IMPLEMENTATION OF TEXT MINING WITH AUXILIARY INFORMATION USING CLASSIFICATION

Miss Monika D Khatri[1], Prof S.S Dhande[2]
[1]Master of Engineering, [2]Guide, Department Of Computer Science & Engineering
Sipna College of Engineering & Technology, Amravati, India

Abstract: The huge amount of information stored in shapeless text cannot simply be used for advance processing by computers, which usually handle text as simple sequence of character strings. So, specific pre-processing, text mining methods and algorithms are required in order to mine useful patterns. Text mining refers generally to the process of mining interesting information and knowledge from shapeless text. In our implementation we have considered text document, PDF and CSV files. We can write document or data copied from web. And on this data preprocessing through NLP is done and then data mining for acquiring clusters.And text mining for side information using classification.
Keywords: Text mining,Side information, NLP, N-gram.

I. INTRODUCTION

As computer networks become the backbones of science and economy huge amount of machine readable documents become available. There are predication that 85% of business information lives in the form of text (TMS05 2005)[1].Text mining is a emerging area of computer science which encourages tough links with NLP, data mining, machine learning, information retrieval and knowledge. Text identification and examination of interesting mining request to extract useful information from shapeless textual data through the patterns [2].In this paper we are going to discuss methods such as NLP, classification, n-gram algorithm.

A. Text mining: Text mining is the data analysis of text resources so that pure, earlier unknown knowledge is discovered.[3].The steps in text mining are:

- Text assets
- Data analysis
- Evaluation/ interpretation
- Knowledge

B. Side information

The problem of text clustering occurs in the background of many application domains such as the web, social networks, etc. An extraordinary amount of work has been done in latest years on the problem of clustering in text collections [4], [5], [6], [7] in the database and information retrieval communities.

Some examples of such side-information are as follows

1) Text Document Contains Links: Text documents (element) contains link which contains a lot of helpful information for mining reasons. As in the previous case, such elements may often provide closes about the relationships among documents in a way which may not be easily reachable from raw content.

2) Meta-data: Many web documents have meta-data associated with them which match up to different kinds of elements such as the origin or other information about the document. In other cases, data such as ownership, locality, or even temporal information may be useful for mining function. In a number of network and user-sharing applications, documents may be associated with user-tags, which may also be quite informative.

C. NLP

NLP includes a wide range of disciplines and responsibilities focused on extending the capabilities of text mining, or the extraction of knowledge from shapeless text (Hearst 1999)[8],most recently by including the machine-learning (ML) paradigm of language processing. NLP algorithms have met with some success in formal, structured fields with limited lexes such as medicine and biochemistry (Tanabe et al. 1999)[9]. The various methods of NLP are:

- Part-of-speech tagging (POS) determines the tagging of part of speech
- Splitting
- Text chunking aims at grouping adjacent words in a sentence;
- Tokenize

D. N-gram algorithm

N gram is a conflation technique.String-similarity approaches to conflation engages the system calculating a measure of similarity between an input query term and each of the separate terms in the database. Those database terms that have a high similarity to a query term are then shown to the user for possible inclusion in the query. N-gram matching techniques are one of the most ordinary of these approaches (Freund & Willett, 1982)[10]. An n-gram is a set of n successive characters extracted from a word. The main idea behind this approach is that, similar words will have a high share of n-grams in common. Typical values for n are 2 or 3, these matching to the use of digrams or trigrams, respectively. For example, the word MONIKA (computer) results in the generation of the digrams

M, MO, ON, NI, IK, KA, A and

the trigram **M, *MO, MON, ONI, NIK, IKA, KA*, A** where '*' denotes a padding space. There are n+1 such digrams and n+2 such trigrams in a word containing n characters. n-grams can also be used for efficient approximate matching. By converting a sequence of items to

a set of n-grams, it can be implanted in a vector space, thus allowing the sequence to be evaluated to other sequences in an capable manner. N-gram-based searching can also be used for plagiarism detection. N-grams find use in several areas of computer science, computational linguistics, and applied mathematics.

