Concept Based Document Management in Cloud Storage

[1]M.R.Sumalatha, [2]E.Pugazhendi, [3]D.J.Archana
[1, 2 & 3] Department of Information Technology
Anna University
Chennai, Tamil Nadu, India
Email :{ sumalatha.ramachandran[1], pugazh.cse[2], dj.archana[3]}@gmail.com

ABSTRACT – Cloud computing is one of the most useful environment that provides various information services in which required information can be retrieved through many web-based tools and applications. Now the new surge of interest in cloud computing is accompanied with the exponential growth of data sizes .There is a need to find the desired content quickly and efficiently by simply consulting the index. Thus there arises a question of how to effectively process these immense data sets is becoming increasingly urgent. Our existing system is searching the content through ontology in cloud which practically suffers from maintaining a consistent logic for the input documents given and taking this advantage into consideration we have brought in new concepts called Document Retrieval Algorithm. In this paper we discuss how effectively and efficiently information can be retrieved from cloud taking into account their storage space too, where we store the metadata of file in cloud and not the entire file which holds a lot of space in distributed environment. Thus we bring in the concept of Named Entity Recognition and Universal Word List with Term frequency, which maximizes the information retrieval more effective and efficient and also to bridge the gap between the semantic web and the users which reduces the complexity met by them in information retrieval.

Keywords – Concept Extraction, Named entity recognition pronoun, Cloud storage, Document retrieval

I. INTRODUCTION

Cloud computing is a platform that possess pay-per-use paradigm for providing the required services over the web in a scalable manner. Some companies that deliver such services are Google, Amazon, Microsoft, etc. Each component in cloud market space possesses both the frontend and backend platform .The communication between the components is via message queues. Retrieval of any/needed information in cloud is being implemented in many ways. One such method that makes information retrieval effective is Ontology. Semantic Web is a best environment to provide meaningful information and make search meaningful and effective. Ontology is considered as one of the main components of the Semantic Web which is used to represent, acquire and utilize knowledge to help machines understand the meaning of content of different web resources that increase the opportunities of automated information processing. Ontology provide a well defined vocabulary that define different heterogeneous data resources including resources, semi-structure and unstructured files enabling a new generation of applications especially that merge the idea of Semantic Web and Cloud Computing. There are several search engines that are based on ontology that are still accessible. These systems accept keyword queries and return matched concepts or ontology's as output. The results are returned as RDF description or basic metadata both of which cannot help users to determine whether the results returned satisfy their needs. Some of the issues are: security issues that arise as a result of storing data at unauthorised host, ability to gain valid information, time consumption and data replication functionalities.

Various techniques like Construction of virtual documents, ranking concepts, ranking ontology and generating query relevant snippets were followed to help people to find the ontology's that satisfy their needs. The ontology's are recommended based on top-ranking concepts. Even though by following these model the search is made efficient with a less time consumption it has been surveyed recently that around 50% of the people are not satisfied with the end result and they avoid using it due to its complexity. With this in mind, in this paper we try to probe new techniques to evolve a new search and storage scheme that can efficiently store and search information across clouds. For this matter, an efficient cloud storage and management system, not considered so far, can be the pivotal for competent management system. Doing so will reduce the intricacy faced by the user in information redemption and thus duplication data can be removed. Note that this model is not against the ontology concept it's just a refinement to make it justifiable to the users. The result thus obtained will be based on the best scrutiny of information. Cloud environment provides a nonpareil support for making this model a lucrative one which will be preferred by every user.

ISBN:978-1-4799-1024-3/13/$31.00 ©2013 IEEE

II. RELATED WORK

Based on the numerous study made on this domain multifarious results were catalogued that helps in making the working model more proficient. As discussed by Danushka Bollegala in [2] though page counts and snippets were used for retrieving the semantic similarity between words in web engine the blunder was when phrase or document is given as input the idea failed. As discussed early the use of virtual documents in the existing system, it has some hitch revealed very clearly by Yasufumi Takama in [7]. Though virtual documents have grown out of a need for interactivity and individualization of documents, particularly on the Web, The emergence of virtual documents reveals some very interesting information retrieval problems like, Search: How do you search for a virtual document? Reference: How do users cite virtual document? Revisiting: Users have an expectation that documents found one day will be available on a subsequent search. The notion of bookmark does not apply to virtual documents in its normal simplistic way. The refinement of ontology based system is based on the facts described by Haytham Feel in [1], the paper proves to give better results in terms of response time and throughput using ontology, but it takes only keywords into account and not the file.

