Efficient Text Summarization Using Latent Semantic Analysis

ABSTRACT

Text Mining is a considered as one of the most emerging field in the area of Data Mining. With the advancement of technology, Text Mining is becoming more lucrative and challenging. Therefore, the need of a competent text analyzer is necessary. The primary aim of Text Mining is to trying to generate or predict information from unstructured textual data. This field consists of a variety of applications and models out of which Text Summarization is one.

In the current scenario of Information Technology, excessive and vast information is available on online resources but it is not always easy to find relevant and useful information. Automatic Text Summarization is a process to reduce the text in a given document or documents and to generate a good and appropriate summary. It is a process of generating a concise but adequate version of a larger source so that the main concept is retained but the size becomes much shorter. Text Summarization is also an integral part of Natural language processing and Machine Learning.

There are two types of summarization techniques, namely abstractive and extractive summarization. Abstractive are more or less related to the Natural Language Processing (NLP) area where a semantically related summary is generated but it may not contain the exact words or sentences of the main source. Extractive summarization however is a process which involves selecting sentences from the source after assigning scores/ranks to them using specific techniques and then building the summary based on the selected sentences.

The work done in this research is related using the statistical concept of Latent Semantic Analysis (LSA) to identify semantically important sentences. LSA can be considered to be a technique which comes under NLP and it produces relationships between documents. This is done by generating connections between the input source i.e. documents and the terms that they contain and giving a set of concepts which are related to the terms in the documents. The assumption over here is that related words of a concept would always lie close to each other in a document.

There are two models that have been developed in the research work. The first one is related to generating summarization using LSA with the Naïve Bayes Classification. The model that has been developed uses Latent Semantic Analysis technique and chooses

sentences based on specific threshold given by the system. Further, using Naïve Bayes approach of machine learning, the model trains the classifier and predicts the summary that is built on the basis of calculation of Singular Value Decomposition (SVD). Before training the model, it selects two important concepts of SVD - feature ranking and recursive feature elimination. The efficiency of the proposed model is compared with existing models in terms of Recall, Precision and F-score.

The second model which has been named as the 'Hybrid Model' uses the Deep Learning method - Self-Organizing Maps (SOM) which is an unsupervised method and Artificial Neural Networks (ANN) which is a supervised method. The work involves investigating the effect of adding mapped sentences from SOM visualization, and re-training the inputs on ANN for ranking the sentences. In individual experiment of the hybrid model, a different mapping of SOM is added to the ANN network as input vector. Hybrid model uses Stochastic Gradient Descent update set of parameters in an iterative manner to minimize the cost function. In addition, using back-propagation weight is being adjusted for the input vector. The empirical results show that the hybrid model using mapping clearly provides a comprehensive result and improves the F-score.

.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1 Introduction

In the recent days, automated text summarization has drawn a considerable interest among the Natural Language Processing and Retrieval communities. During late 1960s, a large number of scientific papers in American research libraries were to be digitally stored to render them searchable. This created the initial interest for automated text summarization. Locating relevant text materials was an arduous task in the earlier days, when the personal computers were not invented, and the World Wide Web had not emerged as a global digital library.

The form and function of text searching has been altered after the invent of WWW, there by facilitating academicians, researches and lay men alike to browse contents online and get huge benefits. In spite of the reduced burden of information gathering, it is still a challenge to acquire relevant information in a concise manner. This issue can be addressed by text summarization. Summarization is the technique in which a computer automatically creates an abstract or summary of one or more text documents.

The process of automatically constructing summaries for a text based on the needs of users is called automated text summarization. Summary of any text is the accurate representation of the information depending on the specified target compression ratio. Systems that involve summarizing single documents are called single document summarization systems, and those that summarize multiple related sets of documents are called multi-document summarization systems. Whereas document summarization in itself is a difficult task, multi-document summarization faces additional difficulties compared to single document aspects, since it involves many tasks like removal of redundancy among the document sets, handling large number of documents, time stamping etc. Hence, multi-document summarization is quite a challenging task along with being an issue to be focused on at the research level.

The process of summarization can be defined in several ways. According to Mani and Maybury (1999), "Summarization is the process of distilling the most important information from the source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks)". According to Mani (2001) "The goal of an automated text summarization system is to read through the information source, extract and then present the most important content to user in a condensed form and sensitive manner based on the user's or application's need".

