TEXT SUMMARIZATION

[1] Dr. Vinayak Shinde
Head of Computer Engineering department
Shree L. R. Tiwari College of Engineering,
Mumbai University.

[2] Sahil Rathi
B. E. Student
Computer Engineering
Shree L. R. Tiwari College of Engineering,
Mumbai University.

[3] Karan Sharma
B. E. Student
Computer Engineering
Shree L. R. Tiwari College of Engineering,
Mumbai University.

[4] Ashutosh Singh
B. E. Student
Computer Engineering
Shree L. R. Tiwari College of Engineering,
Mumbai University.

[5] Yash Jaiswal
B. E. Student
Computer Engineering
Shree L. R. Tiwari College of Engineering,
Mumbai University.

Abstract: The need for text summarization is crucial as we enter the era of information overload. In this paper, we present a summarization system, which generates a summary for a given input document. This system introduces a new criterion to get better performance in selecting the most important sentences of text for extractive text summarization.

1. INTRODUCTION

There are two kinds of criteria to find the most relevant sentences of text: statistical criteria and semantic relations between text sentences. The proposed technique is a statistical criterion. The idea behind our approach is to consider the position of sentence words relative to words in the sentence occurring in title and keywords. We evaluate this criterion in combination with other statistical criteria. Our system is based on identification and extraction of important sentences in the input document. We listed a set of features that we collect as part of summary generation process. We defined a ranking function which ranks each sentence as a linear combination of the sentence features. We also discussed about the role of NLP in text summarization. The experiments showed that the summary generated is coherent the selected features are really helpful in extracting the important information in the document. Automated summarization of a document is a tedious task.

1.1 DESCRIPTION

The amount of information grows rapidly on the web. As a result, we need text summarization systems to save time and access the main concept of the text in a short time. Text summarization is the process of reducing the length of the original text. Text summarization techniques are classified into two categories: extractive and abstractive. Extractive category selects important sentences of text and concatenates them to form the summary, while abstractive category derives the main concept of the original text. Natural language processing is a tool that is used in abstractive text summarization approach.
This technique applies semantic relations between words to determine the main concept of text. Extractive approach forms the summary of text based on characteristics of sentences. Some of these criteria are: number of words in the sentence occurring in title, sentence length, sentence position, and number of numerical data. After calculating these criteria, they are combined to compute scores of the sentences. Most of these criteria are statistical. In spite of simplicity of these criteria, they eventuate in good results. Also, they do not need an external database to determine the scores of text sentences. In this article, we introduce a new statistical criterion. We evaluate this criterion in choosing the most important sentences of text. The results show that this criterion is useful to select the most relevant sentences of text.

1.2 PROBLEM STATEMENT

"Requirement to build a document summarization product to save time and efforts of people and to use human resources efficiently."

The amount of information is increasing every day. Thus finding relevant data becomes hectic and time consuming, more over not all the data is relevant to the user's topic of interest. In order to find relevant data for user's search and to save time is it necessary to have a small summary of the documents. Summary made by humans is time consuming and tedious. Thus there is a need for automatically summarizing the text document to save time and to get quick results. Automatic Summarization can be defined as the art of condensing large text documents into few lines of summary, giving important information

1.3 SCOPE AND MOTIVATION

The intention of our system, text summarization, is to express the content of a document in a condensed form that meets the needs of the user. Far more information than can realistically be digested is available on the World-Wide Web and in other electronic forms. There are many categories of information (economy, sports, health, technology) and also there are many sources (news site, blog, SNS), it is not possible to read everything one would want to read and so some form of information condensation is needed. So to make an automatically & accurate summaries feature will helps us to understand the topics and shorten the time to do it.

1.4 PROJECT OBJECTIVES

- To develop a system which will summarize a text document.
- To pre-process the text document to be analyzed by text summarization algorithm.

- To create an algorithm for extracting the most important text in the document.
- To create and train an NLP based data set for better sentence extraction. (Stop words, prefixes, suffixes, etc.)
- To define a number of text features which are used for scoring the importance of a sentence in text.
- To calculate score for each sentence of text.
- To select the best sentences for summary.
- To serve the end user with summary of text which the individual has uploaded.

2. LITERATURE REVIEW

Here we will elaborate the aspects like the literature survey of the project and what all projects are existing and been actually used in the market which the makers of this project took the inspiration from and thus decided to go ahead with the project covering with the problem statement.

