INTELLIGENT AND PERVASIVE ARCHIVING FRAMEWORK TO ENHANCE THE USABILITY OF THE ZERO-CLIENT-BASED CLOUD STORAGE SYSTEM

Keedong Yoo

Department of Management Information Systems, Dankook University,
Cheonan, Republic of Korea
kdyoo@dankook.ac.kr

ABSTRACT

The cloud storage-based zero client technology gains companies' interest because of its capabilities in secured and economic management of information resources. As the use of personal smart devices such as smart phones and pads in business increases, to cope with insufficient workability caused by limited size and computing capacity most of smart devices have, the necessity to apply the zero-client technology is being highlighted. However, from the viewpoint of usability, users point out a very serious problem in using cloud storage-based zero client system: difficulty in deciding and locating a proper directory to store documents. This paper proposes a method to enhance the usability of a zero-client-based cloud storage system by intelligently and pervasively archiving working documents according to automatically identified topic. Without user's direct definition of directory to store the document, the proposed ideas enable the documents to be automatically archived into the predefined directories. Based on the proposed ideas, more effective and efficient management of electronic documents can be achieved.

KEYWORDS

Intelligent archiving, Cloud storage, Zero-client, Automatic document summarization

1. INTRODUCTION

The zero-client technology, or the empty can-like PC technology, is an emerging ECM(enterprise content management) technology by integrating the VDI(virtual desktop infrastructure) into the cloud storage environment to securely manage and utilize companies' intellectual resources in a more efficient and pervasive manner. Comparing to the service through conventional thin-client technology, the zero-client technology can not only more securely manage documents by minimizing the amount of working documents stored in operator's personal workstation, but also more economically maintain computing systems by directly downloading required software patches and updates in a real time basis. As the needs to apply personal smart devices such as smart phones and pads widely used nowadays increase, the zero-client technology can play a very essential role in coping with insufficient workability caused by limited size and computing capacity most of smart devices have.

David C. Wyld et al. (Eds) : CST, ITCS, JSE, SIP, ARIA, DMS - 2014
pp. 381–389, 2014. © CS & IT-CSCP 2014 DOI : 10.5121/csit.2014.4135

The cloud storage, a model of networked corporate storage where data is stored in virtualized pools of storage, provides the Internet-based data storage as a service. One of the biggest merits of cloud storage is that users can access data in a cloud anytime and anywhere, using any types of network-enabled user devices [1]. Amazon Web Services S3 (http://aws.amazon.com/s3), Mosso (http://www.rackspacecloud.com), Wuala (http://www.wuala.com), Google Drive (http://drive.google.com), Dropbox (http://www.dropbox.com), uCloud (http://www.ucoud.com), and nDrive (http://ndrive.naver.com) are typical examples of corporate and personal cloud storage services. All of these services offer users transparent and simplified storage interfaces, hiding the details of the actual location and management of resources [2]. Once a document is stored in the cloud storage, a user can access and download the document anytime and anywhere under the condition that designated access right has been granted. Because of the advantages in storing and extracting information resources, more companies are implementing the online storage under the cloud storage environment.

While the cloud storage can deliver users various benefits, it also has technical limits in network security as well as in privacy [3]. In addition, from the viewpoint of usability, many users also point out a very serious problem in using cloud storage-based zero client environment, which is the difficulty in storing and retrieving documents. Since the directories in the cloud storage has been defined and structured by companies' decision, most of users are not accustomed to them. To store a working document in the cloud storage, a user has to decide a proper directory that exactly coincides with the contents of the document. Since the directories are naturally varied and the overall structure of directories is complicated, deciding a proper directory is not an easy work: sometimes a user can go astray by the confusion in deciding and locating target directory. Also, when a user tries to retrieve a document, he/she may spend not a little time to locate the file because too many directories exist. Since the directories are not defined and provided by him/herself, relatively much time to make a user be accustomed. Therefore, any automated assistance in concluding the target directory is indispensably needed by analysing contents of the given document with respect to directories in the cloud storage. Since any keywords or topics extracted from the document stand for the possible title of the directory under which the document must be stored, a user can easily complete his/her job to store and retrieve documents. In retrieving a document from the storage, more accurate and fast searching can be made because each document has been archived into the topic-based directory.

This research tries to enhance the usability of a zero-client-based cloud storage system by intelligently and pervasively archiving working documents according to automatically identified topic. To do so, this research suggests not only a framework to automatically extract the predefined directory-specific topic of a working document by applying an automatic document summarization technique, but also required sample codes to pervasively archive documents under the automatically determined directory.