1.1 Approaches to Summarization

Automated text summarization (Fattah and Ren 2009, Wang and Wang 2005) is a multi-pronged endeavor that typically branches out in several dimensions. There exists no clear-cut path for summarization systems, which usually tend to fall into several overlapping categories. According to Jones (1998) and (Lin and Hovy 2000), the following inconclusive divisions are roughly made.

Input (Source Text)

Input: - single Vs multi document

- Genre: - news Vs article (Technical paper)
- Classification: - domain Vs general
- Length: - short Vs long

Purpose

- Use: - generic vs query-oriented
- Audience: - technical Vs non-technical

Summary Generation (output)

- Extract vs abstract
- Text, time line series, table

1.2 History of Summarization

The initial research on summarization tasks dates back to several decades and continues to be a steady subject of research, which is relevant until date.

Systems that were developed in the early 1950s exploited thematic features such as Term frequency, Term Occurrence, product of Term Frequency and Inverse Document Frequency, location based features, and presence of background terms like title, cue words and phrases and were termed as surface level approaches. This was followed by entity level approaches based on syntactic relations, similarity relationships, co-occurrence and co-reference that were developed during the 1960s. Later, during the 1970s, entity level approaches called discourse based approaches, which used the rhetorical structure of text and format of the document, were developed.

Traditionally, these earlier approaches to automated text summarization were developed based on the principles that the salient parts of a text can be determined by applying one or more of the following assumptions:

- Important sentences in a text contain words that are used frequently (Luhn 1958).
- Important sentences contain words that are used in the title and section headings (Edmundson 1969).
- Important sentences are located at the beginning or end of the paragraphs (Baxendale 1958, Mitra et al 1997).
- Important sentences are located at positions in a text that are genre dependent, and these positions can be determined automatically through training techniques (Kupiec et al 1995, Lin and Hovy 1997, Teufl and Moens 1997).
- Important sentences use bonus words such as “greatest” and “significant” or indicator phrases such as “the main aim of this paper” and “the purpose of this article”, while unimportant sentences use stigma words such as “hardly” and “impossible” (Rush et al 1971).
- Important sentences and concepts are the highest connected entities in elaborate semantic structures (Skorokhodko 1971, Lin 1995, Barzilay and Ellahald 1997, Mani and Bloedorn 1997)
- Important and unimportant sentences are derivable from a discourse representation of the text (Jones 1993)

Numerous summarization systems that are robust in nature have opted for statistical sentence extraction. Various systems that extract important sentences from the text, in which the importance of the sentence is inferred from low-level properties, which can be more or less objectively calculated have been designed.

Hence, the result of any extraction process leads to the formation of an extract, containing a collection of sentences that are selected verbatim from the text. Present works (late 1990s) represent the revival of all the three types of approaches and are being explored very aggressively, because of the heightened commercial and government interest. Though recent works focus almost exclusively on extracts rather than abstracts, there is a renewed interest in the earlier surface-level approaches too. As more natural language generation work begins to focus on text summarization, the focus on extracts is likely to be changed in the next few years.

The emergence of new areas such as multi-document summarization (Tjhi and Chen 2007), multilingual summarization (Mihalcea and Paul 2005) and multimedia summarization (Murray et al 2009) are also being seen. Not only are the current sentence extraction based approaches dependent on the similarity measures (Aliguliyev 2009, Qiu and Pang 2008), but also adopt the sentence clustering approach (Alguliev and Aliguliyev 2005).

It is a challenging task to identify sentences for a summary with a focus on reducing similarity among the sentences (Binwahlan et al 2009, Hendrickx et al 2009).

1.3 Applications, Advantages and Necessity for Summarization

There are extensive application areas for automated text summarization (Fattah and Ren 2008). The rapid growth of online information renders the task of retrieving relevant information in an efficient way very difficult.

Information is published simultaneously in many media channels in different versions, for instance, a paper newspaper, web newspaper, SMS message, radio newscast, and a spoken newspaper for the visually impaired.

Computer resources and bandwidth can be saved by adopting the process of summarization. For example, if a large document is intended to be translated by the user since he/she does not understand the source language, and then translating the complete document becomes a wasted exercise. Instead of translating the whole document, if only a summary is translated, then the user can assess if it is worth translating the whole document. A similar example is in the case of text-t-speech conversion for the visually impaired, where the text-to-speech conversion can be profitably applied to a summary, before attempting the conversion of the whole text. Hence, it can be seen that though a topic is simple, it becomes difficult to read all the documents related to it. This has resulted in an increase in the demand to condense documents.

