International Journal of Computer Engineering & Technology (IJCET)
Volume 6, Issue 10, Oct 2015, pp. 58-65, Article ID: IJCET_06_10_007
Available online at
http://www.iaeme.com/IJCET/issues.asp?JType=IJCET&VType=6&IType=10
ISSN Print: 0976-6367 and ISSN Online: 0976–6375

TEXT SUMMARIZATION USING HIERARCHICAL CLUSTERING ALGORITHM AND EXPECTATION MAXIMIZATION CLUSTERING ALGORITHM

Mrs. Meghana P. Lokhande, Mrs. Namrata Gawande, Mrs. Shweta Koprade
Department of Computer Engineering,
Pimpri Chinchwad COE, Pune, MS

Mrs .M. S. Bewoor
Department of Computer Engineering,
Bharati Vidyapeeth Deemed University, COE, Pune, MS

ABSTRACT

Due to an exponential growth in the generation of web data, the need for text summarization of web documents has become very critical. Web data can be accessed by different ways, large amount of data is available which makes searching for relevant pieces of information a difficult task. It is very complicated for human beings to manually summarize large documents of text. Text summarization plays an important role in the area of natural language processing and text mining. Text summarization is compressing the source text into a shorter version preserving its information content and overall meaning. In this paper, we are implementing initially phases of natural language processing that is splitting, tokenization, part of speech tagging, chunking and parsing. Secondly we are implementing Hierarchical clustering Algorithm and Expectation Maximization Clustering Algorithm to find out sentence similarity. Based on the value of sentences similarity, we are summarizing the text document.

Key words: NLP- Natural Language Processing; Parsing; Tokenizing; Chunking; Document graph.

Cite this Article: Mrs. Meghana P. Lokhande, Mrs. Namrata Gawande, Mrs. Shweta Koprade and Mrs .M. S. Bewoor. Text Summarization Using Hiearchical Clustering Algorithm and Expectation Maximization Clustering Algorithm. *International Journal of Computer Engineering and Technology*, **6**(10), 2015, pp. 58-65.
http://www.iaeme.com/IJCET/issues.asp?JType=IJCET&VType=6&IType=10

1. INTRODUCTION

1.1 Overview

In today's information technology world, number of people are searching for relevant data on web, but every time it is not possible that they could get all the related data in one document. They could get the relevant data on number of web pages. This problem can be resolved by associating to data mining which returns query specific information from large set of documents and can be represented as a single document to the user[3]. Therefore, a twofold problem is encountered. Firstly searching for relevant document and second absorbing a large amount of related information. Summarization is a useful method for selecting related articles, and extracting the important points of each document.

The main aim of research work is to twofold. One is to implement Hierarchical clustering algorithm and Expectation Maximization Clustering algorithm by identifying the nodes. Second is query dependent summarization is done by removing ambiguity. We used Open NLP tool for natural language processing of text for word matching and word net dictionary for interpreting the text [1]. The input text is processed where each sentence in the text document is considered as a single node and each node will be compared with all other node. This comparison can be done by calling word net dictionary where it will calculate weight of each node with every other node. It is shown in the form of document graph. Hierarchical clustering algorithm and Expectation Maximization Clustering Algorithm is used before summarization to generate effective summary. Furthermore performance of the summary on the basis of complexity and accuracy using Open NLP tool and clustering techniques will be analysed. The organization of the paper is as follows. Section II system description and section III shows experimental results followed by conclusion, future work and references.

2. SYSTEM DESCRIPTION

System architecture mainly focues on four modules. A module for uploading and processing text file and making document graph, second module for clustering and making clustered graph, third module for making weighted clustered document graph, the last for generating summary for fired query

