IJCSC International Journal of Computer Science and Communication Vol. 3, No. 1, January-June 2012, pp. 95-99

KEY WORD BASED WORD SENSE EXTRACTION IN TEXT: DESIGN APPROACH

Shahana Bano[1], K. Raja Sekhara Rao[2] and M. Sai Sandeep[3]

[1]Department of Computer Science and Engineering, K.L. University, Vijayawada, India,
E-mail: shahana.klce@gmail.com.

[2]Department of Computer Science and Engineering, K.L. University, Vijayawada, India,
E-mail: rajasekhar.kurra@klce.ac.in.

[3]Department of Computer Science and Engineering, K.L. University, Vijayawada, India,
E-mail: muralasaisandeep@yahoo.com.

ABSTRACT

Text Mining is one of the major areas of Data Mining where data in the form of text is used. Processing of text is very important in day to day life as most of the information is available in the form of text. Data stored in most text documents are semi structured data in that they are neither completely unstructured nor completely structured. In such cases the identification of context of word in a text is often needed so as to make the document less complex to understand. This paper aims at processing of text in efficient manner to identi fy keywords by removing stop words. We present an effective way of processing of word so that all the details of a particular word such as meaning, context, part of speech etc., are made available to the user.

Keywords: Context, processing, semi structured, stop words, Text Mining.

1. INTRODUCTION

Text Mining is of wide use in recent days as large part of data is stored in text databases. Data stored in most text databases are *semi structured data* in that they are neither completely unstructured nor completely structured. For example, a document may contain a few structured fields, such as *title*, *authors*, *publication date*, *and category*, and so on, but also contain some largely unstructured text components, such as *abstract* and *contents*. There have been a great deal of studies on the modeling and implementation of semi structured data in recent database research. Moreover, information retrieval techniques, such as text indexing methods, have been developed to handle unstructured documents.

Word processing is very useful in cases where we use large text. In this paper we propose a way in which word processing is done in such a manner that the context of the word including part of speech, meaning, number of appearances. For this we use Keyword Based Association Analysis, a analysis collects sets of keywords or terms that occur frequently together and then finds the association or correlation relationships among them.

2. METHODOLOGY

In the proposed procedure for word processing the word that is typed most recently is taken. The first step in this word processing is to identify keywords for representing documents, a preprocessing step often called tokenization.

To avoid indexing useless words, we use a *stop list*. A stop list is a set of words that are deemed "irrelevant." For example, *a*, *the*, *of*, *for*, *with*, and so on are stop words, even though they may appear frequently.

This word is searched in the entire document for any repetitions. If any they are printed in the text area that is made available in the right side top of the user interface. To find the meaning of the word we need some kind of dictionary integrated into our application. For this purpose we use Word Net data base which serves as a key source to give the meaning of the word and also the part of speech the word belong. The context of the word that is the words those come after and before the specified word are also printed. While printing the context we will not print the words in the stop list i.e., the stop words are removed.

The key feature of the processor is that it not only checks for the word in current file but also checks for the word in all the text files that are present in the current directory. The appearances of the word in the different files of the directory are printed in the right side bottom half of the interface.

The procedure is implemented in java using the handling of events. Here space event is handled using separate event handler and then the word that is caught is used for further processing.

The following are the steps involved in the processing

(a) Tokenize the text in the file.

(b) Identify the word typed.

(c) Get the various files in the current directory and filter the text files in it.

(d) Identify the context of the word.

e.g.: no. of appearances, meaning of the word, parts of speech it belongs etc., in the current file.

(e) Remove the stop words and print the details.

(f) The step is repeated for all the files in the directory.

2.1. Tokenize the Text in the File

In this step the text in the file is split into words i.e., tokenized. Tokenizing of the text is very essential because we need to identify the occurrences of word and from large sentences it is difficult to identify the word. So, we tokenize the text into words by using a string buffer and various methods in java that are used to split the sentence into words.

2.2. Identify the word typed

The key word that is to be searched is to be identified from the user action. Once the user types the word and hits space an event is generated and that event is caught using event handlers. The event handler and the tokenized words are used to identify the word typed which is the word that we need to search for.

2.3. Get various files in the directory

In this step we get the various files in the directory and sort out the files relevant to the current files. This step enables us to identify various files so as to perform searching in those files. The files are fetched from the current directory.

2.4. Identify the Context of the Word

This step emphasizes on identifying the context of the word. The context of the word includes identifying the meaning of the word, finding the words that are before and after the word, number of repetitions of the word.

The flow chart for the entire process is shown in the following figure.