The presence of a great redundancy of information in a document set is the most significant characteristic of multiple document summarization as compared with the single one. Thus, in order to manage the large amount of text that people must read, Summarization becomes a very important approach. It enables the reduction in the amount of text the people have to read, thereby letting them decide if a document is relevant to their information need. One of the first tasks undertaken since the inception of using computers to process a written text was that of summarizing the text by shortening a long document, so as to present the document's content briefly, while preserving the underlying meaning.

1.4 Problem Statement

Nowadays, there is a huge data presented at internet and other online resources. To efficiently manage data, there is a need of mechanism to extract sentences from the corpus to summarize a document. It is a process of generating a concise but adequate version of a larger source so that the main concept is retained but the size becomes much shorter. For retrieving information, People widely use internet such as Google, Yahoo, and Bing. Since amount of material on the internet is growing rapidly, for users it is not easy to find relevant and appropriate information as per the requirement. Once a user sends a query on a search engine for data or information then the response is most of the times thousands of documents and the user has to face the tedious task of finding the appropriate information from this sea of rejoinder.

For text summarization, an effective approach is required. The problem is to find most comprehensive technique to retrieve most important sentences from given documents without redundancy. Therefore, it enhances the quality and readability of the summary. The proposed research work is related to efficient text summarization based on improving the output using LSA in conjunction with Naïve Bayes classification and supervised and unsupervised deep learning algorithms.

1.5 Objectives of the study

This research focuses on exploring the role of deep learning and machine learning algorithms along with the statistical technique - Latent Semantic Analysis (LSA). The main objectives are noted below.

- To implement existing approaches for extraction based automatic summarization, to get idea of difficulties and obtain high quality summarizer.
- Investigate the role of Latent Semantic Analysis in summarization to improve the quality of summarization.
- Incorporating the classic Naïve Bayes classification method as part of summarization.
- To analyze the role of machine learning and deep learning algorithms to provide effective extractive automatic text summarization.
- Comparing the model generated summary against human summary and summaries from existing summarizers to evaluate the proposed model.

Chapter 2
Literature Review

2 Literature Survey

Research on automatic text summarization is the need of the present scenario with respect to Information Retrieval and Internet Surfing being the most popular applications. Many methods and approaches are available for information retrieval from various sources [3]. Many techniques have been developed until date on multiple-document summarization. The existing different methods are explained below.

2.1 Graph Based Methods

Rada Mihalcea[1] (2004) Proposed Text Rank method on Graph based method which takes into consideration local vertex-specific information as well as full graph global statistics repeatedly for determining significance of vertex. Below steps are elaborate in summary generation:

a. To build a graph model, from the graph, identify vertices which describe given task as text units
b. Draw edges between text units on basis of common match and compute relationship for each edge
c. We may have weighted or un-weighted edges as well as directed or un-directed graphs
d. In the model, apply rank algorithm and repeat until convergence takes place
e. In this graph method, all vertices will be sorted on score of respectively vertex based on last mark of each vertex. And finally, scores will be used for selection purpose

Julin Zhang [2] (2005) projected Hub/Authority framework on basis of Graph theory. In that method, content feature is merged with surface feature i.e. location and length of sentence, cue phrase etc. For sentence selection purpose, it may extract significant sub-topic features under Hub/Authority framework. In this model, sentences are ranked and final summary will be generated on basis of score of each sentences under the hub and authority score.

Shanmugasudaran Hariharan [3] (2009), projected two primary methods with differences, with or without omitting the nominated sentences. Where this paper concentrates on summarization of news articles with help of graph based methods. With help of adjacency matrix, representation can be one via similarity measures between sentences of documents which is the first step of this Graph based approach. In this approach, two techniques are discussed wherein primary one proposes cumulative sum and second one degree of centrality. With aid of these two methods, a method is proposed by the author for assessing adjacency matrix. Precision and recall have been used for calculating extractive summaries as metrics. This paper presents two metrics: Effectiveness 1 & Effectiveness 2 for evaluating human summaries against system summaries. With the help of discounting method for testing for single and multi-document summaries, after investigating the result, we come to know that the second method is better than the previous method but there are few scopes for improvement in this area.