2.1 LITERATURE SURVEY PAPERS

2.1.1 Subjective measures

Paper of Kumar, Y. J., Goh, O. S., Halizah, B., Ngo, H. C., & Puspalata [7] have proposed a document summarization framework via deep learning model, which has demonstrated distinguished extraction ability in document summarization. The framework consists of concepts extraction, summary generation and reconstruction validation.

A query-oriented extraction technique has been concentrated information distributed in multiple documents to hidden units layer by layer. Then, the whole deep architecture was fine-turned by minimizing the information loss in reconstruction validation part.

According to the concepts extracted from deep architecture, dynamic programming was used to seek most informative set of sentences as the summary. Experiments on three benchmark dataset demonstrate the effectiveness of the framework and algorithms [7].

2.1.2 Extraction Based Measurements

Paper of Jason Weston et al [6] have proposed a supervised learning for deep architectures, if one jointly learns an embedding task using unlabeled data was improved. Researchers used shallow architectures already showed two ways of embedding to improve generalization. First is embedding unlabeled data as a separate preprocessing step (i.e., first layer training) and the second is used for embedding as a regularized (i.e., at the output layer). More importantly, they have generalized these approaches to the case where, have train a semi-supervised embedding jointly with a supervised deep multi-layer architecture on any (or all) layers of the network, and showed have been could bring real benefits in complex tasks..

2.1.3 Behavioral measurements

F. kyoomarsi et al [3] have presented an approach for creating text summaries. Used fuzzy logic and word-net, they have been extracted the most relevant sentences from an original document. The approach utilizes fuzzy measures and inference on the extracted textual information from the document to found the most significant sentences. Experimental results reveal that come within reach of extracted the most relevant sentences when compared to other commercially available text summarizers.

2.1.4 Matrix Based Measurement

Binwahlan et al [4] has incorporated fuzzy logic with swarm intelligence; so that risks, uncertainty, ambiguity and imprecise values of choosing the features weights (scores) could be flexibly tolerated. The weights obtained from the swarm experiments were used to adjust the text features scores and then the features scores and then the features scores were used as inputs for the fuzzy inference system to produce the final sentence score. The sentences were ranked in descending order based on their scores and then the top n sentences were selected as final summary.

2.1.5 Features Based Measurements

Kiani et al [2] proposed a novel approach that extracts sentences based on an evolutionary fuzzy inference engine. The evolutionary algorithm uses GA and GP in concert. The genetic algorithm is used to optimize the membership functions and genetic programming is used to optimize the rule sets. The problem of competing conventions in fuzzy system optimization is thereby reduced by decoupling the two major categories of optimization in fuzzy systems. Fitness function is chosen to consider both local properties and global summary properties by considering various features of a given sentence such as its relative number of used thematic words as well its location in the whole document.

2.2 EXISTING SYSTEM

2.2.1 Maximum entropy-based summarization

During the DUC 2001 and 2002 evaluation workshops, TNO developed a sentence extraction system for multi-document summarization in the news domain. The system was based on a hybrid system using a naive Bayes classifier and statistical language models for modeling salience. Although the system exhibited good results, the researchers wanted to explore the effectiveness of a maximum entropy (ME) classifier for the meeting summarization task, as ME is known to be robust against feature dependencies. Maximum entropy has also been applied successfully for summarization in the broadcast news domain.

2.2.2 Fuzzy Analyzers based text summarization

In this paper a fuzzy query-relevant text summarization approach has been proposed to create text summaries by ranking and sentences extraction. Fuzzy analyzers sort all sentences in terms of their importance. They measure sentence relevancies to identify semantically important sentences. This is an attempt to create a summary with a wider coverage of the document's main content and less redundancy. A summary is obtained by choosing a number of top scoring sentences. Fuzzy analyzers evaluate the sentence in various aspects and infer the rank of all sentences of the text. Since there is no clear formula among the mentioned parameters, some if-then rules (called fuzzy rules) have been extracted to describe the relationships among parameters.

2.2.3 PageRank algorithm

In this article we'll be learning about a very popular and accurate extractive text summarization algorithm. It's called TextRank. Before diving into TextRank algorithm, we must first make sure we understand the PageRank algorithm, because it's the foundation of TextRank.You might have already heard of PageRank since it's used heavily on the service you use most on the web: Google Search. It's used to compute the rank of web pages, but funny enough, it's not named after its use (ranking pages) but rather after its creator: Larry Page, one of Google's founders. Obviously, the algorithm used inside Google is almost too transfigured to be recognised as PageRank. This is due to the fact that inside Google the algorithm works at an incredible scale. We'll be focusing here only on the original, plain PageRank algorithm.