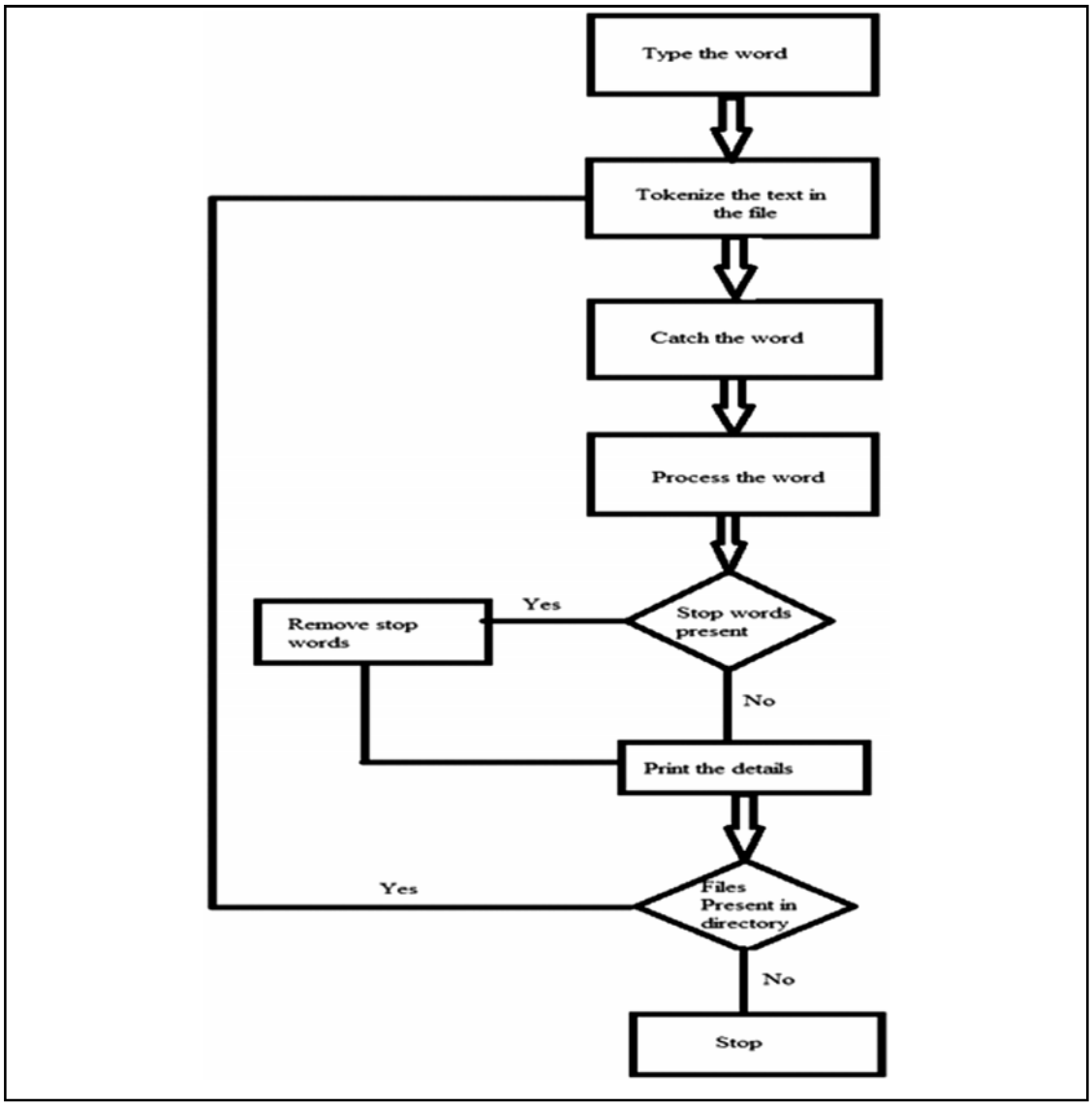

Figure 1: Flow Chart for the Flow of Events.

2.5. Removing stop words

The text contains stop words. These words are unnecessary and are to be discarded while processing of the word. A predefined list of stop words called stop list is taken and the tokens of text are compared to those in the stop list. If stop words are found then they are discarded.

The above steps are repeated for all the files present in the directory.

3. RESULT AND DISCUSSION

The word processor is implemented using the tool net beans IDE. Java is used as the programming language. In order to identify the meaning and the parts of speech the word belong to we use word net database. Word net is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. The purpose is twofold: to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications.