Khushboo [4] (2010), introduced methodology of Text Rank method by few variances. In said method, it uses shortest path algorithm for generating summaries. Sentences will be selected from path with help of shortest path algorithm, where each sentences may be similar to pervious sentences for generating summaries over choosing top ranking sentences such as Text rank. In first step for representing text, it will build graph model. Text units can be word, phrases, collocation, sentence or others, these will have considered as a Text units and it will be added as vertices for the graph. After completion of the step, score will be calculated with help of ranking algorithm (Graph Bases) such as HITS, Page Rank of each vertex. After finishing the above step, shortest path algorithm will be applied for generating summaries.

Shuzhi Sam ge [5] (2010), proposed hybrid approach for weighted graph model that include two concepts, sentences clustering & ranking for text summarization. In other words, method depends on cluster as well as Graph based approaches for generating summaries for text. There are few steps for this approach -

a. There are two ways first is Graph model for sentence ranking and second is cluster for merging same sentences

b. Clustering of sentences can be completed on basis on Singular non matrix factorization, so there are possibilities of using Latent Semantic Analysis, which has gained popularity nowadays for text summarization
c. In weighted graph model, it reflects discourse association between sentences in order to cluster and rank sentences in a document

Tu-Anh Nguyen-Hoang [6] (2012), proposed method which has three steps, during first step, for the data set, specific structure will be added to every document. Undirected weighted graph can be measured as a structure. For graph, title and sentences will play major role for construction of the graph. In the second phase, Weighted page rank which is Graph based ranking algorithm will be used for calculating score of each sentences of the document. Few sentences are extracted from the document for building summaries of documents for that ranks and scores are considered on the basis of relevant features of the document. In later stage, all different summaries will be merged into a single summary. Finally, MMR(Maximal Marginal Relevance) algorithm is used to form the final extractive summary.

2.2 Cluster Based Methods

Judith D. Schlesinger [7] (2008) has presented CLASSY for multi-document summarization. CLASSY (Clustering, Linguistics, and Statistics for Summarization) is a model of extractive automatic summarization which operates both on single and multi-document summarization. Topic or generic summaries can be produced by this model. It practices language method for trimming, statistical method for scoring and that is why it is known as CLASSY. This technique includes trimming rules to reduce the distance of sentences in the document and the identification of sentences on the basis of importance that are probable to be involved in the summary. The summary is generated for individually document and then summaries are re-arranged and then merged to form the final combined summary. CLASSY construction contains of five steps: to prepare document, to trim sentence (using stop word removal, stemming), to compute score of each sentence, redundancy removal and collection of sentence based on score.

Xiao-Chem Ma [8] (2009) has proposed summarization model, which has three parts: pre-processing, soft clustering and summary generation. The main and the most important portion of system is clustering. In the clustering algorithm, there are four stages: primary is to construct Vector Space Model(VSM), second one is preparing relationship matrix, where third is to set initial parameters and finally, build clusters recursively. For summary preparation, Maximal Marginal Relevance(MMR) has been used so summary sentences will designate the core content of the multi-set of documents and deliver connection with the request which is a query.

Virendra Gupta [9] (2012) has introduced a clear approach for multi document summarization by linking simple summary of the document using the phrase clustering. For clustering, syntactic and semantic analytics both are used for similarity between sentences. Document, sentence reference index, location and concept similarity features, all have been used for generating single document summary. Summaries of single document for sentences are clustered and best sentences from each cluster are used to generating multi-document summary.

2.3 Term Frequency Based Methods

Salton [10] (2005) has proposed method of term frequency inverse document frequency model (TF-IDF), where the mark of a term in this document is the ratio between the amount of terms in this document to the frequency of the amount of documents that contain those terms. Importance of evaluating the expression is given by the principle TFI X IDFI, where TFI is the term frequency of 'I' in the document and IDFI is the inverted frequency in which that term 'I' occurs. Therefore, sentences can be scored for illustration with help computing relevance of terms in the sentence.

Jun'ichi Fukumoto [14] (2004) proposed a technique for multi-document summarization in which an easy strategy to build abstract with help of TF-IDF based extraction is used. Summaries for individual documents are generated and same summaries will be used for generating multi-document summary. The proposed system automatically categorizes a document into three different sub-sets with help of info of high frequency nouns and named object, the categories are one topic, multi-topic type and others. To summarize, the first sentences are take out from each document based on TF-IDF, the position of the sentence and weighing of a sentence. During the next step, needless parts of sentences are discarded. Then all sentences which are extracted are sorted in the original order in a document to generate summarized form of each single document. In the next stage, all extracted sentences are grouped in clusters and the repeated clauses are removed. The remaining clauses are sorted for generating the final summary.