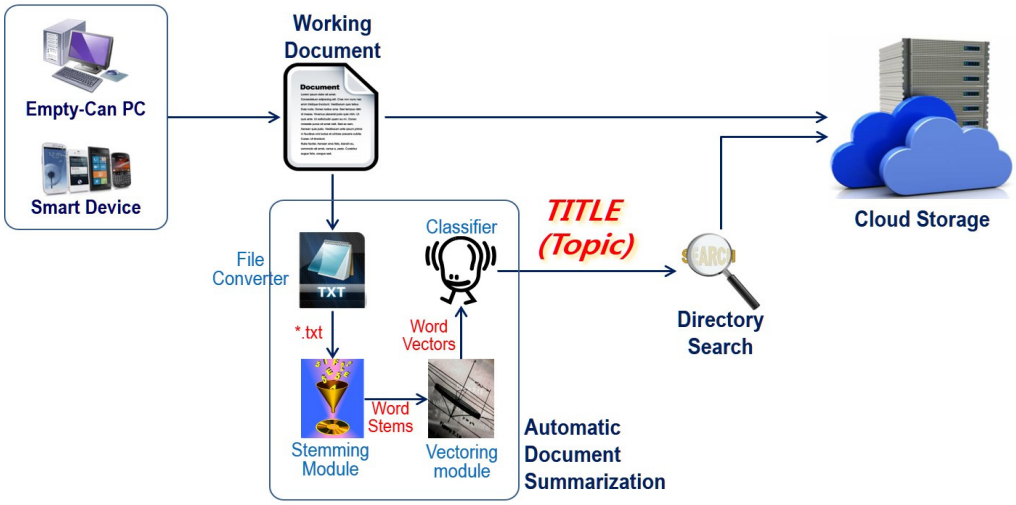

Figure 1. Framework for intelligent and pervasive archiving

2. FRAMEWORK FOR INTELLIGENT AND PERVASIVE ARCHIVING

Figure 1 shows the suggested framework for intelligent and pervasive archiving. Since the cloud storage plays the role of VDI-enabled database, pervasiveness in document storing and retrieving can be guaranteed. Intelligent archiving can be attained by two functionalities: one is automatic document summarization to automatically extract a title of the given document, and the other is automatic directory search to locate given document onto predefined directory according to the extracted title.

Once a working document is created by users using their empty-can PC or smart devices, it must be archived in the cloud storage-based corporate repository because users' terminals are not equipped with internal storages. To archive the working document intelligently, the title of the document must be automatically determined by analysing the words included in the document, and corresponding directory in the cloud storage must be automatically concluded also according to the topic or title of the document. Therefore, as a user finishes creating the document and tries saving it, the module for automatic document summarization initiates its function to extract the title of the working document according to the procedures as follows;

2.1. File format converting

The file format of the working document can be varied with the types of software used in creating the document. To guarantee the efficiency of analysis to extract the title of a given document, the file format must be normalized (or standardized) into analysable one in advance to the rest of procedures involve [4]. In this research, the file formats are designed to be converted into the '.txt' format to promote the readability of following modules.

2.2. Stemming

Once the document formats have been normalized, words in the document must be also normalized so that only stems of each word can be considered by separating inflectional and derivational morphemes from the root, the basic form of a word. For example, the root of the English verb form 'preprocessing' is 'process-'; the stem is 'pre-process-ing', which includes the

derivational affixes ‘pre-‘ and ‘-ing’, but not the present progressive suffix ‘-ing’. After stemming each word, non-necessary stems must be eliminated to promote the efficiency of analysis by setting lists of stop words which need to be filtered out ahead to further analysis.

2.3. Vectorizing

Based on the word stems from the phase of stemming, each stem must be vectorized to extract a document vector. In many cases, the TF/IDF (Term Frequency/Inverse Term Frequency) is usually used, and this study also apply it. TF/IDF is a statistical technique to assess relative importance of a word in a document. A high weight in TF/IDF is obtained by a high term frequency and a low document frequency of the term in the whole collection of documents; the weight therefore tends to filter out common terms. The word with the highest TF/IDF is deemed as the topic of a document.

2.4. Classifying

Resultant topic of a document can be identified by plotting the document vector onto a given vector spaces prepared by predefined category-based sample data. Therefore, to promote the accuracy of classification, the quality of sample data is very crucial, and therefore a corpus which is a collection of predefined categories with sufficient number of example documents must be formally examined. In conventional text mining area, a classifier is based on various algorithms such as SVM (Support Vector Machine), Naïve Bayes, and k-NN(Nearest Neighbors), etc. In this research, a SVM-based classifier is implemented as an example because SVM was reported to outperform other algorithms [5, 6]. The accuracy of SVM-based classification was also verified as satisfactory as up to 90% if the prediction model was sufficiently trained using a formal corpus like Reuter-21578 [7].