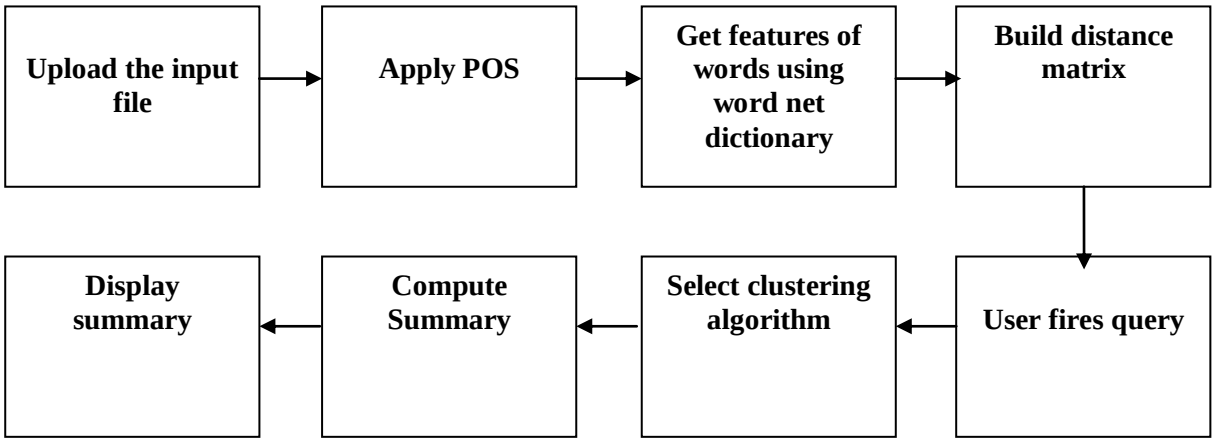

Figure 1 Architecture Diagram

1. Input

Standard input text file containing set of sentences from same context is fed to the proposed system. The text can be preprocessed through NLP phases. This phase is made up of modules like splitting, tokenization, part-of-speech tagging, chunking and parsing. The words like “the”, “him” and “had” called as stop words which do not contribute to understanding the main ideas present in the text can be removed. Distance matrix can be calculated and Hierarchical and Expectation Maximization clustering algorithm are implemented which builds document graph and will generate specific summary. Every node in the cluster maintains association with every other node.

2. NLP Parser Engine

OpenNLP library is a machine learning based toolkit used for processing the text in natural language [10]. OpenNLP can also performs the grammatical checking with the help of wordnet dictionary. It supports the most common NLP (natural language processing) tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing.

2.1 Split Sentence

Its difficult for computer to recognizing the end of a sentence. By using split method, the input data is split into separate sentences by the new line character and converted into the array of paragraphs. This can be done by treating each of the characters '.', '!', '?' as separator rather than definite end-of-sentence markers. The first non-whitespace character is assumed to be the begin of a sentence and the last non whitespace character is assumed to be a sentence end.

2.2 Tokenization

It will separate the input text into separate tokens. Punctuation marks, spaces and word terminators are the word breaking characters. Each sentence in text document is partitioned into a list of words and we remove frequently occurring, insignificant words called as stop words.

2.3 Part Of Speech tagger

Part- of-speech tagging is the process which is implemented after tokenization. The input for a tagging algorithm is a string of words and the output is a single best POS tag for each word. Part-of- speech tags are

NN: Noun

DT: Determiner

VBN: Verb, past participle IN: Preposition

CC- coordinating conjunction

:

2.4 Chunker

Chunker is used to divide text into parts of words. Chunker will form groups like verb group, noun group. Text chunking will divide the input text into phrases and assigns a type such as NP for noun phrase, VP for verb phrase, PP for prepositional phrase where the chunk borders are represented with square brackets. Each word will have only one unique tag.

2.5 Parser

It generates the parse tree for given complete sentence. Parsing is converting a input sentence into a hierarchical structure that corresponds to the units of meaning in the sentencer represented with different phrases.

2.6. Built Distance matrix

Distance matrix shows associativity of different sentences. Here we have to find out organization of the lexicalized concepts that words can express and its relevance with semantic structure.

2.7. User fires a query

We will give distance matrix as an input to the clustering algorithm. Once it is given user can fire a query. Then the similarities between the query and the contents in the clusters is compared. It then build weighted clustered document graph.

2.8. Clustering

Here for each pair of nodes *u, v* we compute the association degree between them, that is, the score (weight) EScore (e) of the edge e (u, v*)*. If Score (e) ≥threshold, then *e* is added to E. The score of edge e (u, v) where nodes u, v have text fragments t(u), t(v) respectively is:

$$\text{EScore} = \frac{\sum ((tf(t(u),w) + tf(t(v),w)) \cdot idf(w))}{size(t(u)) + size(t(v))}$$

Where *t f (d, w)* is the number of occurrences of *w* in *d,*

id f (w) is the inverse of the number of documents containing *w*, and

size(d) is the size of the document (in words).That is, for every word *w* appearing in both text fragments we add a quantity equal to the *tf idf* score of *w.* Notice that stop words are ignored.