Other flaws discussed are fault tolerance, unnecessary file content retrieval, single point of failure, no semantic web search and single co-ordinator exists. Sanjay Ghemawat stated in [12] though Google file system has successfully met all storage needs and deployed within Google as the storage platform, it follows a single master slave system so suffers from single point failure and even write operations faces many challenges in real time. A novel based service retrieval algorithm was proposed by Pierluigi Plebani in [3], but even then care has to be taken in concern with the quality of services in order to extend the registry and the tuning phase requires more refinement. Various flaws that exists in the existing system is analysed effectively and thus by considering the issues discussed above the system proposed in this paper is designed in a way that it overwhelms all the risks in a more prominent way and thus being pragmatic to users.

III. PROPOSED WORK

This section presents the architecture of the working model. The system components, various modules present, input and output for each module, their operation, and the fortuity obtained is as follow. The fig.(1) specifies the working model of the proposed architecture.

3.1 System Architecture

From this diagram we can be able to identify how the information is stored and retrieved effectively and efficiently. This is consummated by undergoing multifarious modules and integrating them into one.

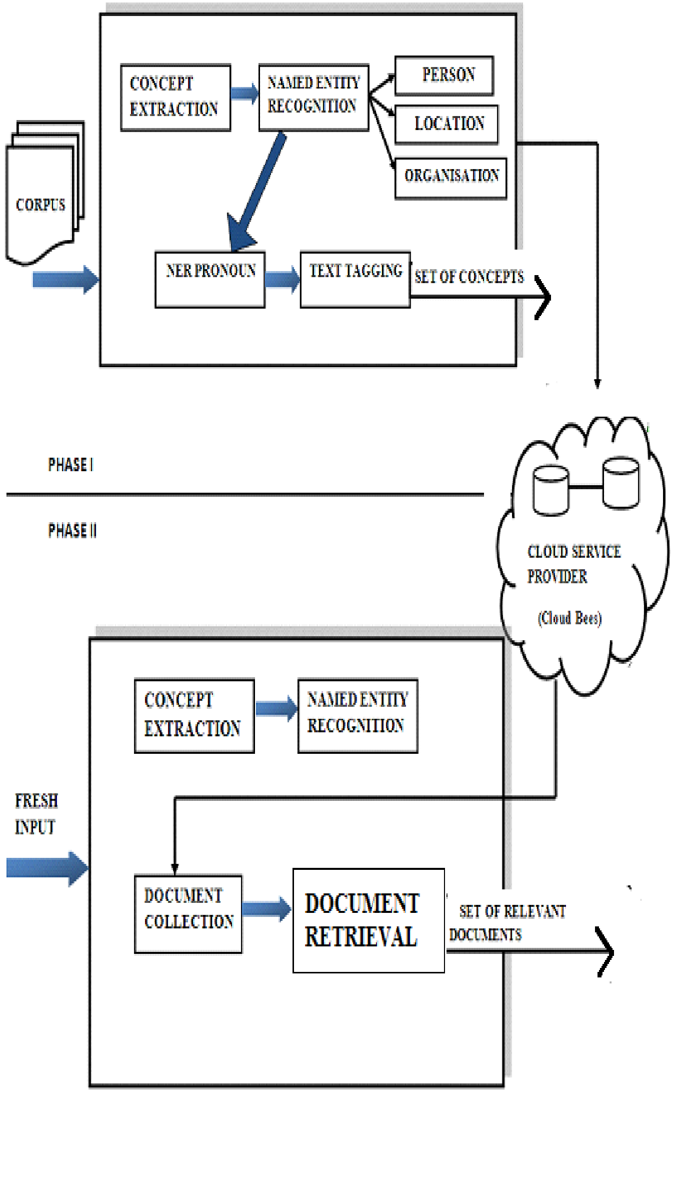

Figure 1. Architecture of working model

Module 1: Concept extraction

The basic requisite to begin with this effective management system is that the files are to be uploaded such that it can be processed in a prominent manner. So, booting up of files that are to be processed into the system becomes the inchoate step. Processing of files that are successfully uploaded is carried out in various steps. The first method to be implemented is term frequency which undergoes four different phases to produce more refined result. First phase tokenization that is very essential for safeguarding various sensitive data and other types of personal information. Though there exists many encryption techniques for safeguarding data tokenization becomes a unique approach that is followed globally. It actually breaks a stream of text in the input file into words, phrases, symbols known as a token which becomes input for further processing of data. After the successful consummation of tokenization the next phase in queue is stemming which actually reduces the words into stems.