The identified topic of a document, then, must be migrated to the directory searching module to conclude the possible directory under which the document archives. The title of the document needs to be formulated by combining the topic, the document creator’s ID, and the time of archiving so that the document can be uniquely identified.

3. TOPIC IDENTIFICATION BASED ON AUTOMATIC DOCUMENT SUMMARIZATION

Automatic summarization is the process for making reduced version including the most important points of a given document using the functionality of computer programs. Making summaries automatically is an indispensable work as the amounts of information and documents increase. The Summly, an iPhone-based automatic summarization application developed by Nick D’Aloisio (http://summly.com/) and acquired by Yahoo.com is a typical example proves the importance of automatic summarization techniques nowadays. There exist two approaches to automatic summarization: extraction and abstraction. Extractive methods select a subset of existing words, phrases, or sentences in the original text to make a summary. Abstractive methods build an internal semantic representation and then use natural language generation techniques to make a summary that is closer to what a human might generate. Abstractive methods can give a liberal translation and therefore perform more comprehensive and realistic summarization. However, because of burdens in implementing and training a prediction model used in concluding keywords or keyphrases by projecting word vectors onto the n-dimensional corpus-based space, extractive methods are more widely used rather than abstractive methods.

Conventional approaches of extractive methods usually require training the prediction algorithm, or a classifier, using predefined category-based sample data, and this type of learning procedure is called as the supervised learning. In supervised learning, each set of sample data is composed of a pair of a document (or a word) and its associated category in the form of a vector. By reading sample data, a prediction algorithm can form a vector space constructed by given categories, and therefore can put the vector of a given document (or word) onto corresponding location within the vector space. While the supervised methods can produce reliable outputs based on pre-validated data, they have limitations in application caused by the large amount of training data as well as by the quality of data sets. Usually a number of documents with identified keywords or keyphrases are required to train a classifier, and therefore burdens in time and computing capacity are indispensably exhibited. Moreover, wrong results can be outputted in case of data with biased subject being inputted. Therefore, to meet with these limitations, unsupervised methods, such as TextRank [8] and LexRank [9], which eliminate the process of training using sample data are gaining much interest. The TextRank algorithm exploits the structure of the text itself to determine keyphrases that appear 'central' to the text in the same way that PageRank [10] selects important Web pages. Because the TextRank enables the application of graph-based ranking algorithms to natural language texts, it produces results independent to the training data and language types.

4. PROTOTYPE DESIGN

The prototype system is designed to initiate the function of topic extraction simultaneously with the user's trial to save the working document. Indexing the document by tagging the identified topic with user's ID and time, the prototype transmits and stores the document into the cloud storage. A dialogue between a user and the prototype is also needed to check whether the resultant topic is proper or not. If the user confirms that topic has no problem, the prototype transmits the file to the cloud storage with tagging required information about the user's ID and the time of archiving: Automatic archiving can be completed. Figure 2 shows the sequence of functions the prototype has.

The Stemming and Vectorizing module are implemented by using 'Word stemming tool' and 'Vector creating tool' of 'Yale', an open source environment for KDD(Knowledge Discovery and Data mining) and machine learning [11], respectively. As announced previously, this research deploys the SVM algorithm as the classification method. Therefore, using the LibSVM [12], a classifier is implemented. Following codes show the procedures to convert file format and to make the classifier read the file.

