c73dda3-ge_ 1	http://rdf.alchemyapi.com/rdf/v1/s/aapi-schema# Name	"Hadoop"

and used in the Virtual Document. The user interface for storing and organizing the content is called content marker. In the content marker the content can be selected or marked for retrieving the stored source documents. And the need for storing the source document is to find the sources of the each and every concept of virtual document for future reference.

4.2.2 Pre-processing and Storing of VD and source document

The source documents are pre-processed before they are stored in to the database. This pre-processing includes the simple functionality like removing the html tags, XML tags and scripting statements from the source document. The virtual documents are not pre-processed since they are created in the user interface, pre-processing is applicable to virtual document when they are in html document form. The source documents should also be pre-processed during the VD creation. The user will provide only the URL's of the source document which is a html document, so to convert it into text document this step is carried out on these source documents.

4.2.3 Storing documents to HDFS – single node and multi node

The collection of source and virtual documents from the legacy databases are transferred to the Hadoop Distributed File System (HDFS) with the help of the user interface and Plugins by specifying its database name. This clearly portraits that HDFS is not only suitable for growing bigdata but it can also suitable for the existing traditional databases for the same processing done on the bigdata. Hadoop is a framework written in Java for running applications on large clusters of commodity hardware and incorporates features similar to those of the Google File System (GFS) and of the MapReduce computing paradigm. Hadoop provides high throughput access to application data and is suitable for applications that have large data sets. While looking inside the hadoop the daemon processes working under it are: A *Data Node* stores data in the Hadoop File System. A functional file system has more than one Data Node, with the data replicated across them. The *Name Node* is the centerpiece of an HDFS file system. It keeps the directory of all files in the file system, and tracks where across the cluster the file data is kept. It does not store the data of these file itself. The *Job tracker* is the service within hadoop that farms out MapReduce to specific nodes in the cluster, ideally the nodes that have the data, or at least are in the same rack. A *Task Tracker* is a node in the cluster that accepts tasks – Map, Reduce and Shuffle operations – from a Job Tracker. *Single Node* configuration of this HDFS all these daemon processes operates in single local machine. Hence we configured our proposed work in single node environment. The semantic analysis starts with the formation of ontology for the selected virtual document based on the concept and keyword relevancies using the NLP API namely AlchemyAPI which ends its process by generating the RDF document output with the entities and its corresponding relevancies which serve the base attribute to compute the Centrality score of the document. The same procedure is applied for source documents of the selected virtual document for calculation of authoritative score. Table 1 shows the response document derived from the Alchemy API. And those RDF statements are given to protege to get intermediate result which is ontology tree of semantic analysis.

4.2.4 Ranking the best matched documents

As mentioned in section 3.1 the two scores will be calculated for the document at runtime Centrality score (Cs), Authoritative score (As). The score pertaining to virtual document relevancy called as the centrality score (Cs) and the relevancy score of the source documents of that particular virtual document called as authoritative score (As).

4.3 Mathematical model

4.3.1 Centrality score: The centrality of a concept within ontology

It is a measure of the concept match and relevancy to other concepts within the ontology itself called Centrality Score (Cs) the threshold value of it is suppressed to 1. The entity which gets the centrality score between 0.1 to 0.5 is expressed as the loosely coupled (lc) entities and which get between 0.6 to are tightly coupled (tc) entities For i ontologies the centrality score given as follows:

$$\sigma_{(lc,tc)} = \frac{1}{N_e}\sum_{i=1}^{Ne} Cs_e(lc,tc) - \mu(lc,tc) \tag{1}$$

$$Cs(lc,tc) = \frac{Cs(lc,tc) - \mu(lc,tc)}{\sigma_{(lc,tc)}} \tag{2}$$

Input: A finite set of virtual document $V_1, V_2, V_3 \ldots V_n$
Output: Centrality Score based on Classification of entities and Centrality concept Cs
Assumption: Let Threshold value of Centrality Score is assumed to be 0 to 1
Variables: Let lc be loosely coupled entities, tc be tightly coupled entities, N # of virtual documents, $\sigma_{(lc,tc)}$ be the mean of the centrality score , $Cs(lc,tc)$ Standardized centrality score, Kw Retrieved entities, Rs Relevancy score, N_f # of occurrence of the each entity in the document.

```
1   for(i=0; i<N; i++) //VD in Alchemy API
2   {   Kw ⟵ Retrieve entities from virtual documents
4       Rs ⟵ Retrieve relevancy of each entity
5       Nf ⟵ # of occurrence of the each entity
6   }
7   for each Kw //Calculation of Centrality Score
8   {
9       Rs      Retrieve relevancy of each entity
10      if(0.1 ≤ Rs > 0.5)
11       Rs = lc    Map and assign the entity to loosely coupled entity
12      else
13      Rs = tc    Map and assign the entity to tightly coupled entity}
14
15  for each Kw ///Calculation of Mean of Centrality Score
16      σ(lc,tc)+ ⟵ ((Cs - μ(lc,tc))/ No. of. Kw)
17      Cs(lc,tc) ⟵ Cs(lc,tc) - μ(lc,tc) / σ(lc,tc) //Calculation of
Standardized of Centrality Score
```

