ISSN:0975-9646

Text Rank: A Novel Concept for Extraction Based Text Summarization

[1] Dipti.D.Pawar, [2] M.S.Bewoor, [3]S.H.Patil

[1]M.Tech student, Department of computer BVUCOEP
[2]Associate Professor, Department of computer BVUCOEP
[3]Professor, Department of computer BVUCOEP

Abstract: **Indexing used in text summarization has been an active area of current researches. Text summarization plays a crucial role in information retrieval. Snippets generated by web search engines for each query result is an application of text summarization. Existing text summarization techniques shows that the indexing is done on the basis of the words in the document and consists of an array of the posting lists. Document features like term frequency, text length are used to assign indexing weight to words. Hence indexing weights of the document words are used to calculate the sentence similarity value between document words which remains independent on context. The word based index seems to be less efficient due to information retrieval problems like polysemy and Synonymy. Thus the significance of term for building the index is reduced and the emphasis is laid on the context of the document. This paper proposes an indexing structure in which index is built on the basis of context of the document rather than on the terms basis. While doing so we have also used novel concept of Lexical association (semantic association) between document words to calculate the similarity between sentences using computed indexing Weights. The proposed concept of sentence similarity measure has been used with the graph-based ranking method to create document graph and get summary of document.**

Keywords: **document word indexing, document graph, Lexical association, NLP, sentence similarity, text summarization, sentence vector.**

I.INTRODUCTION

Today internet contains vast amount of electronic collections that often contain high quality information. However, usually the Internet provides more information than is needed. User wants to select best collection of data for particular information need in minimum possible time .Text summarization [1] is one of the applications of information retrieval, which is the method of condensing the input text into a shorter version, preserving its information content and overall meaning. There has been a huge amount of work on query specific summarization [2] of documents using similarity measure. This paper focuses on sentence extraction based single document summarization. Most of the previous methods on the sentence extraction-based text summarization task use the graph based algorithm [4] to calculate importance of each sentence in document and most important sentences are extracted to generate document summary. These extraction based text summarization methods [3] give an indexing weight to the document terms to compute the similarity values between sentences Document features like term frequency, text length are used to assign indexing weight to terms. Therefore document indexing weight remains independent on context in which document term appears.

The indexing methods used in existing models cannot distinguish between terms reflected in sentence similarity values. Very little work has been done for the problem of context independent document indexing for the text summarization task. This proposed method aims at providing novel idea of context sensitive document indexing to resolve problem of context independent document indexing. We are considering the problem of context independent document indexing using Lexical association (semantic association). Main motivation behind using Lexical association [7] is the central assumption that the context in which word appears gives important information about its meaning.

This research work proposes an indexing structure in which index is built on the basis of context of the document rather than on the terms basis. So for this purpose first we have introduced the concept of OpenNLP tool. OpenNLP library [8] is a machine learning based toolkit for the processing of natural language text. It supports the most common Text preprocessing tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and co reference resolution. These tasks are usually required to construct more advanced text processing services. Sentence similarity values calculated using the context sensitive indexing provides the contextual similarity between two sentences. Context sensitive document indexing is implemented using Text-Rank algorithm [5] to compute how informative each of the document term is.

II.SYSTEM ARCHITECTURE

As shown in given block diagram we are accepting input as a text file only. Then file undergoes through different NLP phases.

We get the meaningful tokenized words as output of text preprocessing. Once grammatical aspect of word is clear we use Lexical association (semantic association) to find out association between document words from our offline wordnet dictionary. It is a dictionary in which we have statically stored semantic relationship between all the words in online dictionary and more that are not in it.