This process actually follows few rules that are written generally in an algorithmic form. It reduces English words that exist in the document into common stems. For example if a document contains the words look, looking, looked, looks etc, all these words are reduced into a common stem look. In few cases it just removes the suffix 'ed', 'ly', 'ing' from the words and forms the corresponding stem. With the fruitful creation of stems the next phase data cleaning is to be executed.

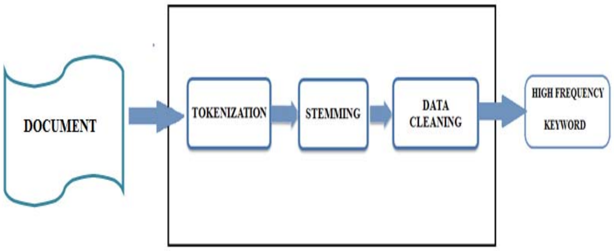

Figure 2.Phases in concept extraction

This phase concentrates mainly on removal of stop word present in output of the before phase. The main motive for executing data cleaning is that, it is very obvious that incorrect data can lead to fake results and thus counterfactual data are to be removed so that accessibility to reliable data can be made possible. Two main processes in data cleaning are harmonization and standardization. Last phase that is yet to be completed is high threshold frequency that filters the keyword with high frequency. Fig (2) depicts the phases of concept extraction. From the diagram it is very clear that whenever a new document is given as input to the system, it passes through every phase of term frequency as discussed earlier. As a result the refined output is obtained from which the words with high threshold value is noted.

Module 2: Named entity recognition

NER is a subtask of information extraction that seeks to locate and classify elements in text into predefined categories such as person names, organisations, locations, expression of times, quantities etc. The output of this consists of names as well as pronouns list. Stanford NLP tool is used to perform this type of extraction. This tool is also known as CRF Classifier which is a Java implementation of a Named Entity Recognizer. NER labels sequences of words in a text which are the names of things, such as person and company names, etc. This software provides a general implementation of linear chain Conditional Random Field (CRF) sequence models, coupled with well-engineered feature extractors for Named Entity Recognition. The main categories under which the contents are tagged includes person, organization, location which are the named entity recognizers. The files which are processed with the help of this tool will provide a resultant text file that contains the tagged output of person, organization, location that exists in the original text file. This gives the clear idea of what the document talks about which makes the document retrieval easier. Now we are done with NER. To enhance the performance even more we move into the concept called Named Entity Recognition Pronouns.

Module 3: NER pronoun

Named Entity Recognition pronoun takes into account the presence of pronouns in the document. A pronoun is a word or forms that substitute for a noun or noun phrase in a document. No pronouns were excluded from the evaluation. The output of the before phase that is NER contains only the names that are tagged according to the category that is been specified in the tagger class selected in the Stanford tool. Now the pronouns that exist in the document which debits the nouns in the document are to be tagged and included along with the tagged nouns. For this purpose the before document is processed such that the pronouns are tagged according to the category they belong to, for example he, she, him, his also fall under the category of person. Finally the text file with the entire tagged text will be given to the next phase. The count of occurrence of the noun will give the clear idea of the document. After the execution of this module the output will be saved a in a text file.

Module 4: Cloud deployment

The corpus that as undergone all these steps are to be deployed in cloud. Only after the successful deployment of files in cloud environment the information can be accessed globally. This helps people to use the system properly in a better way with a reduced complexity level in their side. For our work we utilized CLOUD BEES - This takes an entire development, build and deploy cycle of a Java web application to the Cloud Bees PaaS. It emulates a typical development scenario where a developer builds a database-backed application in his environment, has to hand the application to QA, deploys the application in production and monitors it as well. Along the way, the developer moves to using a MySQL database (to be in sync with the production environment) and sets up continuous integration jobs to monitor his builds. Administrator has the dexterity to add any number of employees who work together for the successful deployment of information. Once after adding up of members, with the identification provided for each they are allowed to view the DB content that is been stored in cloud. So information can be accessed globally by each member.