2.3 ALGORITHM USED FOR FEATURE DETECTION

2.3.1 Association of Deep Learning Algorithm with Fuzzy Logic

The proposed algorithm by Megala, S. S., Kavitha, A., & Marimuthu, A. [9] is an association of the Deep learning Algorithm with Fuzzy Logic and it is characterized by two phases. The phases are namely, the training phase and the testing phases. The training phase is used gain the advantages from fuzzy logic and deep learning algorithm to make the text summarization process an effective one. Similar to every training phase, the proposed training phases is also possessed with known data and attributes. Letter to the training phase, the testing phases is implemented to test the efficiency of the proposed approach.

2.3.2 Deep Learning Algorithm

The sentence matrix S=(s1,s2,….sn) which is the feature vector set having element as si which is set contains the all the five features extracted for the sentence si . Here this set of feature vectors S will be given as input to deep architecture of RBM as visible layer. Some random values is selected as bias 3_ where i = 1,2 since a RBM can have at least two hidden layer. The whole process can be given by following equation: S=(s1,s2,….sn).
where si= (f1,f2,….fn), i<=n,
where n is the number of sentences in the document. [9]

Restricted Boltzmann machine contains two hidden layers and for them two set of bias value is selected namely
These set of bias values are values which are randomly selected. The whole operation of Sentence matrix is performed with these two set of randomly selected value. The whole operation with RBM starts with giving the sentence matrix as input. Here (s1,s2,….sn). are given as input to RBM. The RBM generally have two hidden layers as we mentioned above. Two layers are sufficient for our kind of problem. To get the more refined set of sentence features.
RBM works in two step. The input to step 1 is our set of sentence matrix, S==(s1,s2,….sn), which is having the four features of sentence as element of each sentence set. During the first cycle of RBM a new refined sentence matrix set
S'=(s1',s2',….sn'). The above expressed s' is generated by performing:

During step 2 the same procedure will be applied to this obtained refined set to get the more refined sentence matrix set with H1 and which is given by
S''=(s1',s2'',….sn'')

After obtaining the refined sentence matrix from the RBM it is further tested on a particular randomly generated threshold value for each feature we have calculated. For example we select threshold thru a threshold value for the extracted concept feature. If for any sentence f4<thr then it will be filtered and will become member of new set of feature vector.

2.3.3 Title Similarity Feature

The ratio of the number of words in the sentence that occur in title to the total number of words in the title helps to calculate the score of a sentence for this feature and it is calculated by the formula given below

$$Title\ Feature\ (f1) = \frac{S \cap t}{t}$$

where,f1 is the features extracted according to the title similarity of the document [9]. S is the set of words extracted by analyzing the sentences present in each documents and t is the words extracted from analyzing the titles in each documents.

2.3.4 Positional Feature

From the paper published by Megala, S. S., Kavitha, A., & Marimuthu, A. [9] to calculate the positional score of sentence, the proposed approach considers the
following conditions. If the sentence given is in the starting of the sentence or the last in the sentence of the paragraph then the feature value f_2 is assigned as 1. Else if the sentence is in the middle of the paragraph then the feature value of f_2 is assigned as

2.3.5 Term Weight Feature

The Term Frequency of a word will be given by TF (f, d) where f is the frequency of the given word and d is text present the document. [9] The Total Term Weight is calculated by Term Frequency and IDF for a document .Here IDF denotes the inverse document frequency which just implies that the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents holds the term, and then computing the log of that
quotient. The IDF value retrieved by

$$IDF(t, D) = \log\left(\frac{D}{d \in D: t \in d}\right)$$

WHERE, D is the total number of documents ϵD: $t\epsilon d$, it is the number of documents in term t appears. The total term weight is given by TFxIDF
which can be calculated by

$$f_3 \Rightarrow TF \times IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

2.3.6 Concept Feature

[9] The concept feature from the text document is retrieved using the mutual information and windowing process. In windowing process a virtual window of size ‘k’ is moved over document from left to right. Here we have to find out the co-occurrence of words in same window and it can be calculated by following formula,

$$f_4 \Rightarrow MI(w_i, w_j) = \log 2 \frac{P(w_i, w_j)}{P(w_i) \times P(w_j)}$$

where, $P(w_i, w_j)$ - joint probability that both keyword appeared together in a text window. $P(w_i)$- Probability that a keyword w_i appears in a text window and can be computed by

$$P(w_i) = \frac{|sw_t|}{|sw|}$$

where, $|sw_t|$ is the number of windows containing the keyword w_i.$|sw|$- total number of windows constructed from a text document.