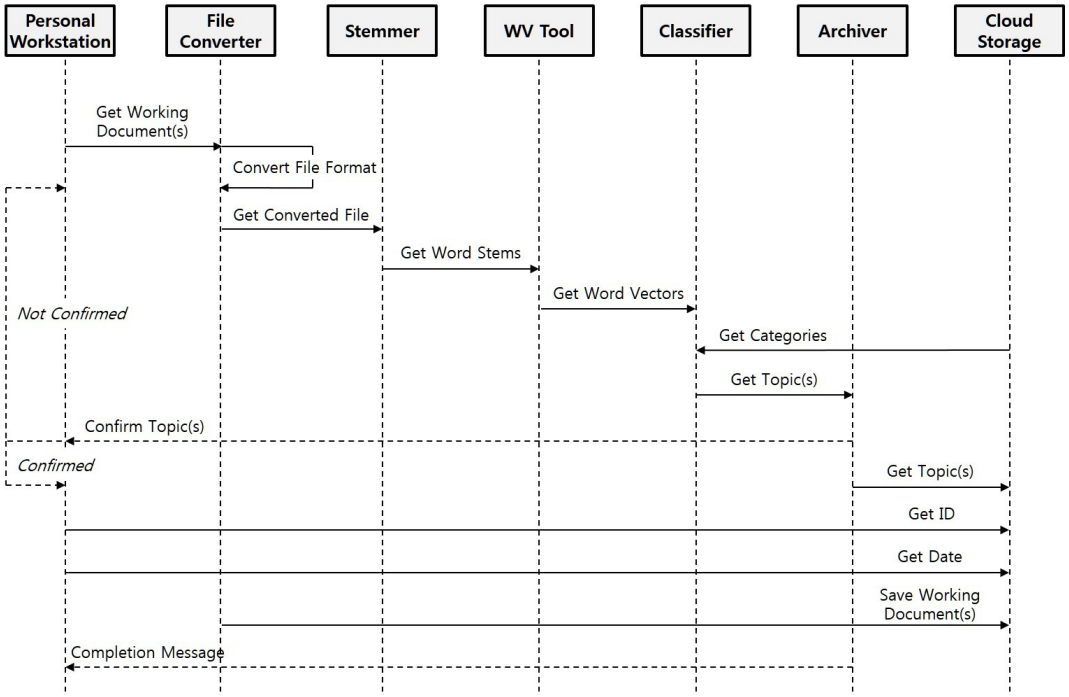

Figure 2. Execution sequence of the prototype system

```
if(predict_probability == 1) {
       if(svm_type == svm_parameter.EPSILON_SVR || svm_type ==
svm_parameter.NU_SVR) {
       System.out.print("Prob. model for test data: target value = predicted
value + z,\nz:      Laplace distribution e^(-
|z|/sigma)/(2sigma),sigma="+svm.svm_get_svr_probability(model)+"\n");
       }
       else {svm.svm_get_labels(model,labels);
prob_estimates = new double[nr_class];
       output.writeBytes("labels");
       for(int j=0;j<nr_class;j++)
       output.writeBytes(" "+labels[j]);
       output.writeBytes("\n");
       }
}
while(true) {
       String line = input.readLine();
       if(line == null) break;
       StringTokenizer st = new StringTokenizer(line," \t\n\r\f:");
       double target = atof(st.nextToken());
       int m = st.countTokens()/2;
       svm_node[] x = new svm_node[m];
       for(int j=0;j<m;j++) {
       x[j] = new svm_node();
       x[j].index = atoi(st.nextToken());
       x[j].value = atof(st.nextToken());
       }
```

Identified topic needs to be combined with creator's (user's) ID and the time to archive as the following codes show.

```
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyyMMdd", Locale.KOREA);
String recordedDate = dateFormat.format(new Date());
String tagName = topicName + "-" + userID + "-" + recordedDate;
System.out.printf("Indexing tag is %s\n", tagName);
JOptionPane.showMessageDialog(null, tagName);
```

Under the condition that the cloud storage has i directories (i=1,2,…,n), the document must be archived in one of existing directories. To search proper directory coincide with the identified topic, the 'Hash' function can be used, and corresponding programming codes can be as follows;

```
HashMap<String, Integer> categoryHash = new HashMap<String, Integer>();
categoryHash.put("Topic1", 0);
categoryHash.put("Topic2", 1);
categoryHash.put("Topic3", 2);
.
.
.
categoryHash.put("Topici", i-1);

int indexOfCategory = categoryHash.get(topicName);
System.out.printf("Searching result is %d index\n", indexOfCategory);
JOptionPane.showMessageDialog(null, indexOfCategory);
```

If the searching result is correct and the user confirm it, then a message of processing archiving needs to be sent to the user, as the following codes show;

```
try {
     BufferedWriter tagFile = new BufferedWriter(new FileWriter(filePath));
     tagFile.write(tagName);
     tagFile.close();
} catch (IOException e) {
     System.err.println(e);
     System.exit(1);
}
JOptionPane.showMessageDialog(null, "The document is to be saved as '" +
filePath + "'");
```

Finally the document is to be archived in the concluded directory with the title of 'topic-ID-date', and a message informing the completion of archiving is to be notified to the user with displaying the title and location of archiving, as following codes show;

```
String msg = "The topic of working document is '" + topicName + "' ?";
int ret = JOptionPane.showOptionDialog(null, msg, "Message Window",
JOptionPane.YES_NO_OPTION, JOptionPane.PLAIN_MESSAGE, null, null, null);
switch (ret) {
case JOptionPane.YES_OPTION:
     JOptionPane.showMessageDialog(null, "The file '" + tagName + "' has been
archived.");
     break;

case JOptionPane.NO_OPTION:
     JOptionPane.showMessageDialog(null, "user canceled");
     break;
}
```

