AUTOMATION TOOL FOR EVALUATION OF THE QUALITY OF NLP BASED TEXT SUMMARY GENERATED THROUGH SUMMARIZATION AND CLUSTERING TECHNIQUES BY QUANTITATIVE AND QUALITATIVE METRICS

Roma V J\textsuperscript{1}, M S Bewoor\textsuperscript{2}, Dr. S.H. Patil\textsuperscript{3}

\textsuperscript{1}Research Scholar
\textsuperscript{2}Associate Professor
\textsuperscript{3}Head of Department of Computer Engineering
\textsuperscript{1,2,3}(Bharati Vidyapeeth University College Of Engineering, Pune, India)

ABSTRACT

The internet is overloaded with large amount of data due to tremendous expansion of World Wide Web. Query Specific Information Retrieval of relevant documents from large unstructured data stored on the internet becomes a tedious task. This problem can be resolved by using Text Summarization based on Query Specific NLP. Text Summarization is the process of deriving shorter or abstract version of original text, the generated summary should preserve the content and meaning of original text. This paper describes a system, where its first module processes the input text using the phases of Natural Language Processing. The second module applies the document summarization algorithms such as Query specific, Graph based and clustering techniques such as DBSCAN, Fuzzy C-means, Hierarchical and Expectation Maximization on the output distance matrix obtained from the first module and generates the summary of the original input text. The third module evaluates the quality of generated summary based on quantitative and qualitative metrics such as Precision, Recall, F-measure, Compression Ratio, Retention Ratio and CPU processing time.

Keywords: Compression Ratio, Document graph, Document matrix, F-measure, NLP, Precision, Recall, Retention Ratio.
I. INTRODUCTION

The automatic text summarization is a technique through which computer program generates the summary or abstract of an input text. The generated summary should preserve the semantics and central idea of an input text [1]. There are mainly two types of text summarization techniques: query based text summarization and generic text summarization. In Query based text summarization, the keywords of query are compared with the contents of text and only sentences relevant to query are retrieved. In Generic text summarization, only significant sentences are retrieved independent of query. The use of Web search engines has resulted in the enhanced use of query based text summarization. The various IR tools are responsible for retrieving a large number of relevant documents. These documents need to be summarized in order to have fewer burdens over the end user. This system evaluates the query based text summarization, where the sentences are extracted based on the query given by end user. The system implements various techniques for summary generation and also evaluates the quality of summarized text. These evaluation criteria will help the users regarding the selection of particular summarization tool or technique.

II. RELATED WORK

The paper [1] proposes the development of qualitative and quantitative metrics such as precision, recall, compression ratio, retention ratio which can be used for evaluating the quality of generated summary. The paper [2] proposes the creation of query specific summaries by identifying the most query-relevant fragments and combining them using semantic associations within the document, and the best summaries are computed by calculating the top spanning trees on the document graphs. In document summarization graph method, document graph of each text file is generated. For creating document graph each paragraph is assumed as one individual node. Node score and Edge score are calculated using mathematical formulas [3]. The paper [4] describes a simple system for choosing phrases from a document as key phrases. This phrase is made up of modules, splitting, tokenization; part-of-speech tagging; Chunking and parsing. The paper [5] introduced an integrated model for automatic text summarization problem; which tried to exploit different techniques advantages in building of a model like advantage of diversity based method which can filter the similar sentences and select the most diverse ones and advantage of the differentiation between the most important features and less important using swarm based method. Evaluating summaries and automatic text summarization systems is not a straightforward process. What exactly makes a summary beneficial is an elusive property. There are two properties of the summary that must be measured when evaluating summaries and summarization systems - the Compression Ratio, i.e. how much shorter the summary is than the original, and the Retention Ratio, i.e. how much of the central information is retained [8].