II. LITERATURE REVIEW

There are lots of clustering problems clarified by database community [11], Several evaluation of different clustering algorithms found in [12]. In text mining various methods are based on the statistical study of a word or term [13].Document clustering has been examined for use in a number of different regions of text mining and information retrieval [14]. Initially, document clustering was examined for improving the accuracy or considered in information retrieval systems and as an efficient way of finding the adjacent neighbors of a document. Agglomerative hierarchical clustering and K-means are two clustering techniques that are usually used for document clustering. Agglomerative hierarchical clustering is always represented as "better" than K-means, although slower. A widely known study, discussed in [15] indicated that agglomerative hierarchical clustering is greater to K-means, although we hassle that these results were with non-document data. In the document domain, Scatter/Gather, a document browsing system based on clustering, uses a mixture approach involving both K-means and agglomerative hierarchical clustering. K-means is efficient because of its efficiency and agglomerative hierarchical clustering is used to improve quality. In NLP, the work of the first stage was focused on machine translation (MT). Following a few early birds, including Booth and Richens' investigations and Weaver's influential memo on translation of 1949 (Locke and Booth, 1955)[16], research on NLP began in earlier in the 1950s.The Teddington International Conference on Machine Translation of Languages and Applied Language Analysis in 1961 was perhaps the high point of this first stage: it descripted work done in many countries on many ideas of NLP including morphology, syntax and semantics, in understanding and generation, and ranging from formal theory to hardware [17]. Plath's account (1967) [18]of NLP research at Harward shows the development of computational grammar with its lexicon and parsing strategy very clearly. But as Plath also makes clear, those concentrating on syntax did not suppose that this was all there was to it: the semantic problems and needs of NLP were only too clear to those focusing, as many MT workers were, at the translation of unrestricted real texts like scientific papers. The strategy was rather to deal with syntax first, if only because semantic uncertainty resolution might be finessed by using words with broad meanings as output because these could be given the necessary more specific interpretations in context. Schank's arguments for the Yale group's use of more event-oriented scripts developed this line in the context of earlier work by linking individual propositional case frames with the larger structures via their semantic primitives (cf Cullingford, 1981). The NLP field from late 1940's to until now is flourishing.

III. PROPOSED METHODOLY

1.Collection of datasets for data mining- In which we would be selecting various datasets and finding the best out of them for text mining with side information. This may be include methods like crawling, filtering, etc.

2. Pre processing data with natural language processing-In which we would be applying Natural language processing techniques like splitting the document, tokenizing, part of speech tagging and chunking to find only the action words from the given text datasets. The field of Natural Language Processing (NLP) aims to convert human language into a formal representation that is easy for computers to manipulate. The general goal of NLP is to achieve a better understanding of natural language by use of computers. It employs simple and durable techniques for the fast processing of text.

3. Development of mining algorithm with Natural Language Processing-for demonstration of text mining approach, we would developing the mining algorithm like k-means to extract data.

4. Use of Word Net-It is a dictionary of language containing meanings, senses, etc of words. In this step we will be showing the meaning of searched data.

5. Combination of mining with natural language processing-In this all the above work would be combined in order to demonstrate our algorithm. This would demonstrate the use of NLP in text mining and obtaining the efficient output by using semantic score and n gram.

IV. PROCESS FLOW

The first step is to read the document. After reading the document the preprocessing of document is done. The preprocessing is done so that the computer can understand the language very well and process it. The split process makes each sentence separate from the other. The tokenize method considers each word as a separate tokens. Here we have considered "|" symbol for separating tokens.

The next part is POS tag assigning part in which words are given part of speech such as noun, pronoun, adjective, adverbs, preposition, etc.

Due to POS tag we will be able to separate the useless words from the preprocessed documents. Now the unwanted data are removed and the words are also reduced to their stems the output will be filtered. Now the user enters the input for mining. The input words are separated by space. Here for text mining we are using k means for clustering and then semantic matching for scores. For side information we are using bisecting k means for clustering, n gram algorithm for searching purpose and mean value based classifier for showing efficient output. The word net dictionary is also used for displaying synsets. The word net dictionary that we are using is lexical database written in English language. It can have storage of words with their meanings. This is called as synsets. The senses of words are the meaning of words as noun, adjective, etc. The programming is done to implement this word net dictionary in our project. Due to this if the meaning of required word is searched then the output is

shown with that meaning also.
Now semantic matching is used for matching the input and the filtered words. The input words can be in any number. The semantic matching is the algorithm used for matching the string. The scores obtained by this is then given to the classifier. The mean value based classifier arranges them in descending order and displays only those scores whose value is greater or equal to mean. Due to this the output do not contain unnecessary data. The n gram algorithm is used for searching purpose in the side information. The side information is searched with this n gram algorithm. The value of n can be 2, 3 and so on. The larger the value of n the better the output. Here the value of n depends on the number of input keywords. That means it works best for larger input. The n gram can also be used for stemming purpose. Bur we are using sharp NLP environment which has inbuilt stemmer. This stemmer will stem the filtered words into their root forms. The root forms allow the searching of data easy because the different words in the document may have same root word.