5. CONCLUSIONS

Zero-client-based cloud storage is gaining much interest as a tool for centralized management of organizational documents. Besides the well-known cloud storage's defects such as security and

privacy protection, users of the zero-client-based cloud storage point out the difficulty in browsing and selecting the storage directory because of its diversity and complexity. To resolve this problem, this study proposes a method of intelligent document archiving by applying an automatic summarization-based topic identification technique. Since the cloud storage plays the role of VDI-enabled database, pervasive document storing and retrieving can be naturally enabled. Although not a few researches also tried to enhance the functionality of corporate archiving systems, no research has suggested the intelligent archiving by automatically attaching the title of documents to leverage the usability of zero-client-based cloud storage, which is the main contribution of this study.

Issues in this paper remain points to discuss concerning technical limitations and future works. Especially, discussions around the algorithms for automatic document summarization need to be addressed, because the application efficiency of SVM is doubted because of the burden in training the prediction model. Training the prediction model via server-side computing might be a solution for this problem, however the computing load a server must endure can also keep increasing as the use of smart devices increase. Therefore, approaches of unsupervised methods can yield very effective solutions to meet this problem. However, more formal and statistical validation on the performance of the unsupervised methods is required to acquire the reputation supervised methods have gained without the smallest strain. Meanwhile, a formal corpus must be developed to guarantee the performance of conventional text mining techniques, because most of conventional algorithms in the area of text mining are much dependent upon the quality of corpus. More formal, general and universal corpus must be developed so that the results from applying the corpus can be unbiased and objective. Since the corpus can be applied in setting the directories of cloud storage, this supplement can also make up for the applicability of intelligent document archiving suggested by this study.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2013S1A5A2A01017530).

REFERENCES

[1] Liu, Q., Wang, G., & Wu, J., (2012) “Secure and privacy preserving keyword searching for cloud storage services”, Journal of Network and Computer Applications, Vol.35, No.3, 927-933.
[2] Pamies-Juarez, L., García-López, P., Sánchez-Artigas, M., & Herrera, B., (2011) “Towards the design of optimal data redundancy schemes for heterogeneous cloud storage infrastructures”, Computer Networks, Vol.55, 1100-1113.
[3] Svantesson, D. & Clarke, R., (2010) “Privacy and consumer risks in cloud computing”, Computer Law & Security Review, Vol.26, 391-397.
[4] Kim, S., Suh, E., & Yoo, K., (2007) “A study of context inference for Web-based information systems”, Electronic Commerce Research and Applications, Vol.6, 146-158.
[5] Basu, A., Watters, C., & Shepherd, M., (2003) “Support Vector Machines for Text Categorization”, Proceedings of the 36th Hawaii International Conference on System Sciences, Vol.4.
[6] Meyer, D., Leisch, F., & Hornik, K., (2003) “The support vector machine under test”, Neurocomputing, Vol.55, 169-186.
[7] Hsu, C.W., Chang, C.C., & Lin, C.J., (2001) “A Practical Guide to Support Vector Classification: LibSVM Tutorial”. In http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.
[8] Mihalcea, R. & Tarau, P., (2004) “TextRank: Bringing order into texts”, Proceedings of EMNLP, Vol.4, No.4.
[9] Erkan, G. & Radev, D.R., (2004) “LexRank: Graph-based lexical centrality as salience in text summarization”, Journal of Artificial Intelligence Research, Vol.22, No.1, 457-479.
[10] Brin, S. & Page, L., (1998) “The anatomy of a large-scale hypertextual Web search engine”, Computer Networks and ISDN Systems, Vol.30, 1-7.

[11] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T., (2006) "YALE: Rapid Prototyping for Complex Data Mining Tasks", Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06).

[12] Chang, C. & Lin, C., (2011) "LIBSVM: a library for support vector machines", ACM Transactions on Intelligent Systems and Technology, Vol.2, No.3, 1-27.

AUTHOR

Keedong Yoo is an associate professor in the Department of MIS at Dankook University, South Korea (kdyoo@dankook.ac.kr). He has B.S. and M.S. in Industrial Engineering from the POSTECH (Pohang University of Science and Technology), South Korea; and a Ph.D. in Management and Industrial Engineering from the POSTECH. His research interests include knowledge management and service; intelligent and autonomous systems; context-aware and pervasive computing-based knowledge systems.