2.3.7 Feature Matrix

Here sentence matrix where $S=(s_1,s_2,\ldots.s_n)$ where $s_i= (f_1,f_2,\ldots.f_n)$, $i<=n$ is the feature vector. The five features are the main attributes of the proposed text summarization algorithm. The whole documents under consideration are subjected for the feature extraction and a set of features are extracted accordingly. Now based on the collected features a feature matrix is formed by mapping the features values. The feature matrix is constructed according to the sentences extracted from the multiple documents. In addition to the five features, an additional attribute also associated with the feature matrix. The addition feature associated with the feature matrix is the class labels for each sentence. The Figure represents the feature matrix of the set of documents under consideration

$$S = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ \cdots \\ s_n \end{pmatrix} = \begin{pmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & C \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}$$

Here, the attribute defined by S represents the sentences and Class represents the class values of each sentences. Usually, the class labels are assigned manually by experts from the domains, but in the proposed approach [9]; we define a fuzzy classifier for assigning the class labels for the sentences. The fuzzy classifier assigns the class labels to the sentences according to the fuzzy rules by processing the sentences.

2.4 Proposed System

For summarizing the text there is a need of structuring the text into certain model which can be given to Restricted Boltzmann Machine (RBM) as input.

Restricted Boltzmann Machine

RBM is a stochastic neural network (that is a network of neurons where each neuron has some random behavior when activated). It consists of one layer of visible units (neurons) and one layer of hidden units. Units in each layer have no connections between them and are connected to all other units in other layer as shown below in Figure.

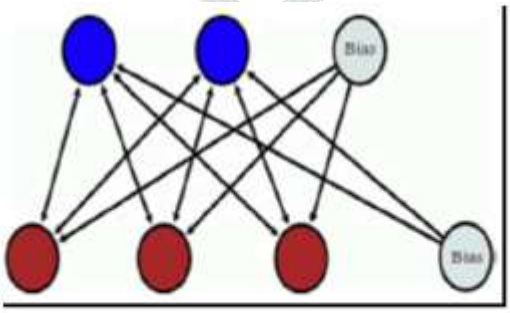


Fig1: Restricted Boltzmann Machine

Connections between neurons are bidirectional and symmetric. This means that information flows in both directions during the training and during the usage of the network and those weights are the same in both directions.

First the network is trained by using some data set and setting the neurons on visible layer to match data points in this data set. After the network is trained we can use it on new unknown data to make classification of the data which is known as unsupervised learning.

During text summarization the text document is preprocessed using various prevalent preprocessing techniques and then it is converted into feature matrix defined over a vocabulary of words. This feature matrix each row will work as an input to our RBM. Based on the structured matrix, the proposed text summarization algorithm uses the fuzzy classifier to assign class labels for the sentences, in order to compute the relevance of each sentence based on the rule selector. The rules are then divided into corresponding sentences and the sentences are then used to form the new feature matrix.

3. REQUIREMENT ANALYSIS AND PLANNING

In requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. Project planning is part which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined

3.1 FUNCTIONAL REQUIREMENT

Requirement Analysis will cover the topics like the Functional, Non-Functional and the specific requirements of the project and touching all the software and the hardware requirements as well.

3.1.1 Text Summarizer Requirements

- The system should provide text parser functions which can take the whole text and separate into sentences, paragraphs and words.
- The system should provide text feature function which can take the necessary part and obtain a feature vector
- The system should provide a well-trained Autoencoder to generate better inputs for classifier.
- The system needs a classifier which is well trained to select summary sentences.
- The system should provide a sentence modifier to beautify and polish output text while changing some words with their synonyms etc.

3.1.2 Summarize Web Page Requirements

- The system should provide a “Web Sayfasını Özetle” button with complete functionality. When clicked on this button, browser extension send the html of the current web page to the server
- A function which detect body part and select text. This function needs to extract unnecessary text from html.
- The system should provide communication between server and client with necessary network functions such send and receive.