III. SYSTEM DESCRIPTION

The system accepts the input text to be summarized in .txt format. The input text is processed using natural language processing in order to get most relevant data. The similarity between words is calculated and document matrix is generated.
Fig (1) Shows the Work flow throughout the System

This generated document matrix is processed by various document summarization algorithms such as Query specific document summarization, graph based method and clustering techniques such as DBSCAN , Fuzzy C-means, Expectation Maximization and Hierarchical method for summary generation. The generated summary from various techniques is compared using metrics such as Recall, Precision, F-measure, Compression ratio, Retention ratio and CPU time.

IV. SYSTEM IMPLEMENTATION

The system comprises of three modules described as follows-

1. Document Matrix Generation through NLP

The system accepts the text to be summarized as .txt file. The input text is processed using OpenNLP tool for which a text has to go through the various phases of Natural Language Processing. Fig (2) shows the various phases of NLP for text preprocessing.

Fig (2) Phases Of NLP

1.1 Sentence Detection

It is used to split the input text into various sentences. The Characters such as ‘.’, ‘?’,’!’ are used as separators. Sentence detection is used to detect various sentences. For example – India is a great country.

India is world’s second most populous country.
1.2 Tokenization
After identifying the sentences, sentences are divided into streams of individual tokens. These tokens are differentiated by spaces.
India is a great country.
India is world’s second most populous country.

1.3 Assignment of POS Tags
After tokenization POS is applied for grammatical semantics. The output of POS tagger is assignment of a single best POS tag for each word such as –
Noun --NN, Coordinating Conjunctions - CC,
Adjective – JJ, Determiner –DT

1.4 Chunking
It is mainly used to find various phrases. Text chunking divides the input text into Various phrases and assigns such as–
NP - Noun Phrase
VP - Verb Phrase
PP - Propositional Phrase
India/NN is/VBZ great/JJ country/NN.
India/NN is/VBZ world/NN ’s/POS second/JJ Most/RBS populous/JJ country/NN.

1.5 Parsing
The parser tags the tokens and groups the various phrases into a hierarchy which will build sentence tree of parse objects.
Similarity between words can be estimated if –
They mean the same thing,
They are Synonyms of each other,
They represent the same way in same context. Thus lexical semantics of the words is practiced.

2. DOCUMENT GRAPHS GENERATION

The document graph is generated using Query specific document summarization, Graph based algorithm, Expectation Maximization, DBSCAN Clustering, Fuzzy C-means clustering and hierarchical clustering techniques. From these documents graphs summaries generated are evaluated and compared by third module of the system. Fig(3) shows the snapshot of output screen through which user can communicate with the system.

2.1 Query specific document summarization
In query specific document summarization, sentences are scored based on query that is frequency of word counts from input text is compared with the words of query. The user inputs a threshold value to the system, if word count exceeds the given threshold value. The sentences containing those respective words will be embedded in the summary.
2.2 Expectation maximization Clustering

It is iterative in nature. It uses two steps- Expectation step - uses the current estimates of parameters.

Maximization step- uses the new estimates of distribution parameters.

Maximum likelihood function is given as below where weight for k distribution parameters is calculated as –

$$ \text{Weight}_k = \sum P_v (a_i \in C_k) / n $$

The sentences with maximum weight will be considered in summary.

2.3 Graph based method

In document summarization graph method, document graph of each text file is generated. For creating document graph each paragraph is assumed as one individual node. Node score and Edge score are calculated using mathematical formulas. Input query is applied on the document and according to that summary from the text file is generated [3].