The word whose details are to be found is searched in the word net and the meaning and the parts of speech the word belongs are identified. The word net data base is available online as a open source on world wide web.

After the word net is installed the path of the word net is to be set for the correct usage of word net. All the software used in the project are open source and can be obtained free of cost

The tokenizing of words is done using the java functions. The jaws-bin jar file is used to associate java with the word net data base. The various packages used are javax.swing, java.io, java.awt. A string buffer is used to store the text and also the tokens of the text. The event handlers play a key role in catching the word. Here the space event is handled for word processing. The details of the word in the current file is displayed in the text area on the right side top of the interface and the details of the word in the files in the current directory is displayed in the bottom right side of the interface.

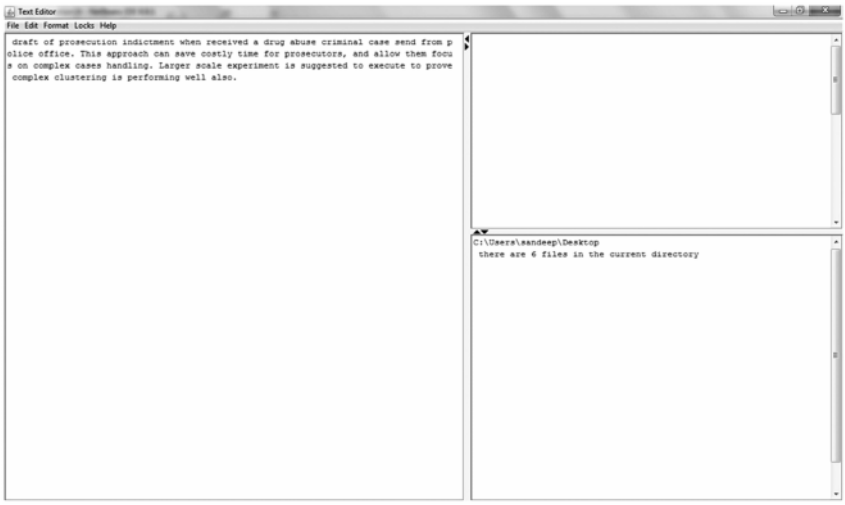

Figure 2: Screen Shot Showing Number of Files in the Directory.

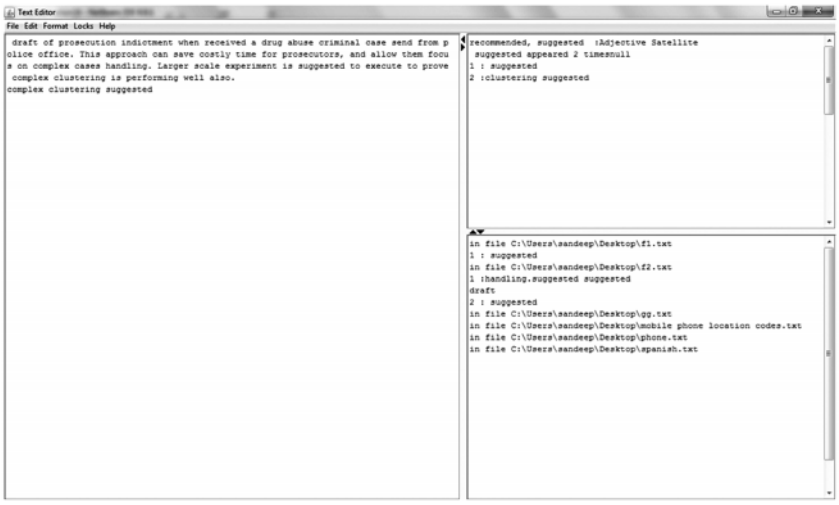

Figure 3: Screen Shot Showing the Details of Word "Suggested".