1. Hierarchical Clustering Algorithm

Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering is as follows.

1. Start by assigning each node to a cluster, so that if you have N nodes, you now have N clusters, each containing just one node. Let the distances (similarities) between the clusters the same as the distances (similarities) between the nodes they contain.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
3. Compute distances (similarities) between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all the nodes are in single cluster or until some certain termination condition are satisfied.
5. Here termination criteria are if weight is greater than HR Threshold and weight is greater than intra cluster weight.

2. Expectation Maximization Clustering Algorithm

Each cluster is represented mathematically by probability distribution. Each individual distribution is referred as component distribution. Instead of assigning each

object to a dedicated cluster, EM assigns each object to a cluster according to a weight representing the probability.
It has two steps

1. E step: Assign each object to a cluster with a probability.
2. M steps: Re estimates the parameters.

For Probability distribution calculation, one sentence considered as source and another sentence as target. The word count and word occurrences will be calculated for that. The probability distribution is calculated by formula where parameters are word frequency and word count.

2.9. Build Document Graph

The document graph can be build by considering each sentence as a single node. After firing query, generated clusters are processed as follows to create the best query-specific summary.
Show Summary (document graph *G*, query *Q*)

{

1. *Results* = NULL; /*stores summaries*/
2. Find all clusters in *G* that contain some keyword of *Q*; /*use full-text index*/
3. Find all minimal combinations of nodes that when taken together contain all keywords in *Q*;
4. Calculate the best match found for clusters with respect to given query ;
5. Trim *nodes* to make cluster minimal;
6. Display summaries in *Results*;