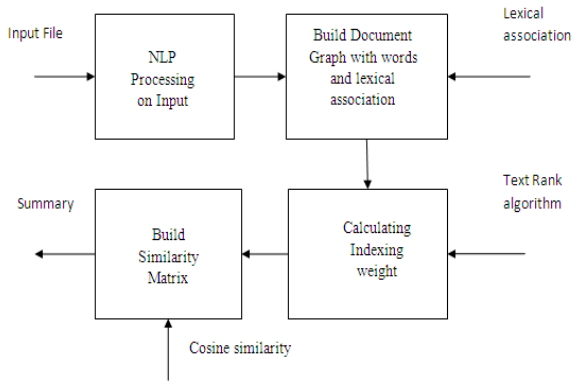

Fig.1 System Architecture Diagram

Next step document graph build using document words and lexical association. In graph word appear as node or vertex and lexical association between two words appear as an edge between those two words.TextRank [5] algorithms applied on this graph to calculate indexing weights of each document word. From these indexing weights we get the similarity between two nodes or words but not in sentences. Our next task is to find out sentence similarity using these indexing weights. Cosine similarity measure is used to construct similarity matrix. It is one of the similarity measure in which similarity between sentences is found using vectors of that sentences. Our method takes advantage of this similarity measure. In vector representation [10] of sentence words are simply replaced by its calculated indexing weight. Then this computed sentence similarity measure has been used with the graph-based ranking [4] method to build document graph and scored sentences. Then topmost scored sentences are included in summary of document.

III. System Implementation

Implementation is the stage in the project where the theoretical design is turned into a working system. Total implementation workflow of our system is divided into following modules

***Module 1**: Uploading input file and NLP processing on the input text file*

Input text file undergoes through different NLP[8] processing phases like Splitting, Tokenization, and Pos Tagging, Parsing etc., which results into meaningful document words. These document terms appears as a node in document graph. We have used OpenNLP library, which is a machine learning based toolkit for the processing of natural language text.

***Module 2**: Finding semantically associated words (Lexical Association)*

Once grammatical aspect of word is clear we use WordNet to find out different senses of the word. We use WordNet to understand the links between different parts of the document; subsequently extract the Lexical associations [9] between two document terms which are most relevant. In the table of lexical association first column shows words in text file and next column lists semantically related words with respective each word in first column.

***Module 3**: Finding context based indexing weight of document words.*

Given the semantic association between two terms in a document from LEXICAL ASSOCIATION table, the next task is to calculate the context sensitive indexing weight of each word in a document. A Text Rank algorithm [5] is used to find the context sensitive indexing weight of each term.

For this graph for given document is built. Let G= (V, E) be an undirected weighted graph to reflect relationship between terms in document, where each vertex V= {vj |1≤ j≤ |V|} denotes set of vertices and each vertex is document term and E is a matrix of dimensions |V | ×|V |. Each edge ejk € E gives the lexical association between the terms corresponding to the vertices vj and vk. The score of each vertex calculated using equation "(1)" given below.

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \qquad \textbf{(1)}$$

***Module 4**: Finding Sentence similarity and generating summary*

Next step is to find Similarity between sentences using the function SIM (Si, Sj). Similarity values calculated using context based indexing weights of document terms reflects the contextual similarity between words. Then this similarity between words is used to find similarity between sentences. So for each sentence Sj in the document, the sentence vector Sj is built using calculated indexing weights of sentences. The sentence vector [10] is calculated such that if a term vt present in sentence Sj, it is given a weight of term vt; else it is given a weight 0. The similarity between two sentences Si and Sj is computed using cosine similarity equation "(2)"given below.

$$\mathbf{SIM}\ (\mathbf{S_i}, \mathbf{S_j}) = \frac{\overrightarrow{S_i} \cdot \overrightarrow{S_j}}{|\overrightarrow{S_i}|\,|\overrightarrow{S_j}|} \qquad (2)$$