Module 5: Document retrieval

Algorithm 1 Document retrieval

```
Step 1: Let Cd= {c1, c2...cn} denotes set of concepts identified in document d.

Step 2: Let Sci be a set of all document objects associated with concept ci.

Step 3: Let Rd denotes set of related documents for document.

        Now Rd can be populated as

                For $i \leftarrow 0$ to $n-1$

                        For $j \leftarrow i+1$ to n

                                $Rd \leftarrow Ri \cap Rj$.

Step 4: Let Pm be on document object. It can be associated with one or more concepts.

Step 5: Let W (Pm) denotes the weight of document object Pm associated with concept ci.

        W (Pm) is computed based on the relative frequency of ci in the document.
```

Once after the lucrative deployment of data in cloud all the operations are completed successfully, the output will be some specific words of preferred threshold. Thus creating field called “Description” in Metadata and storing the threshold words in them intern in cloud database too. The result thus obtained will be stored and the documents matching each concept will be retrieved from the cloud database. The algorithm proposed for this purpose is document retrieval algorithm that is explained in algorithm 1. Then after retrieving the relevant documents for each concept, next intersection has to be performed. This intersection is performed to get the common documents between each concept. So by doing intersection and filtration the set of relevant document will be obtained successfully. Thus the complexity present in the before system can be reduced and make a easy use of the system.

3.2 Advantages of proposed system

1. Effective and efficient Functioning.

2. No delay in Operation.

3. Comparatively better health issues.

4. In-time and proper search operation performed. .

3.3 DESIGN

The general design that is followed in this working model is that as discussed before the corpus are stored in cloud after the successful consummation of various processes. All these are stored in the database repository in the cloud environment. Cloud service provider in the backend helps in performing all these activities. After the lucrative completion of these steps we end up with the phase I that is designer side. Next phase fully concentrates only on the users and their approach towards the system. When the user gives the input to the system, the processing of that particular input file undergoes various modules such as term frequency, NER and NER pronoun. Once after getting the refined output for the entered file, the information stored in cloud is searched. Based on the documents hoarded in cloud the text intersection and filtering takes place. Keyword combination algorithm discussed earlier will be used in this stage for making intersection and filtering of keyword more gullible. As a result of this the most relevant set of documents are retrieved from the cloud database. The tool that is used for performing NER count is discussed in the next section.

3.3.1 CLOUDBEES

We are aware about various free and open source software that is available in web for meeting our requirement and making our job deludable. These tools are available at our door steps. Just clicking a single button makes it run in any environment. One such kind of open source software is Stanford NER tool.

As discussed in section 3.1.2 and 3.1.3 Stanford NER is used mainly for extracting the named entities from the given input text. Named entities fall into few categories such as persons, organizations, location, status etc. Stanford NER is included in various packages globally and its embedded tool Stanford Named Entity Recognizer processes the given text locally. This tool in particular is used for providing the natural languages set which takes English language text input and will provide their basic forms of words, parts of speech. It also mark ups the structure of sentences in terms of word dependencies and phrases. The main point to be noted is that it clearly debits which noun phrases refer to the same entities. With help of this feature it is possible to feed any number of input texts to the system. In spite of availability of various tools available this is chosen because of its unique features. The goal of using this tool is to help people to quickly and painlessly obtain the complete concept they required. This is possible because of its flexible and extensible feature.

3.3.2 CLOUD STORAGE

Cloud Storage is a enterprise storage where data is stored not only in user's computer, but hosted in third party storage area too. In our proposed system we have taken **Apache Subversion** as our cloud storage, where all the files are stored in a privately secured url. There are 2 modes of uploading file: (1) CloudBees SVN console (2) CloudBees webpage.

IV. PERFORMANCE ANALYSIS

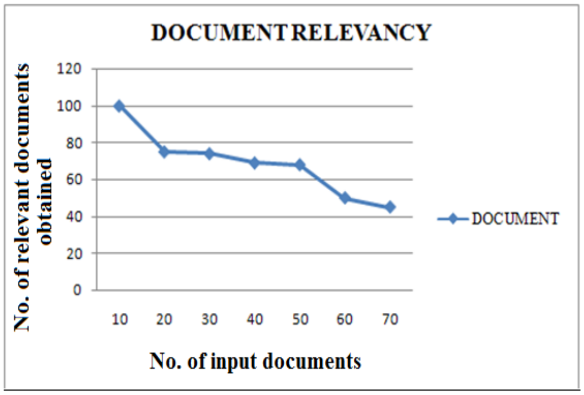

Figure 4.1 Average Case

Fig(4.1) Speaks about the average case scenario of relevant document retrieval where as the corpus size increases relevant document retrieval maintains to an average level.