}

3. RESULT

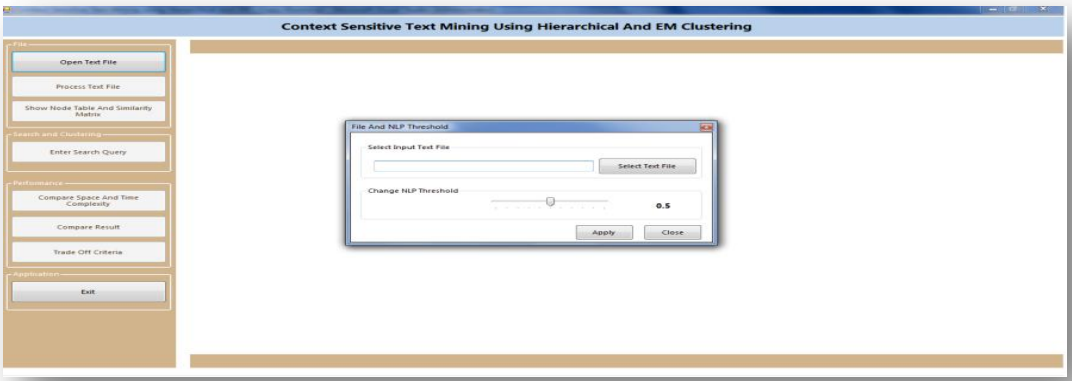

Figure 2 Screen for uploading the text file as input

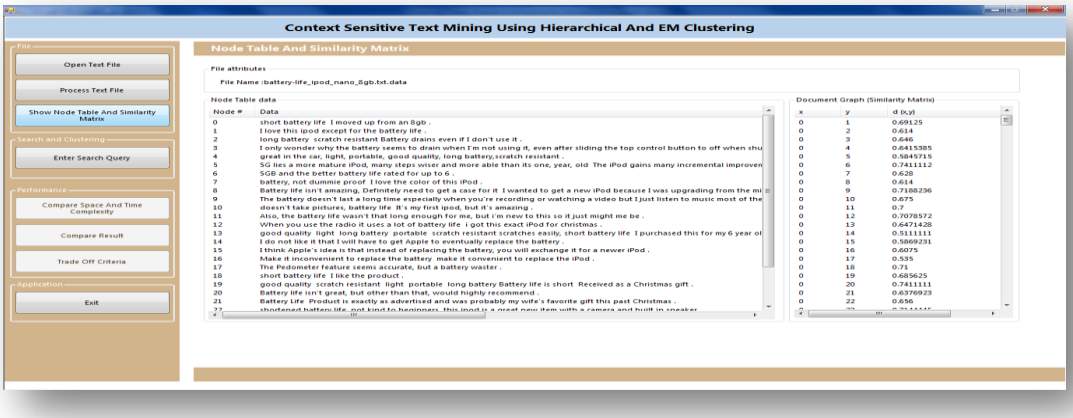

Figure 3 Screen node table and similarity matrix for given text file

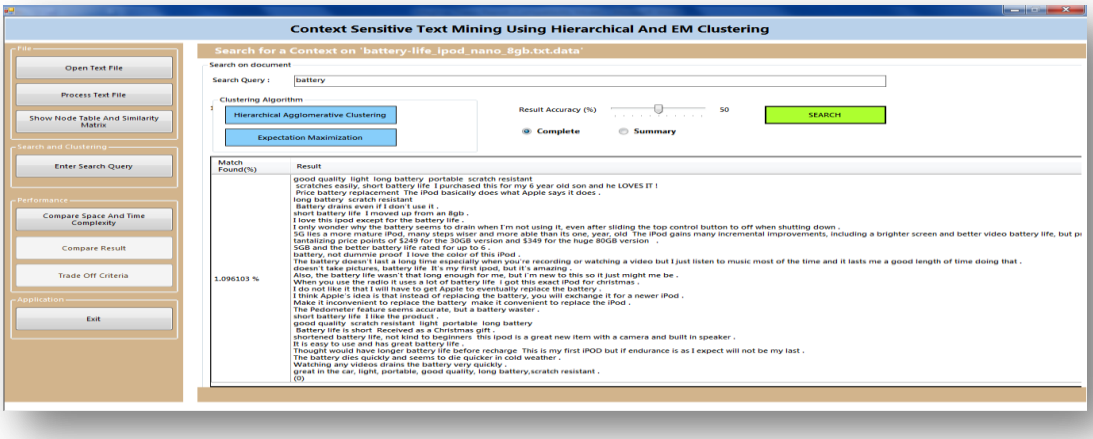

Figure 4 Screen showing Cluster after applying Hierarchical Clustering algorithm

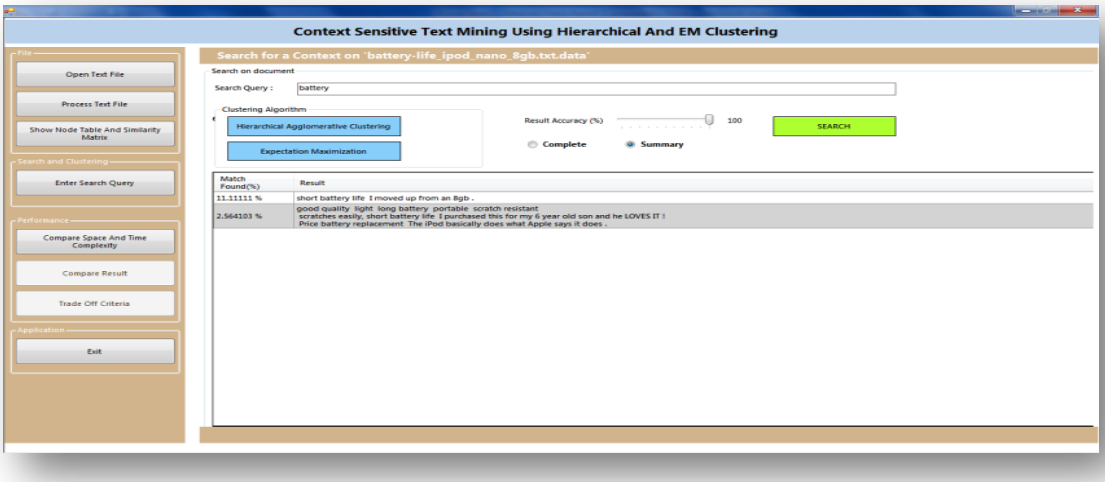

Figure 5 Screen showing summery (Expectation Maximization algorithm)