3.1.3 Summarize File Requirements

- The system should provide a “Dosyayı Özetle” button with complete functionality. After user selected target file, the user presses the “button and web page application send the file to the server
- A set of functions provide the reading from file depends on file extension
- The system should provide communication between server and client with necessary network functions such send file and receive file.

3.1.4 Summary Setting Requirements

- The system should take parameters such as summary length from user before summarizing.

3.1.5 Train System Requirements

- The system should provide login screen for admin.
- The system should provide taking new data from admin to train Autoencoders or classifiers to improve reliability

3.2 FUNCTIONAL REQUIREMENT

3.2.1 Usability

The system should be easy to use. The user should reach the summarized text with one button press if possible. Because one of the software’s features is timesaving. The system also should be user friendly for admins because anyone can be admin instead of programmers. Training the Autoencoders and classifiers are used too many times, so it is better to make it easy.

3.2.2 Reliability

This software will be developed with machine learning, feature engineering and deep learning techniques. So, in this step there is no certain reliable percentage that is measurable. Also, user provided data will be used to compare with result and measure reliability. With recent machine learning techniques, user gained data should be enough for reliability if enough data is obtained. The maintenance period should not be a matter because the reliable version is always run on the server which allow users to access summarization. When admins want to update, it take long as upload and update time of executable on server. The users can be reach and use program at any time, so maintenance should not be a big issue.

3.2.3 Performance

Calculation time and response time should be as little as possible, because one of the software’s features is timesaving. Whole cycle of summarizing a page/file should not be more than 30 seconds in order to 3 pages long document. The capacity of servers should be as high as possible. Calculation and response times are very low, and this comes with that there can be so many sessions at the same times. The software only used in Turkey, than do not need to consider global sessions. 1 minute degradation of response time should be acceptable. The certain session limit also acceptable at early stages of development. It can be confirmed to user with “servers are not ready at this time” message.

3.2.4 Supportability

The system should require C, Java, Python and Matlab knowledge to maintenance. If any problem acquire in server side and deep learning methods, it requires code knowledge and deep learning background to solve. Client side problems should be fixed with an update and it also require code knowledge and network knowledge.

4. METHODOLOGY

The goal of this project is to explore automatic text summarization and analyse its applications on datasets. The research was done in three phases: a text document collection phase, training phase and testing phase. Below shows the stages of training and testing. To achieve the goal, we completed the following steps:

Step 1: Collection Phase: Collection of document phase I have mentioned in reference part.

Step 2: Training Phase: Each time we fetched binary form of sentences and applied fuzzy rules and calculated weight of each sentence. We had made GUI (Graphical User Interface). We made this GUI using (.NET) Language. Firstly, we made Login page in order to restrict the unauthorized user. After that to show each step of text summarization individually we had decided multiple features to add on it. Those Feature are:

1. Upload File: Here we are allowing authorized used to add one or more valid folder.
2. View File: Here we are allowing the user to select file.
3. Categorization: We now taking the sentence from file and converting them into binary form in order to reduce the complexity and speed up the performance.
4. Pre-Processing: This is the main part of our system or we can call it as ALGORITHM of our system.
   Here I have used Fuzzy Classification. Fuzzy Classification is nothing but set of rules and these rules are provided by us.
   Fuzzy Classification Rules:
   Title Feature: The number of title word in sentence, words in sentence that also occur in title gives high score. This is determined by counting the number of matches between the content words in a sentence and the words in the title. We calculate the score for this feature which is the ratio of the number of words in sentence that occur in the title over the number of word in title.
   Score (Si) = (No. Title word in Si) / (No. Word in Title)
   Sentence Length: The number of word in sentence, this feature is useful to filtering out short sentences such as datelines and author names commonly found in news articles. The short sentences are not expected to belong to the summary. We use normalized length of the sentence, which is the ratio of the number of words occurring in the sentence over the number of words occurring in the longest sentence of the document
   Score (Si) = (No. word occurring in Si) / (No. Word occurring in longest sentence)
   Term Weight: Calculating the average of the TF-ISF (Term frequency, Inverse sentence frequency). The frequency of term occurrences within a document has often been used for calculating the importance of sentence.
   Score (Si) = (Sum of TF-IFS in Si) / MAX(Sum of TF-ISF)
   Sentence Position: Whatever it is the first and last Sentence in the paragraph, sentence position in text gives the importance of the sentences. This feature can involve several items such as the position of a sentence in the document, section, paragraph, etc., proposed first and last sentence highest ranking. The score for this feature:
   Score (Si) =1 for First and Last Sentence
   0 for other Sentence.
   StopWord Cleaning: After a text is obtained, we start with text normalization. Text normalization includes:
   - converting all letters to lower or upper case.
   - converting numbers into words or removing numbers.
   - removing punctuations, accent marks and other diacritics.
   - removing white spaces
   - expanding abbreviations
   - removing stop words, sparse terms, and particular words
   - text canonicalization