Algorithm 1. Algorithm for centrality score calculation.

Table 2. Centrality score & authoritative score.

Sl no.	No of virtual documents	Centrality score	No of source documents	Authoritative score
1	1	0.7589	4	0.6235
				0.5635
				0.5236
				0.4231
2	1	0.7255	4	0.7568
				0.6253
				0.5369
				0.3256
3	1	0.6325	4	0.9621
				0.8625
				0.7523

4.3.2 Authoritative score: The authoritativeness of a concept

It is a measure of a concept which is computed depends on reused entities between ontologies and the weight of these relationships between them, based on which the authority of the source ontology is being measured. The assumption behind this is that ontologies that are reused by other ontologies are more authoritative than others the threshold value of it also suppressed to 1. The entity which gets the authoritative score between 0.1 to 0.5 is expressed as the *reused entities (R)* and which get between 0.6 to 1.0 are *non-reused entities (R′)*. It is a measure of a concept which is computed depends on reused entities between ontologies and the weight of these relationships between them, based on which the authority of the source ontology is being measured. The entity which gets the authoritative score between 0.1 to 0.5 is expressed as the *reused entities (R)* and which get between 0.6 to 1.0 are *non-reused entities (R′)*. For i ontologies the authoritative score given as follows:

$$\sigma(R, R^1) = \frac{1}{N_e} \sum_{i=1}^{Ne} As_e(R, R^1) - \mu(R, R^1) \tag{3}$$

$$As(R, R^1) = \frac{As(R, R^1) - \mu(R, R^1)}{\sigma_{(R,R^1)}} \tag{4}$$

4.4 Retrieval of top k best matched documents

As mentioned in section 3.1 the two scores will be calculated for the document at runtime Centrality score (Cs), Authoritative score (As). Based on which the source documents are ranked for the concept of search in the virtual document.

Then the ranked results are prioritized based on the frequency relevance entities in the ranked source document. The more the frequency of the relevance entities in the source document will be given more priority. The relevance entities imply that the entities occurring in the virtual document which are occurring with more frequency in the source document are the more relevant entities.

Input: A finite set of virtual document $S_1, S_2, S_3 \ldots S_n$
Output: Classification of Source Documents based on Authoritative concept As.
Assumption: Let Threshold value of Authoritative Score is assumed to be 0 to 1.
Variables: Let **R** be Repeated entities, R' be Non Repeated entities, N = # of Source documents, $\sigma_{(R,R')}$ be the mean of the Authoritative score, $As(R, R')$ Authoritative score, Kw Retrieved entities, Rs
Relevancy score, N_f = # of occurrence of the each entity in the document.

```
for(i=0; i<N; i++) //VD in Alchemy API
{
        Kw  ←  Retrieve entities from source documents
        Rs  ←  Retrieve relevancy of each entity
        N_f ←  # of occurrence of the each entity
}
//Calculation of Authoritative Score
for each Kw
{
        Rs  ←  Retrieve relevancy of each entity
        if(0.1 ≤ Rs > 0.5)
                R  ←  Map and assign the entity to Repeated entity
        else
                R' ←  Map and assign the entity to Non Repeated entity
}
//Calculation of Mean of Authoritative Score
for each Kw
        σ_(R,R') ←  ((As - μ_(R,R')) / No. of. Kw)
//Calculation of Standized of Centrality Score
        As(R,R') ←  As(R,R') - μ_... / σ_(R,R')
```

Algorithm 2. Algorithm for authoritative score calculation.

Input:	Virtual Document be the VD and A finite set of Ranked Source document $S_1,S_2,S_3,\ldots,S_n$
Output:	Prioritization of Source document based on relevancy of repeated # of entities.
Assumption:	Let as assume $S_1,S_2,S_3\ldots S_n$ be the ranked source document from modified DW ranking algorithm.
Variables:	E_{Rel} be the intersection of entities in VD and Best Matched Documents, $\varepsilon_{Rel[1..n]}$ be the sorted list of documents based on relevancies of the entities, $\varepsilon_{Rel[MAX]}$ be the Most Best Matched Document from the sorted list of document.