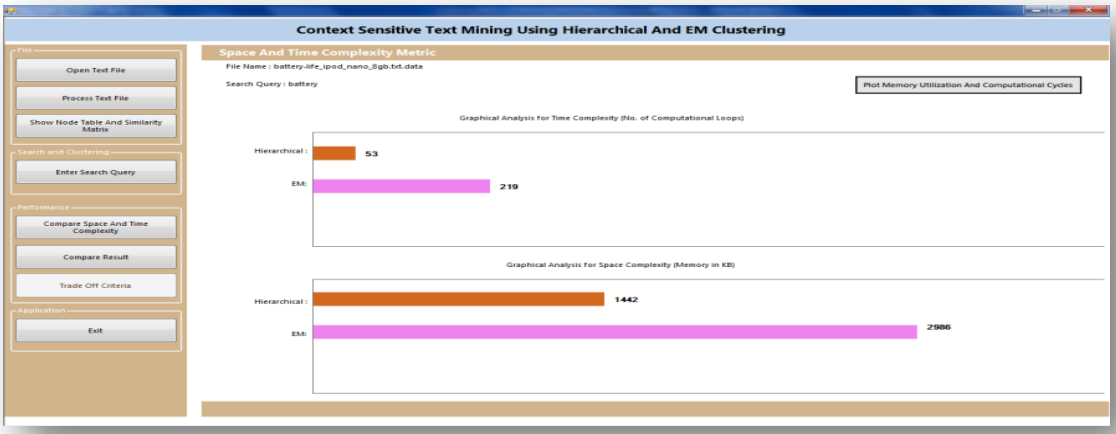

Figure 6 Screen showing space and time complexity for both algorithms

4. CONCLUSION

In this work two different clustering algorithms are used on the same framework and their performance are measured in term of quality of result and execution time. The Hierarchical Clustering algorithm and Expectation Maximization clustering are implemented on document successfully. The performance is measured with respect to time and space complexity. It is concluded that hierarchical clustering needs less space and time complexity as compare to Expectation Maximization Clustering Algorithm. Hierarchical algorithm gives better result compare to Expectation Maximization algorithm.

5. FUTURE WORK

In the future, we plan to extend our work to account for links between documents of the dataset. For example, exploit hyperlinks in providing summarization on the Web. Furthermore, we are investigating how the document graph can be used to rank documents with respect to keyword queries. Finally, we plan to work on more elaborate techniques to split a document to text fragments and assign weights on the edge of the document graph.

REFERENCES

[1] M Ramakrishna Varadarajan, Vangelis Hristidis, “A System for Query-Specific Document Summarization”

[2] Mahmoud El-Haj, Udo Kruschwitz, Chris Fox, “Experimenting with Automatic Text Summarization for Arabic”

[3] “A Novel Technique for Efficient Text Document Summarization as a Service”,2013 Third International Conference on Advances in Computing and Communications

[4] Harshal J. Jain, M. S. Bewoor, S. H. Patil “Context Sensitive Text Summarization Using K Means Clustering Algorithm” International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2, May 2012

[5] “An Efficient Linear Text Segmentation Algorithm Using Hierarchical Agglomerative Clustering”, 2011 IEEE Seventh International Conference on Computational Intelligence and Security

[6] Constrained Text Co-clustering with Supervisedand Unsupervised Constraints", IEEE Transaction on Knowledge and Data Engineering2012.

[7] Parul Agarwal, M. Afshar Alam, Ranjit Biswas ,"Analysing the agglomerative hierarchical Clustering Algorithm for Categorical Attributes "

[8] A Context-Sensitive Clustering Technique Algorithm Based on Graph-Cut Initialization and Expectation-Maximization", IEEE Geosciences and Remote Sensing Letters", Vol. 5, No. 1, January 2008.

[9] Synergy between Object Recognition and Image Segmentation Using Expectation-Maximization Algorithm", IEEE Transaction on Pattern Analysis and Machine Intelligence ,Vol. 31, No. 8, August 2009

[10] http://opennlp.apache.org/

[11] Meghana N. Ingole, M. S. Bewoor, S. H. Patil "Context Sensitive Text Summarization Using Hierarchical Clustering Algorithm", International Journal of Computer Engineering and Technology ENGINEERING (IJCET), ISSN 0976 – 6367(Print), ISSN 0976 – 6375(Online) Volume 3, Issue 1, January- June (2012)

[12] Meghana N. Ingole, M. S. Bewoor, S. H. Patil "Text Summarization using Expectation Maximization Clustering Algorithm", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 ,Vol. 2, Issue 4, July-August 2012, pp.168-171

[13] Neeti Arora and Dr. Mahesh Motwani A Distance Based Clustering Algorithm. *International Journal of Computer Engineering and Technology*, **5**(5), 2014, pp. 109-119.

[14] Deepika Khurana and Dr. M.P.S Bhatia, Dynamic Approach to K-Means Clustering Algorithm. *International Journal of Computer Engineering and Technology*, **4**(3), 2013, pp. 204-219.