These all feature we have achieved by just simply using inbuild function or method in .NET Language.

Stemming: Stemming is a process of reducing words to their word stem, base or root form (for example, books—book, looked—look). The main two algorithms are Porter stemming algorithm (removes common morphological and inflexional endings from words). We achieved stemming by just providing algorithm link in .NET Language.

(Part Of Speech) POS Tagging: Part-of-speech tagging aims to assign parts of speech to each word of a given text (such as nouns, verbs, adjectives, and others) based on its definition and its context. Google provide this facility by just providing algorithm link in system.

Semantic Analysis: Semantic analysis or context sensitive analysis is a process in compiler construction, usually after parsing, to gather necessary semantic information from the source code.

Positive Keyword: Positive keyword is the keyword that is frequently included in the summary.

Negative Keyword: In contrast to, the negative keyword is the keywords that unlikely occurs in the summary.

Step 3: Testing Phase: The experiment conducted in the testing phase compare the three identified and characterized text summarization products, with respect to performance and acceptability.

Performance: As a performance measurement, each product is examined by the degree to which the important sentences extracted (or identified) by the software matches the manually identified important sentences. It is a pure extraction in the sense that the text summarization tools do not perform additional compaction techniques1 on top of the extracted sentences.

Acceptability: Acceptability is measured by a survey to which meeting participants grade the summaries by the means of answering questionnaires. The questionnaire is answered by selecting one of the appropriate weighted choices: Unsatisfactory = 1 point, Somewhat Satisfactory = 2 point, Satisfactory = 3-point, Above Satisfaction = 4 point, and Exceeds Satisfaction = 5 point. The overall grade (total sum of weighted choices attained from the questionnaires) will in turn imply the quality of the summary generated by the software product. Conducting a survey is informative as it removes the subjectivity of the experimenter and allows multiple judgments to be taken into consideration. Respectively, this provides a more accurate and realistic assessment on the experimental evaluation.

Summary: After this we take highest weight sentence from database and combine them to form summarized text.

NLP (NATURAL LANGUAGE PROCESSING) :