Algorithm 3. Algorithm for top *k* prioritization of ranked documents.

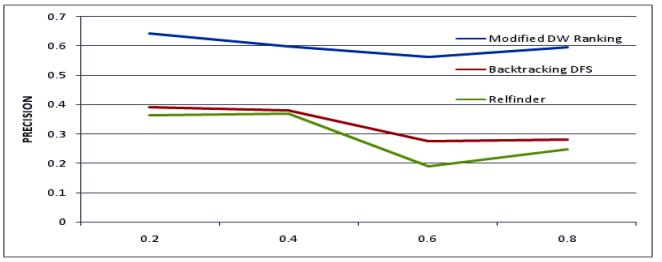

Fig. 2. Precision graph.

Table 3. Comparison of modified DW ranking with DFS backtracking technique and relfinder.

Algorithm	Measures	Set 1	Set 2	Set 3	Set 4
Relfinder	Precision	0.3626	0.3692	0.1897	0.2464
	Recall	0.2654	0.3708	0.2762	0.5142
	F-Measure	0.2534	0.3700	0.2249	0.3331
DFS Backtracking Technique	Precision	0.3900	0.3779	0.2754	0.2800
	Recall	0.7444	0.7269	0.6695	0.6808
	F-Measure	0.5119	0.4996	0.3903	0.3968
Proposed Modified DW Ranking Algorithm	Precision	0.6433	0.6003	0.5632	0.5963
	Recall	0.9323	0.9231	0.8562	0.8769
	F-Measure	0.761295	0.7275	0.679459	0.709877

4.5 Implementation issues

The proposed system involves the working of both the Relational Database Management system and NOSQL like MongoDB. The Performance of these systems vary drastically. The documents are extracted from the source URL through Google API and Wiki API automatically to the system. The System documents can be increased up to 2 TB using multimode hadoop environment with the semantic analysis performed in the Hadoop based cloud environment.

4.6 Performance analysis

The graph is plotted with Recall and Precision and is plotted for 5000 documents with four sets of documents and it is compared with the algorithms implemented in hadoop with parallel processing. Table 3 shows the comparison of Relfinder and DFS Backtracking technique which uses Linking Object. Our modified work is with inking URLs which uses the same logical flow as above two methodologies. Our proposed work shows an obvious increase in precision, recall and F-Measure as shown in the graph, Fig. 2.

5. Conclusion & Future Works

In this paper, first user creates virtual document by referring multiple source web pages and then the complete virtual document with all of it source web pages are stored in the database and the documents from legacy databases are moved to the HDFS, where semantic analysis is done using DW Ranking algorithm to rank the best matched

source documents for the same virtual document based on the centrality an authoritative scores of document. And the resulting ranked documents are filtered using Top K algorithm based on the frequency of the relevance entities in the source document. In future the threshold for number of source documents for creation of virtual document will be increased by testing more number of documents and the same concept will be used for Big Data and performances will be analysed.

References

[1] Daewoo Lee, Jin-Soo Kim and SeungryoulMaenga, Large – Scale Incremental Processing with MapReduce, *FGCS*, vol. 36, pp. 66–79, September (2013).
[2] Aidan Hogan, Antoine Zimmermann, JurgenUmbrich, Axel Polleres and Stefan Decker, Scalable and Distributed Methods for Entity Matching, Consolidation and Disambiguation Over Linked Data Corpora, *WebSem*, vol. 10, pp. 224–313, November (2011).
[3] Y. Zhang, An Excellent Web Content Management System, In *Conference on Multimedia Technology*, Hangzhou, pp. 3305–3307, June 26–28, (2011).
[4] A. Ismail and M. Joy, Semantic Searches for Extracting Similarities in a Content Management System, In *Conference on Semantic Technology and Information Retrieval*, Putrajaya, pp. 113–118, June 28–29, (2011).
[5] AnilaSahar Butt, Armin Haller and LexingXie, Relationship – Based Top-K Concept Retrieval for Ontology Search, *SemWebIOS*, vol. 6, pp. 485–502, (2014).
[6] B. Khaled and Shaban, A Semantic Approach for Document Clustering, *JOS*, vol. 4, no. 5, pp. 391–404, (2013).
[7] R. Priyadarshini and LathaTamilselvan, Document Based Semantics CMS in Cloud, *ITJ*, vol. 13, pp. 217–230, (2014).