Two nodes can be connected using edges. Such edge weight is calculated by following formula. Here t (p) means first paragraph and t (q) means second paragraph. Like this edge weights between all paragraphs are calculated and stored in the database. Size t (p) shows number of keyword in first paragraph and t (q) shows number of keyword in second paragraph. Edge weight can be calculated before applying the input query because no. of text files are present on the system.

$$ \text{Edge Score} = \sum \text{word} \in (t(p) \cap t(q)) (t(f(t(p), word)+ t(f(t(q), word).id f(word))/ size(t(p))+size(t(q))); $$

Summary module is referring the concept of spanning tree on the document graph because multiple Nodes may have input query so which nodes will be selected.

$$ \text{Score (T)} = x \sum \text{edge} \in T (1/\text{Edge Score (e)}) + y (1/ \sum \text{node} q \in T \text{ Node Score (q)}) $$

Where x and y are constants assume x=1 and y=0.5. The above equation returns spanning tree score in the document graph. From spanning tree table the minimum score of spanning tree is considered and that paragraph is generated as summary.
2.4 Fuzzy C-means Clustering

The Fuzzy C-means clustering allows the data objects to belong to more than one cluster. This technique is based on the minimization of the following objective function given below –

\[ W(a) = \frac{1}{\sum_{i=1}^{n} c_i(x)} \]

The document graph retrieved from the clustering is traversed using shortest path spanning algorithm and obtained graph is processed using cosine similarity technique [2].

Fig(d) Summary generated by Graph Method

2.5 DBSCAN Clustering

The general idea behind it is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold that is for each data point within a given cluster; the neighborhood of a given radius has to contain at least a minimum number of points.

\[ N_{\varepsilon}(u) = \{ u \in X / \text{dist}(u) \leq \varepsilon \text{-neighborhood} \} \]

2.6 Hierarchical Clustering

A hierarchical clustering method groups the data objects into a tree of clusters. Start assigning each item into a cluster. Find the closest pair of clusters and merge them into single one [4]. This can be carried out either in top-down approach or bottom-up.
3. EVALUATING THE QUALITY OF GENERATED SUMMARY THROUGH METRICS

The summaries generated from second modules are compared with each other using qualitative and quantitative metrics such as Precision, recall, F-measure, Compression ratio, retention ratio and CPU processing time. The summaries generated are also compared with Copernic and Web Summarizers which are query specific document summarization tools. The formulas for quantitative and qualitative metrics are as below:

- **Precision** is the probability that a retrieved document is relevant.
- **Recall** is the probability that a relevant document is retrieved in a search.
- **F-measure** is the harmonic mean of Precision and recall; both have been given equal importance.
- **Compression Ratio** = Number of terms in summary/ Number of terms in data
- **Retention ratio** = Number of correct matching sentences in summary/Numbers of relevant sentences in all data.

The formulas for metrics are as below:

\[
\text{Precision} = \frac{\text{Number different terms in summary}}{\text{Number of different terms in Query}}
\]

And

\[
\text{Recall} = \frac{\text{Number of correct matching sentences in summary}}{\text{Numbers of relevant sentences in all data}}
\]

\[
\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

\[
\text{Compression Ratio} = \frac{\text{Number terms in summary}}{\text{Number total terms in data}}
\]

\[
\text{Retention ratio} = \frac{\text{Number relevant query words in summary}}{\text{Number query terms in data}}
\]

The values of these metrics vary between 0 and 1, with 0 as minimum and 1 as maximum.

Fig(5) Performance Analysis Of Summarization Techniques
V. CONCLUSION

The system aims at the generation of text summary of input text based on the query given by user which preserves the semantics and central idea of original text. The summaries generated through Query specific document summarization, EM, Hierarchical, Fuzzy C-means, DBSCAN and Graph Based method are evaluated and compared with Copernic and Web Summarizer query specific document summarization tools using various quantitative and qualitative metrics.

VI. REFERENCES


AUTHOR’S PROFILE

1. Roma V J, currently Pursuing MTech in Computer Engineering from Bharati Vidyapeeth University College Of Engineering, Pune, India.

2. M S Bewoor is an Associate Professor at the Department of Computer Engineering of Bharati Vidyapeeth University College Of Engineering Pune, India.

3. Dr. S. H. Patil is Head of Department of Computer Engineering at Bharati Vidyapeeth University College Of Engineering Pune, India.