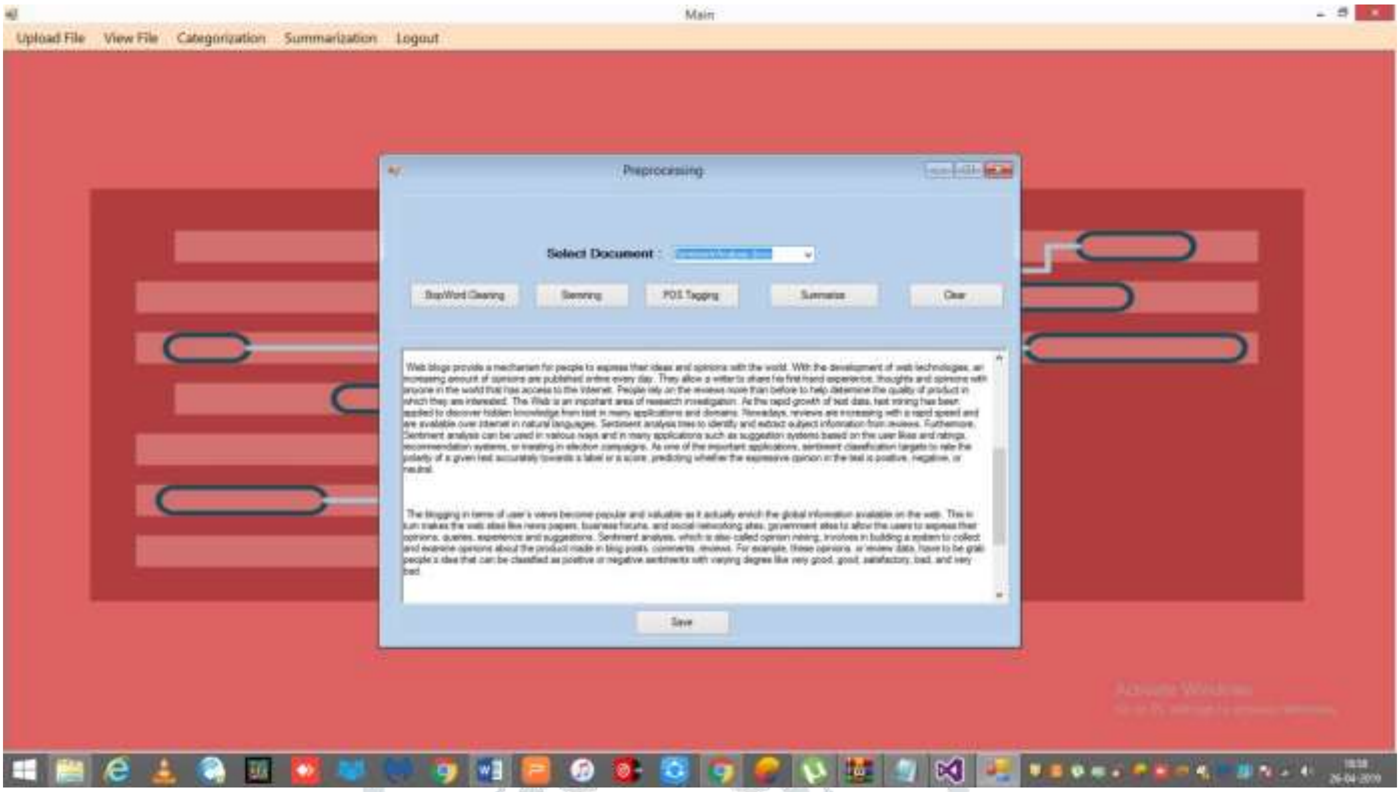
NLP basically we used to make sentence grammatically correct after performing all operations.
NLP (Natural Language Processing) is already provided by Google or we can say it's a in build function that we can use by just importing some libraries and the reference of NLP algorithm's.
In our project we have used in build functionality by just passing peace of reference code in our source code.

5. RESULTS

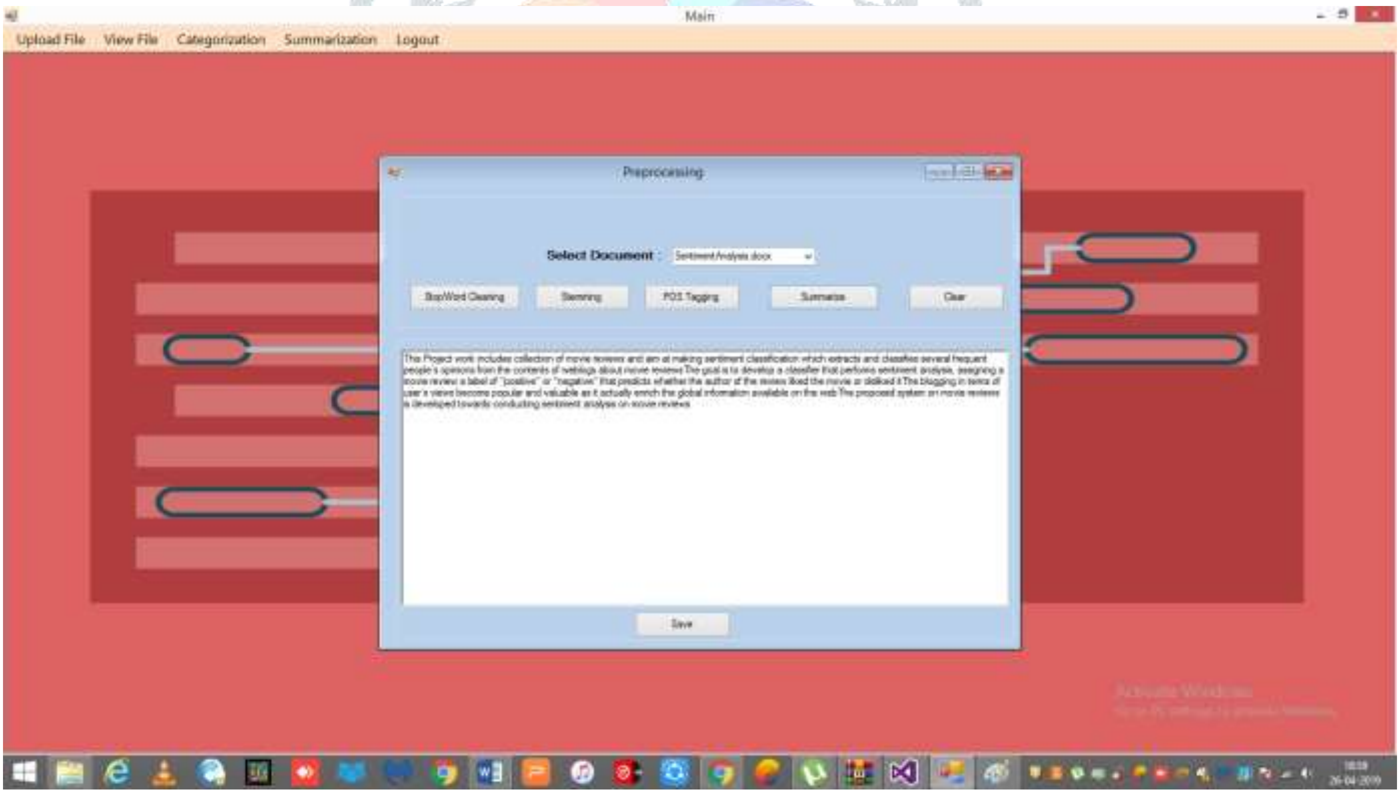
In this section, we show the result of summarization of the text document using the Natural Language Processing Summarizer in (.Net).