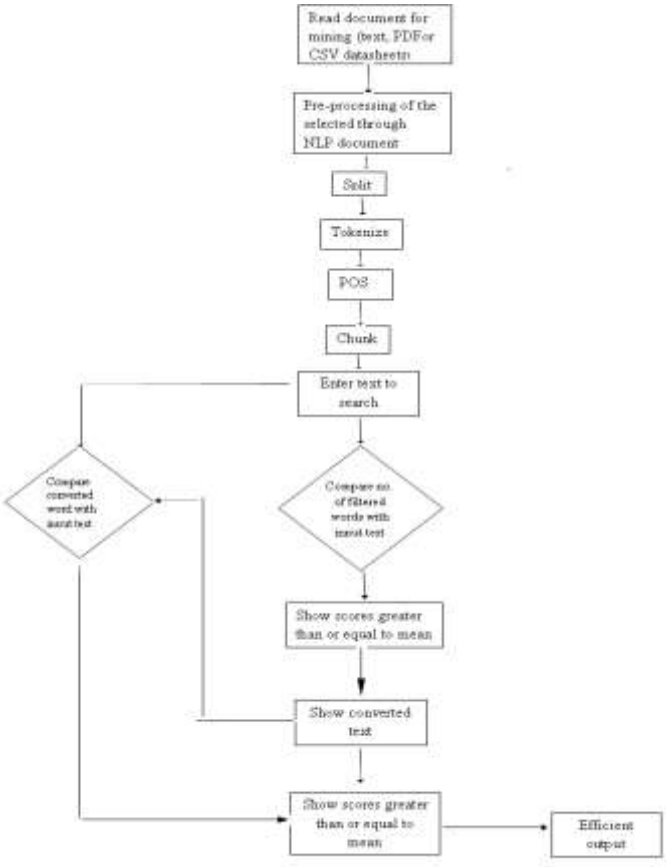

Figure 1:Process Flow

The stems words are easy to find in the dictionary. The input keywords are used in their original form. The other factor that improves the performance of the project is that we are using bisecting k means for side information. The bisecting k means is an improvement over k means algorithm. The bisecting k means although requires same complexity as of k means but it produces a good quality cluster than the k means. The clusters produced by this bisecting k means are then given to n gram for searching. The n gram searches for the matching and forms scores with the sysnets of it. The output is shown by mean value based classifier which gives optimized output.

V. EXPERIMENTAL RESULT

In this paper we have performed text mining using classification algorithm. Firstly the Document such as text file, PDF file or CSV file is read. We can copy the text from web also or we can write it by our self on the window if we need so. After that we perform Natural Language Processing on it. Methods such as split, tokenize, POS tag and chunking are applied. After pre processing user is said to enter the keywords for searching. The input text is then compared with filtered words and score is being displayed. The first scores are for text data and later the mining is done for side information stated in brackets. The side information is the information that gives some additional part about text in the document. For ex, consider the sentence "Hello World (world means everyone)". Now in the sentence Hello World is considered as text part and (world means everyone) is considered side information. After comparing the input text with the filtered words the final score is displayed. That score shows the efficient output. We have used Sharp NLP for NLP processing and Word Net 2.1 for acquiring synonyms.

A. Ideas to concept

Today, almost all work is done through the computer. As all people are busy they need everything to be done at fast speed. The internet provides us with that facility but it also increases our task of selecting the interesting data on us, the user. This task of finding interesting data from a large set of documents is cumbersome and requires a lot of time. So for the purpose of this the text mining software has been developed. The main advantage of this is that it doesn't consume much time. All a user needs is to put the document or data that he/she thinks is needed to be checked in the software. The software do not that you can check with the document. Hence no storage requirement is there.