2.4 Latent Sematic Analysis Methods

Shuchu Xiong[11] (2014) proposed a method based on LSA wherein sentence taking out summarizer evaluates a set of summary sentences based on its prediction similarity to that of the full sentences set on the top latent singular vector. There are few steps required to build summary with the help of Latent semantic analysis. First step is applying singular value decomposition (SVD) to document. Second is choosing sentence by its capacity of projection similarity. And finally, LSA-based forward sentence selection algorithm is applied to build summary. Here they have used centroid-based MEAD and MMR (Maximal Marginal Relevance) methods.

Josef Steinberger [12] (2004) shows that basic LSA has two main disadvantages, first is that it uses matching number of dimensions as is the number of sentences that we want in a summary. Second disadvantage is that large index value will not be chosen even when required for the summary. The author has proposed modification in the existing SVD-based summarization. In the proposed method, he recalculates SVD of a term by sentences matrix. For summary evaluation, this paper shows few techniques such as similarity of main topic, Term Significance, etc.

2.5 Machine Learning

Naïve Bayes Methods

Kupiec [13] presented a method that derived from Edmundson [14], that able to learn from data. With help of Naïve Bayes classifier, function categorized each sentence that it is worth to take a sentence as part of extractive summarization. Let s be a particular sentence, S the set of sentences which prepare summary, and F1, F_k the feature.

$$P\left(c \mid x\right) = \frac{P(x \mid c)P(c)}{P(x)} \qquad 1$$

- *P*(*c*/*x*) is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- *P*(*c*) is the prior probability of *class*.
- *P*(*x*/*c*) is the likelihood which is the probability of *predictor* given *class*.

- $P(x)$ is the prior probability of *predictor*.

Score assigns to each sentences by (1). Top most n sentences are extracted on basis of score. For evaluate the system, a corpus of text documents with manual summaries was used. The author manually checks the sentences of actual statement with manual summaries, and prepared a mapping i.e. strict match in sentence, two sentences matching, not match etc. Auto generated summaries evaluated against mapping. System use the position, cue feature, sentence length for evolution.

Aone[32] introduced model as DimSum with using naïve-bayes classifier with special features like term-frequency(tf) and inverse document frequency(idf) to get signature words. The idf was computed from a large corpus of the same domain as the concerned documents. In model, name entity tagger used to find single token from each entity. Model also implemented shallow discourse analysis to maintain cohesion in text. The references were resolved at a very shallow level by linking name aliases within a document like U.S to “United States", or IBM for “International Business Machines". Synonyms and morphological variants were also merged while considering lexical terms, the former being identified by using Wordnet (Miller,[65]).

Chapter 3
Pre-processing Text & Latent Semantic Analysis

3 Preprocessing & Latent Semantic Analysis

Preprocessing is very important work to be done on textual data/documents before performing the actual text mining.

3.1 Read input text file

Read input file from source.

3.2 Pre-process the file

- Remove all unnecessary characters.

In this step, all unnecessary characters like punctuations, symbols will be removed.

- Convert all word into lower case.

All words are converted into lower case with Python built in function lower()

- Split each word by sentence – segmentation.

In segmentation, it is the task where text is divided into word, unit, or topic.

- Tokenize each word using Porter Stemmer.
- Remove all stop words.
- Feature extraction

Occurrence of a word in a file.

It is known as term-document matrix. A mathematical matrix explains the occurrence of term in a collection of text. Word (or n-gram) frequencies are typical units of analysis when working with text collections. It is term-document matrix and a vocabulary list. It converts a collection of text document to matrix of token counts. At process, if system does not provide a-prior dictionary and analyzer then system can use feature selection as equal to vocabulary size of analyzing data. When preparing a matrix, rows represent the document and columns represent terms.

From occurrence to frequencies (tf-idf).