LOGIN PAGE:

PREPROCESSING/ FUZZY RULE:



SUMMARIZED TEXT:



6. CONCLUSION

As the data is increasing at an exponential rate, Automatic Summary generator is essential in time bound situations and retrieval of accurate text document. A lot of research has been done on summarization techniques and most of them are extractive summarizers. A hybrid approach of soft computing techniques is proposed using deep neural network and fuzzy logic system. Here the training of sentences over a set of data and applying rules base over it which gives a human reasoning to the summary. Prioritizing the features is best for particular document types where numerical data is very important. We have also used user query based summary extraction, thereby mapping the user query with the sentences using membership function. The proposed system is found to give 84.73% accuracy in giving summary which is to an average 31% more than the individual summary generator. The system scope can be increased by applying back propagation network to the deep neural network. Also a large set of rule base can be applied to the features.

7. REFERENCES

1. Aik, L. E. (2008). A study of neuro-fuzzy system in approximation-based problems. Neuro-Fuzzy System ANFIS : Adaptive Neuro-Fuzzy Inference System, 24(2), 113–130.
2. Albertos, P. (1998). Fuzzy logic controllers. Methodology. Advantages and drawbacks. In X Congreaso Espanol Sobre Technologias Y Logica Fuzzy (ESTYLF), Sevilla, España, pp. 1–11.
3. Babar, S. A., & Patil, P. D. (2015). Improving performance of text summarization. In Procedia – procedia computer science (Vol. 46, pp. 354–363). Elsevier Masson SAS.
4. Dixit, R. S., & Apte, P. S. S. (2012). Improvement of text summarization using fuzzy logic based method. IOSR Journal of Computer Engineering (IOSRJCE), 5(6), 5–10.
5. Fattah, M. A., & Ren, F. (2008). Automatic text summarization. International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2(1), 90–93.
6. Kontostathis, A., & Kulp, S. (2008). The effect of normalization when recall really matters. International conference on Information and Knowledge Engineering (IKE), Las Vegas, NV, pp. 96–101.
7. Kumar, Y. J., Goh, O. S., Halizah, B., Ngo, H. C., & Puspalata, C. S. (2016). A review on automatic text summarization approaches. Journal of Computer Science, 12(4), 178–190. ISSN 1549-3636.
8. Loganathan, C., & Girija, K. V. (2014). Investigations on hybrid learning in ANFIS. International Journal of Engineering Research and Applications, 2(10), 31–37.
9. Megala, S. S., Kavitha, A., & Marimuthu, A. (2014). Enriching text summarization using fuzzy logic. International Journal of Computer Science and Information Technologies, 5(1), 863–867.