B. Prerequisties

As the project is being developed in .net framework the skills of software development should be improved for it. The knowledge about Natural Language Processing is necessary along with text mining.

C. Configuration of system for development

The software is only on user side that is desktop based application. The need for software is that it should be installed on any machine having operating system windows XP, windows 7 and Linux. No specific hardware requirements. The .NET framework needs to be installed on the computer. We are using Microsoft Visual Studio 2010.As we are showing synonyms of the input words in our project. So the need of Word net is there. We have used the word net 2.1 version.

D. Design

1)Read document:In this the user have to read the document from his computer or flash drive or can write the data in that place or copy some data from web to it.

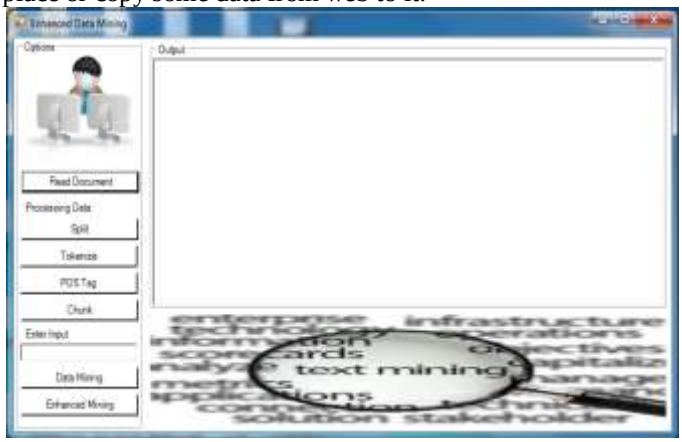

Figure 2: Read document

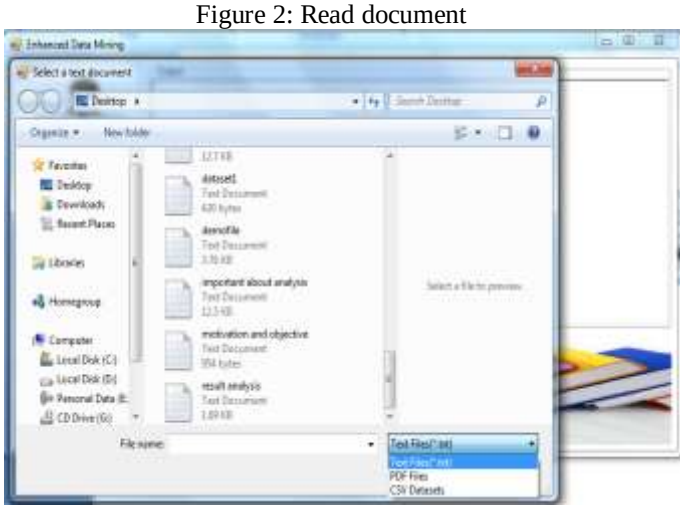

Figure 3: After clicking on read document

You can select the type of document that you want to mine.

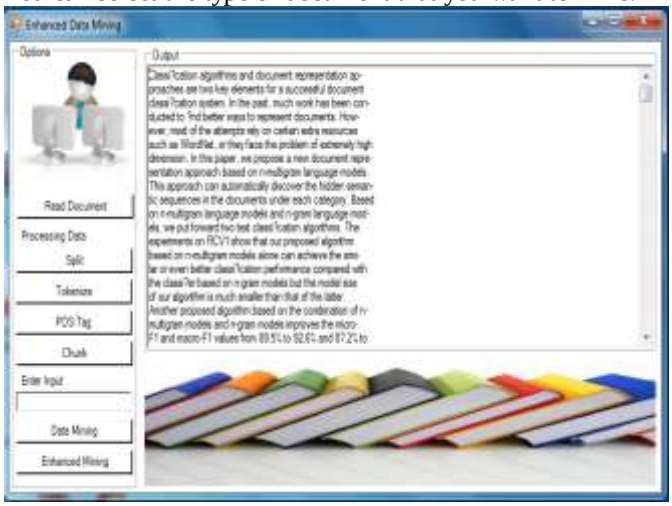

Figure.4: After reading document

2) Preprocessing data: In this the data that is being read is preprocessed by the means of NLP functions. The functions that are used as follows:

a)Split: In this document is split that is each sentence is being separated from each other in a single line.