VI. Results

Following figures shows results generated by our system

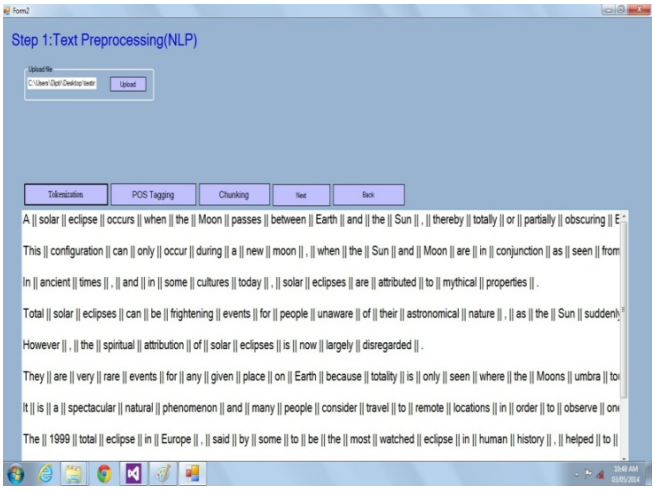

Fig: 2 Output screen of Text preprocessing on uploaded text file

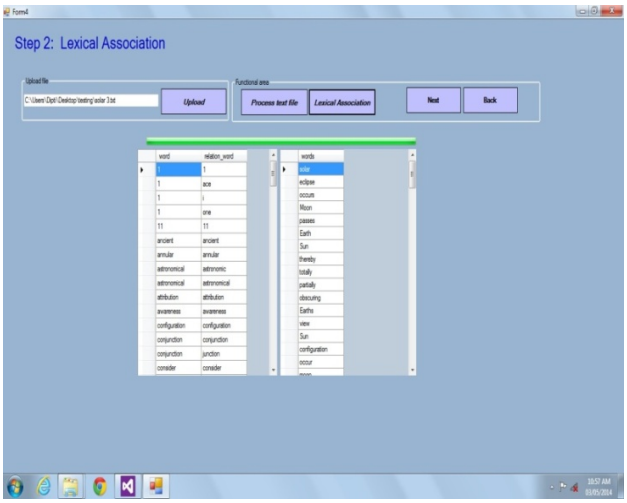

Fig: 3 Output screen for table of lexical association (semantic association)

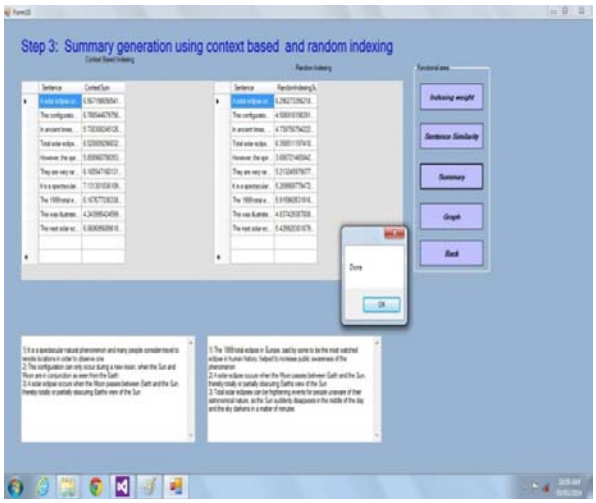

Fig: 6 Output screen shows summary of uploaded text file using both indexing Algorithms

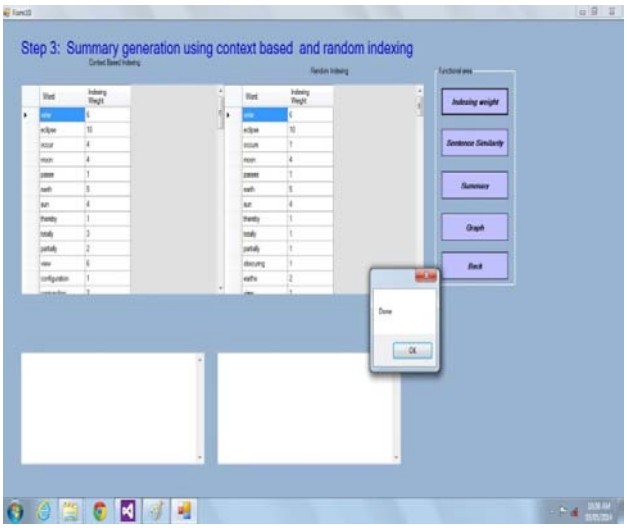

Fig: 4 Output screen for Context based indexing weights and Random indexing weights of document terms