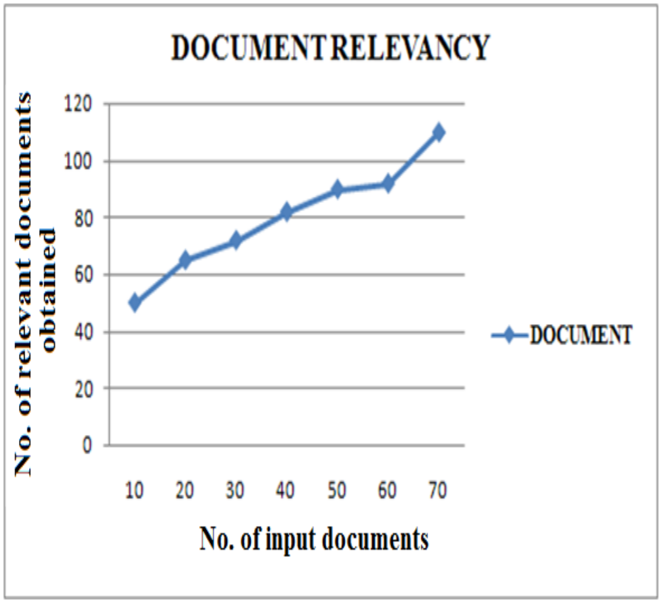

Figure 4.2 Best Case

Fig(4.2) depicts the best case scenario of relevant document retrieval where as the corpus size increases relevant document retrieval also increases.

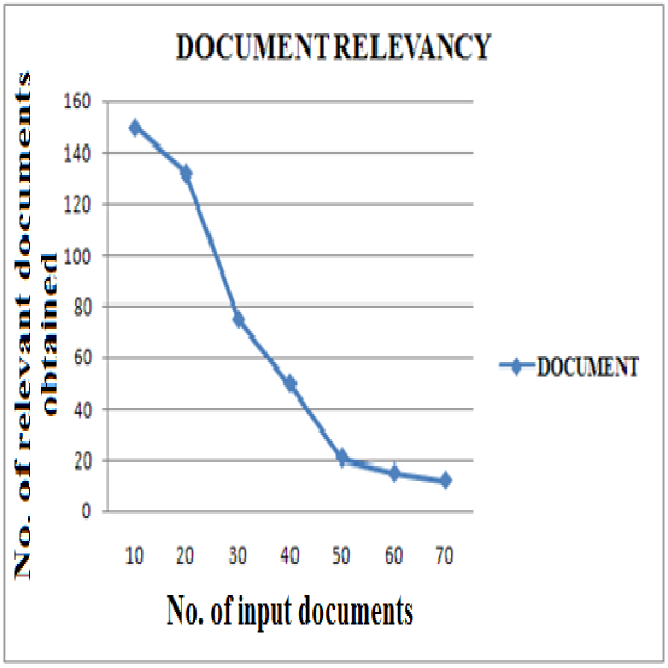

Figure 4.3 Worst Case

Fig(4.3) depicts the worst case scenario of relevant document retrieval where as the corpus size increases relevant document retrieval decreases vastly .

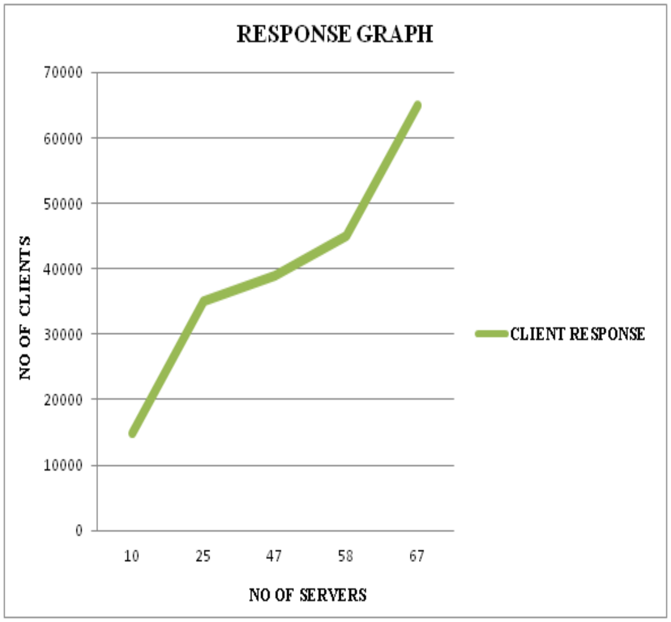

Figure 4.4 Response Graph

Fig(4.4) Speaks about response of client to "N" number of servers provided by the cloud platform showing increased availability of information needed. As the client request increases there will not be any degradation in the performance increasing their availability all the time.