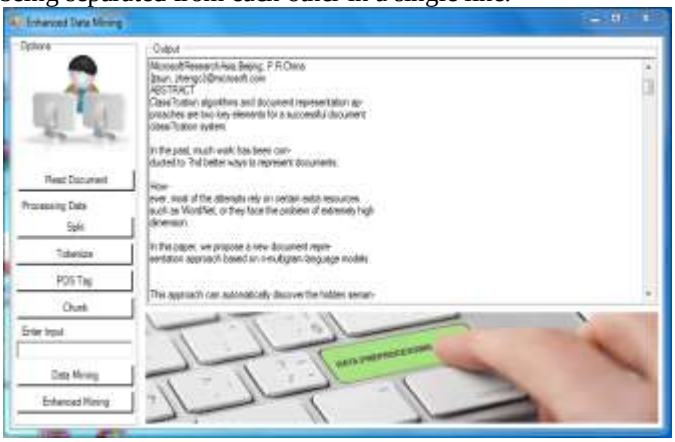

Figure 5: Splitting of data

b) Tokenize: The document is then tokenized with the each word as a separate token.

Figure 6: Tokenizing the data

c) POS Tag: The POS tag is used for assigning parts of speech to each word that is already tokenize in the previous step. The parts of speech such as noun, adjective, conjunction, preposition, etc.

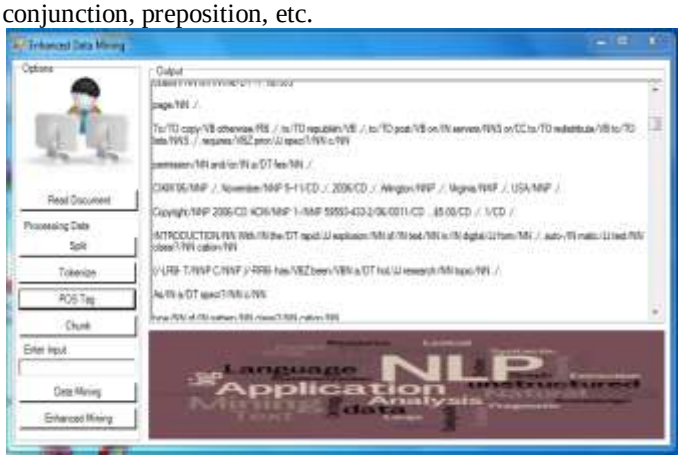

Figure 7: POS tag each tokenizied word

d) Chunk: It will chunk the tokenized POS tagged words that is removes unwanted words from the document by removing unwanted words such as conjunction, preposition, stop

words, etc. The output will list of filtered words that are useful words. This will reduce the number of words from the original document.

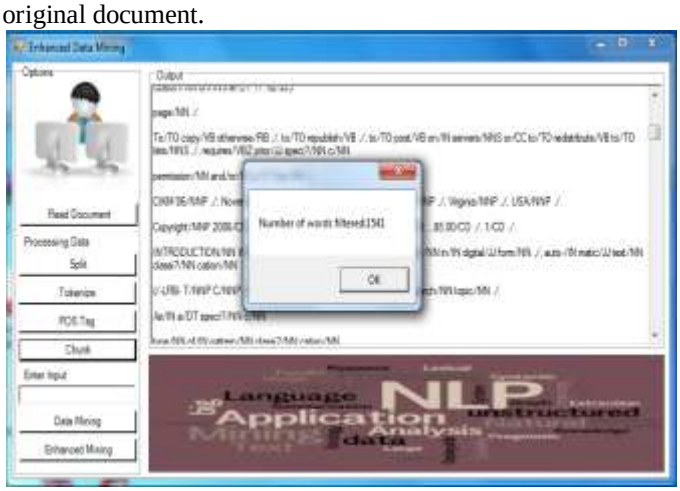

Figure 8: Number of filtered words

Figure 9: List of filtered words

3) Taking input from user: The user need to enter some text that he/she is interested in to find in the document. The words should be separated by space in between.

Figure 10: Input from user

4) Text mining: In this the list of filtered words and the input text are compared to find the matching between them and the scores are calculated by mean value based classifier. The classifier will show the output that is the line containing the searched words and the scores of that line.