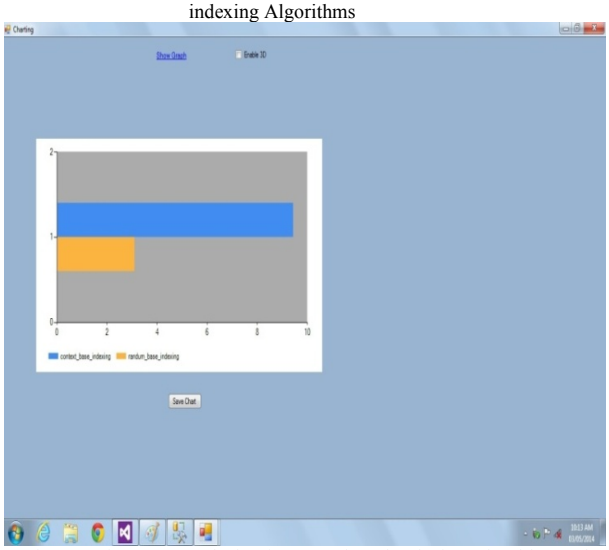

Fig: 7 Performance analysis of Context based Indexing Algorithm and Random Indexing Algorithm with respect to accuracy.

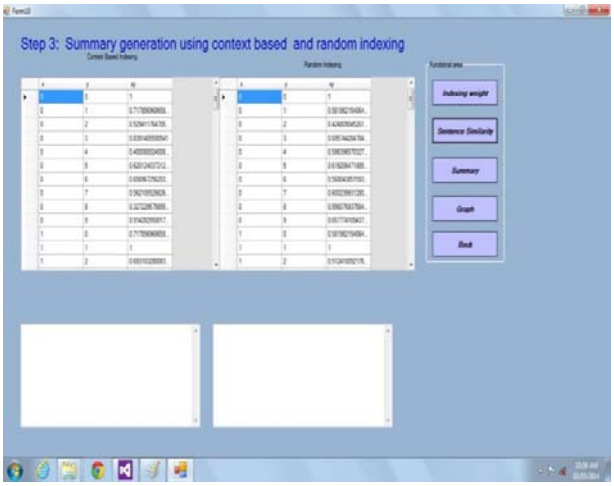

Fig: 5 Shows Sentence Similarity Matrix

V. Conclusion And Future Work

In this work we presented a context based indexing structure in text summarization using Natural language processing (NLP). In particular, we first find out lexical association between document terms in the form of semantically related words and then document graph is built. Then Text rank algorithm is used to calculate indexing weight of terms in document. The context based indexing enables extraction from index on the basis of context rather than keywords. This aids in improving the quality of the retrieved results. In this thesis we have studied the normal indexing and context based indexing and implementation of both indexing methods in text summarization. Also we have compared the performance of Text summarization using indexing and context based indexing method. We have compared algorithms on various parameters like space complexity, time complexity, accuracy, redundancy and the performance of context

based indexing is better than normal indexing. This system can be applied with stand alone machine by retrieving text files within short period of time.

In the future, we plan to extend our work to account for links between documents of the dataset. Also we will try to implement same algorithm in different applications. Furthermore same technique can be applied on different file formats and best indexing method can be suggested for different file formats.

REFERENCES

[1] V.Gupta ,G. S. Lehal, "A Survey of Text Summarization Extractive techniques", Journal of Emerging Technologies in Web Intelligence, Vol. 2, No. 3, August 2010.

[2] D. Sureshrao, S. Subhash and P. Dashore, Analysis of Query Dependent Summarization Using Clustering Techniques, International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 1.

[3] N.Chatterjee, S.Mohan, "Extraction-Based Single-Document Summarization Using Random Indexing".

[4] R. Mihalcea, "Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization", Department of Computer Science University of North Tex asrada@cs.unt.edu.

[5] R. Mihalcea, P.Tarau, "Text Rank: Bringing Order into Texts".

[6] J.Leskovec, N. Milic-Frayling, M.Grobelnik, "Extracting Summary Sentences Based on the Document Semantic Graph", by Jurij Leskovec Natasa Milic-Frayling Marko Grobelnik.

[7] G. Ercan, I. Cicekli, Lexical Cohesion Based Topic Modeling for Summarization", Dept. of Computer Engineering Bilkent University, Ankara, Turkey.

[8] G. G. Chowdhury, "Natural Language Processing".

[9] P. Pecina, "Lexical Association Measures Collocation Extraction".

[10] Stephen, "Vector Space Models of Lexical Meaning".

[11] The complete Reference of .NET - by Matthew, Tata McGraw Hill Publication Edition 2003.

[12] http//www.google.com