4.1 COMPARISON WITH EXISTING SYSTEM

Comparison between existing and proposed system is done based on the number of relevant documents obtained as output. Taking existing system in hand , say for example , if we give 150 documents as input it fetches 224 relevant documents as output but in our proposed system ,if we give 150 documents as input it fetches 267 relevant documents as output by introducing Document Retrieval Algorithm.

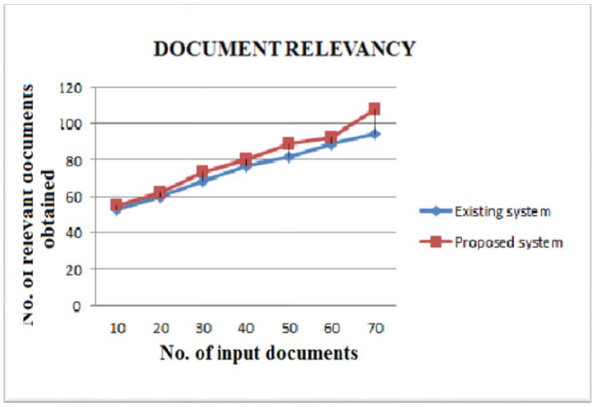

Figure 4.5 Comparison

Fig(4.5) depicts the comparison of proposed system with the existing system where the number of relevant documents obtained is compared with number of input documents.

V. CONCLUSION

Finally we can say that, an effective and efficient search can be made by following the phases that are been proposed in this paper. This method overcomes the complexity of the before system and thus make people to feel free to use this system. It ensures that the purpose of this system is been satisfied in a way that is paltry for the people. Semantic web is taken into consideration which increases the informational retrieval effectively and efficiently. As a part of which Term frequency and Named Entity Recognition pronoun implementation is completed successfully. The entire information is deployed in cloud fortuitously and the meta data is stored. The location of files is stored in a database. When a user makes the search in the system the keyword filtration and intersection is done based on this data stored and relevant set of documents are retrieved more prominently.

VI. FUTURE WORK

This work can be further enhanced by seeking images as input to the system and retrieving the document relating to that image or any relevant images with respect to the input. This can be done because irrespective of the content pictorial representation will be helpful in making people understand the concept. A more complex task of fetching videos and audios as input can be done and the retrievals can be based on these inputs. The main aim of doing this is to reduce the complexity in retrieving documents and help people in satisfying their need.

REFERENCES

[1] Haytham Feel and Mohamed Khafay, "Search content via Cloud Storage System", 2011.

[2] Danushka Bollegala, Yutaka Matsuo, And Mitsuru Ishizuka, "Web Search Engine-based Approach To Measure Semantic Similarity Between Words", 2011.

[3] Pierluigi Plebani and Barbara Pernici, Member, "URBE: Web Service Retrieval Based on Similarity Evaluation", 2009.

[4] Yajing Zhao, Jing Dong and Tu Peng, "Ontology Classification For Semantic-web-based Software Engineering", 2009.

[5] Yufei Li, Yuan Wang, and Xiaotao Huang, “A Relation-Based Search Engine in Semantic Web”, 2007.

[6] Rudi L. Cilibrasi and Paul M.B. Vita ́ny, “The Google Similarity Distance”, 2007.

[7] Yasufumi Takama, Member, IEEE, and Shunichi Hattori, “Mining Association Rules for Adaptive Search Engine Based on RDF Technology”, 2007.

[8] A. Kilgarriff, “Googleology Is Bad Science,” Computational Linguistics, vol. 33, pp. 147-151, 2007.

[9] Y. Qu, W. Hu, and G. Cheng, ”Constructing virtual documents for ontology matching”, 2006.

[10] M. Sahami and T. Heilman, “A Web-Based Kernel Function for Measuring the Similarity of Short Text Snippets,” Proc. 15th Int’l World Wide Web Conf., 2006.

[11] Hang Zhang, Wei Hu, and Yuzhong Qu, “Constructing Virtual Documents for Ontology Matching Using Map Reduce”, 2005.

[12] Sanjay Ghemawat, Howard Gobi off, and Shun-Tak LeungGoogle, “The Google File System”, 2005.

[13] H. Chen, M. Lin, and Y. Wei, “Novel Association Measures Using Web Search with Double Checking,” Proc. 21st Int’l Conf. Computational Linguistics and 44th Ann. Meeting of the Assoc. for Computational Linguistics (COLING/ACL ’06), pp. 1009-1016, 2006.