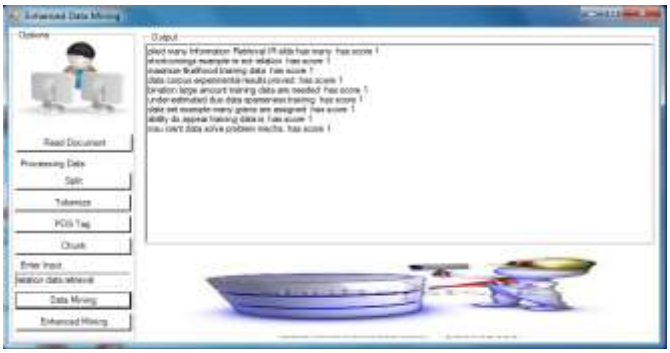

Figure 11: Mining of text data

5) Enhanced mining: In this the input keywords are searched through Word Net 2.1 dictionary for synonyms and they are displayed as "Converted Words" here.

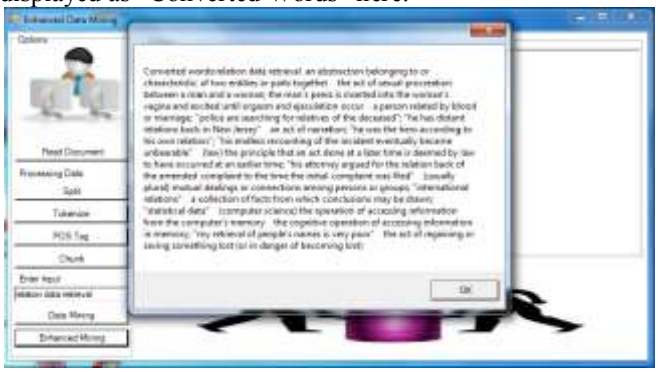

Figure 12: Converted words

After this for side information mining is done. The side information that we have considered is in different brackets.

Figure 13: Enhanced Mining.

E. System Configuration

- The output is checked on system having Windows 7 operating system and following results were found.
- The output time is calculated by embedding time ( ) function in the programming part.

F. Comparison

Table 1: Time evaluation of mining process with number of inputs 12.

Sr. No	Size	Filtered Words	Input Words	Time in MS
1	File 1	292	2	55ms
2	File2	1235	4	360ms

3	File3	3128	6	573ms
4	File 4	4128	8	645ms
5	File 5	4450	10	798ms
6	File 6	4437	12	890ms

G. Statistics

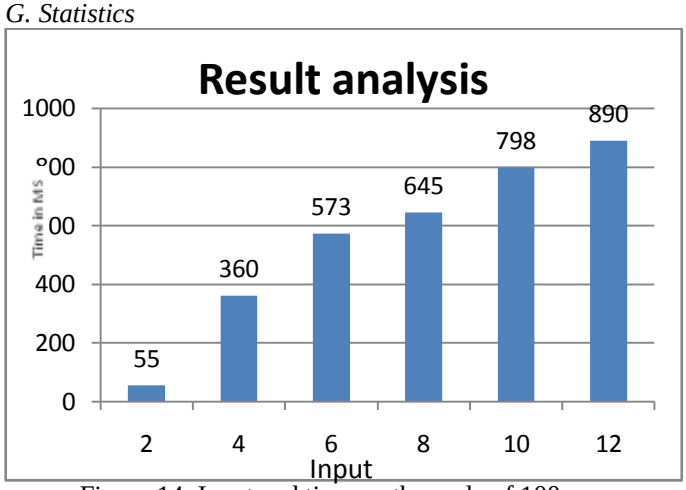

Figure 14: Input and time on the scale of 100ms

In the above graph the comparison is between the number of inputs given to the software and the time required to show the output. The first input is 2 that is number of words entered to mine in the document. The output is shown in 55ms. The second entered words were 4 that is number of inputs to be mined were 4 and the output was shown in 360 ms. The time increased due to number of inputs and it is also dependent on the number of filtered words from the input document. As you can observe that the graph grows in upward direct but the time required is not much. It is natural that as the input increases the time required will increase but not much.