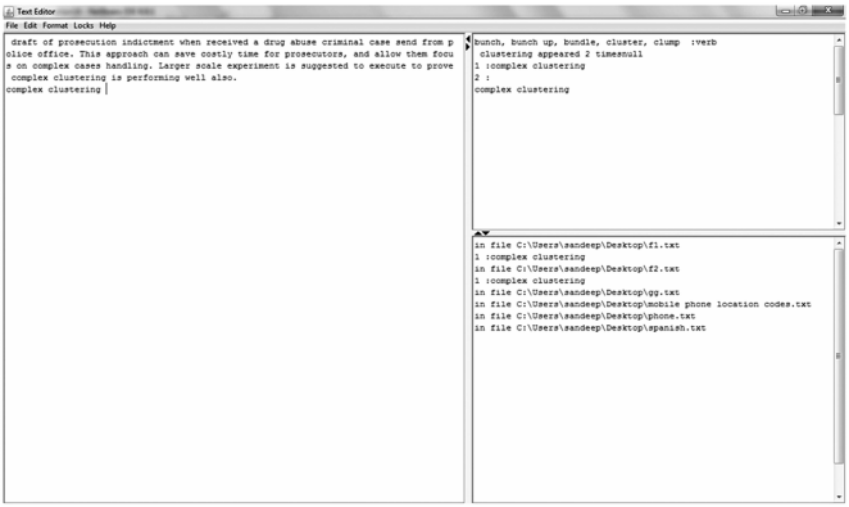

Figure 4: Screen Shot Showing the Processing of the Word "Clustering".

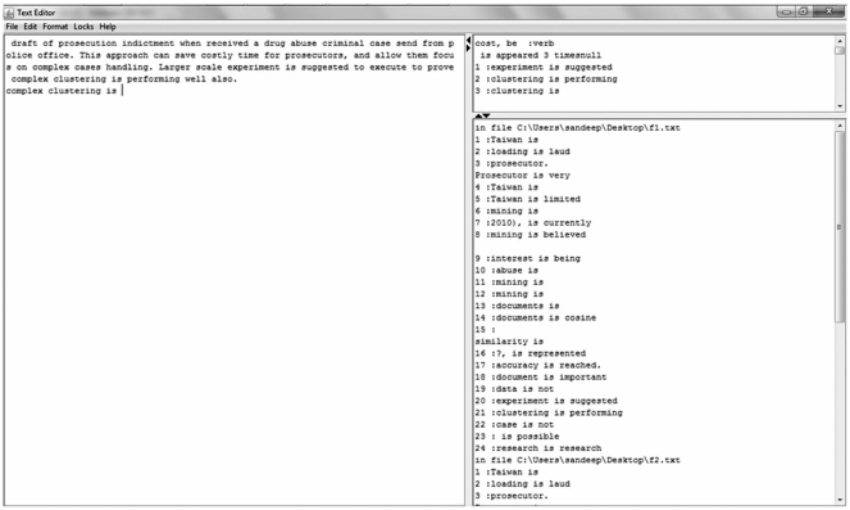

Figure 5: Screen Shot Showing the Processing of the Word "is".

4. CONCLUSION & FUTURE WORK

4.1. Conclusion

The work processes the word present in a file when we want to edit that file. The main advantage is that we can identify which files contain the word we want and in what context they are used which provides a easy method of searching a word in the similar files present in the entire directory. The processing explained can be applied for already existing file. If we want to process a word from a newly creating file then it is recommended to save the file first and then start editing the text. It is very advantageous in places where frequent use of certain words is employed and also to reduce the redundancy of words thus redundancy of files is reduced which in turn reduces the disk usage.

4.2. Future Work

The proposed approach of word processing can be used in areas where lot of text processing is done. The approach searches the word only in the current directory. This can be extended to searching within the drives. The approach can be used for searching the files in a distributed system which involves more complex operations to be performed.

REFERENCES

[1] Anupama Surendran,"Data Mining Techniques to Analyze the Risks in Stocks/Options Investment", *International Conference on Intelligent Agent & Multi-Agent Systems*, IAMA 2009, pp. 1-3.

[2] "Data Mining and Concepts", by Jiawei han and Micheline kamber.

[3] Weiguo Fan, et. al., "Tapping the Power of Text Mining", *Communications of the ACM*, **49(9)**, 2006.

[4] Roberto Navigli, "Word Sense Disambiguation: A Survey", *ACMComputing Surveys*, **41**, No. 2, Article 10, Publication date: February 2009.

[5] WordNet : http://wordnet.princeton.edu/ Wordnet is a database for English words developed in Princeton university.

[6] Gobinda G. Chowdhury, "Natural Language Processing", *Dept. of Computer and Information Sciences*, University of Strathclyde, Glasgow G1 1XH, UK.

[7] Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996), "From Data Mining to Knowledge Discovery: An Overview", in U. Fayyad et al. (eds.), *Advances in Knowledge Discovery and Data Mining*, MIT Press, Cambridge, Mass., 1.

[8] Text Mining http://en.wikipedia.org/wiki/Text_mining.

[9] Web Intelligence: kis.maebashi-it.ac.jp/wi01/www.webintelligence.com/.

[10] Intelligence on the Web: www.fas.org/irp/intelwww.html WIN:home WEB INTELLIGENCE NETWORK,smarter.net/.