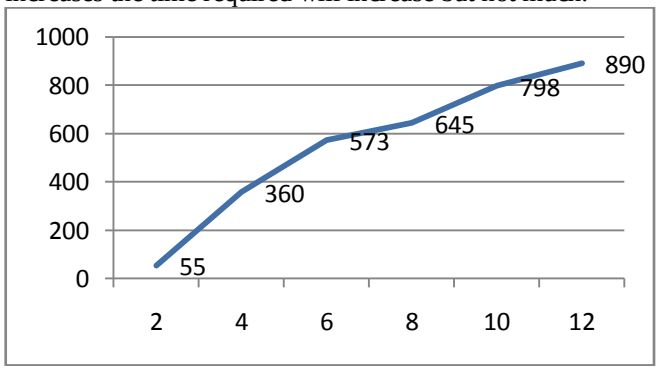

Figure 15: Line graph with the scale of 100ms

Above is the representation of Graph 1 in line graph. In this we can clearly notice the performance of the software.

VI. CONCLUSION

In this paper we have stated a software that can search for the required information and side information such as links, additional information about data that is meta data which are generally written in braces like ( ),{},[ ], etc. Our software will read all the side information in the document enclosed in any type of braces. We have used the Word Net 2.1 for acquiring synonyms of the input words so that the search result would be more specific. S-score is being used to display the scores of the sentences in the document through the user can gain access to the required word in the document without worrying. The software is user friendly. Any user can use it without knowing about mining process. Thus, the software is simple to use and provide efficient output

REFERENCES

[1] A Brief Survey of Text Mining Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß LDV FORUM – Band 20 – 2005 pg no.1-62
[2] Feldman, R., Sanger, J., The Text Mining Handbook. Cambridge University Press, 2007
[3] Hearst, M.A. 1999. Untangaling of text data mining. In Proc. Of the 37th ACL, College Park, MD, pp.3-10
[4] C.C.Aggarwal and P.S.YU, "A framework for clustering massive text and categorical data streams," in proc. SIAM Conf. Data Mining, 2006, pp. 477-481.
[5] D.Cutting, D.Karger, J.Pedersen, and J.Tukey, "Scatter/Gather:A Cluster-based to browsing large document collections," in Proc. ACMSIGIR Conf., New York, NY, USA, 1992, pp. 318-329.
[6] H. Schutze and C.Silverstein, "Projections for efficient document clustering," in Proc. ACMSIGIR Conf., New York, USA, 1997, pp. 74-81.
[7] M. Steinbach, G.Karypis, and V.Kumar, "A Comparison of document clustering techniques," in Proc. Text Mining Workshop KDD, 2000, pp. 109-110.
[8] Hearst, M. A. (1999). Untangling Text Data Mining. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, pp. 3–10, College Park, MD:Association for Computational Linguistics
[9] Tanabe, L., Scherf, U., Smith, L. H., Lee, J. K., Hunter, L., & Weinstein, J. N. (1999). MedMiner: an Internet text-mining tool for biomedical information, with application to gene expression profiling. BioTechniques, 27(6), 1210–4, 1216–7.
[10] Freund, G.E. & Willett, P. (1982) Online identification of word variants and arbitrary truncation searching using a string similarity measure. Information Technology: Research and Development, 1, 177-187.
[11] M. Steinbach, G. Karypis, and V. Kumar, "A Comparison of document clustering techniques," in Proc. Text Mining Workshop KDD, 2000, pp. 109-110.
[12] Jain and R. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ,USA: Prentice-Hall, Inc. , 1988.
[13] Shady Shehata, Fakhri Karray"An Efficient Concept based Mining Model for enhancing text clustering" IEEE transaction on knowledge and data engineering, vol. 22, no. 10, pp. 1360-1371, 2010
[14] U. Y. Nahm and R. J. Mooney, "A Mutually

Beneficial Integration of Data Mining and Information Extraction," Proc. 17th Nat'l Conf. Artificial Intelligence ,(AAAI '00), pp. 627-632, 2000

[15] G. Salton, A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing," Comm. ACM, vol. 18, no. 11, pp. 112-117, 1975

[16] Locke, W.N. and Booth, A.D. (eds). Machine translation of Languages, New York: John Wiley,1955.

[17] Karen Sparck Jones Natural language processing: an historical review, Computer Laboratory, University of Cambridge William Gates Building, JJ Thomson Avenue, Cambridge CB3 0FD, UK, oct 2001, pg. no-1 to 12.

[18] Plath, W. 'Multiple path analysis and automatic translation', in Booth 1967,267-315.