Term frequency-inverse document frequency, is a numerical method to understand importance a word is in corpus. The tf-idf value increases regularly to the number of times a term appears in the document, but is often compensated for by the frequency of the word in the corpus, which helps to adjust the fact that some words appear more frequently in general [wiki]. Different types of tf-idf weighting methods are used for scoring and ranking a document.

$$S = TF * IDF \qquad (2)$$

$$TFi = \frac{Ti}{\sum_{k=1}^{n} Tk} \text{ ,} IDF = log \frac{N}{ni} \qquad (3)$$

3.3 Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is a one of the statistical technique for extracting the meaning of contextual-usage of words by statistical computations applied to a large corpus of text. The principal aim is that the information about all the word contexts in which a given word appear provides a set of mutual constraints that largely determines the similarity of meaning of words and set of words to each other. The adequacy of LSA's reflection of human knowledge has been proven in a variety of ways.

Latent Semantic Analysis (LSA), is a statistical model that compares semantic similarity between fragments of textual information for word usage. It used for improving the efficiency for methods of information retrieval. By using LSA, the problem of synonymy, in which a different word or term can be used to explain the same semantic concept, can be solved. LSA is also used to analyze the relationships between the pair documents and their terms, which it contains by producing a set of concepts related to documents and terms. LSA accepts that words, which are close in meaning, will occur in similar pieces of text. A matrix includes rows and columns, rows will represent unique terms from document and columns will represent each paragraph. A matrix built from text. Moreover, we can truncate rows with the help from Singular Value Decomposition (SVD), which is a mathematical technique, which conserves the resemblance structure among columns.

LSA has three main steps, which are describe below.

1. Creation of Input Matrix
2. SVD- Singular Value Decomposition
3. Selection of sentences

Input Matrix Creation

The input needs to be presented in such a way that computer can understand and do calculation as necessary. For that, representation can be done via matrix. Where columns are represented as documents/paragraphs and rows are represented as unique words/terms, which are appearing in documents. In matrix, a cell indicates the importance of the word in sentence. Various approaches can be used for filling the cell values but words do not appear in all documents, and hence the so created matrix will become sparse matrix [15].

For summarization, the input matrix is significant because it directly effects on result of SVD. As SVD is very complex technique, and the complexity increase with size of input matrix. To reduce the matrix, the words can be reduced by various ways such as removing stop words, punctuation marks, tokenization etc. Various approaches can be used for filling cell values, which are described below.

— Frequency of word: - the frequency of word in sentence value filled in cell values.

— Binary representation: - the value of cell is fill with either 0 or 1 on the being of a word in sentence.

— TF-IDF - TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY. With this method, we can fill cell values. Higher values show that words that are more common appear in sentence but not in the documents. A higher value also indicates that word is more representative for particular sentence.

— Log entropy: - The cell value is filled with the amount of information that can be held by the sentence.

Singular Value Decomposition.

SVD is a statistical model that shows relationship among words/terms and sentences. It decomposes the input matrix into three other matrices as shown below.

A=U∑VT

A= input matrix (m X n)

U = Words X Extracted concepts (n X n)

∑ =Scaling values, diagonal descending matrix (n X n)

VT =Sentences X Extracted Concepts (nXn)

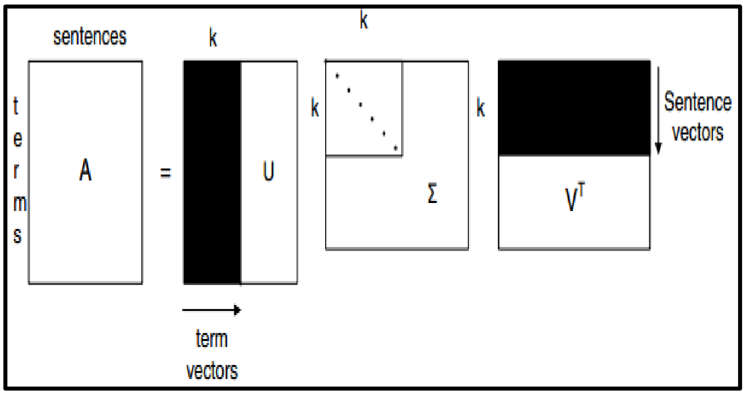


Figure 3.1. Singular Value Decomposition Diagram [3]

Sentences selection

To select relevant sentences using the singular value decomposition results, various approaches and algorithms like Such as Gong & Liu approach, Steinberger & Jezek's approach, Murray, Renals and Carletta's approach are proposed. In the present study, Gong & Liu approach for summarizing the paragraph that uses V_T matrix for sentence selection is adopted.

Chapter 4
Text Summarization using Naïve Bayes Classifier Using Recursive Feature Elimination

4 Text Summarization using Naïve Bayes Classifier Using Recursive Feature Elimination

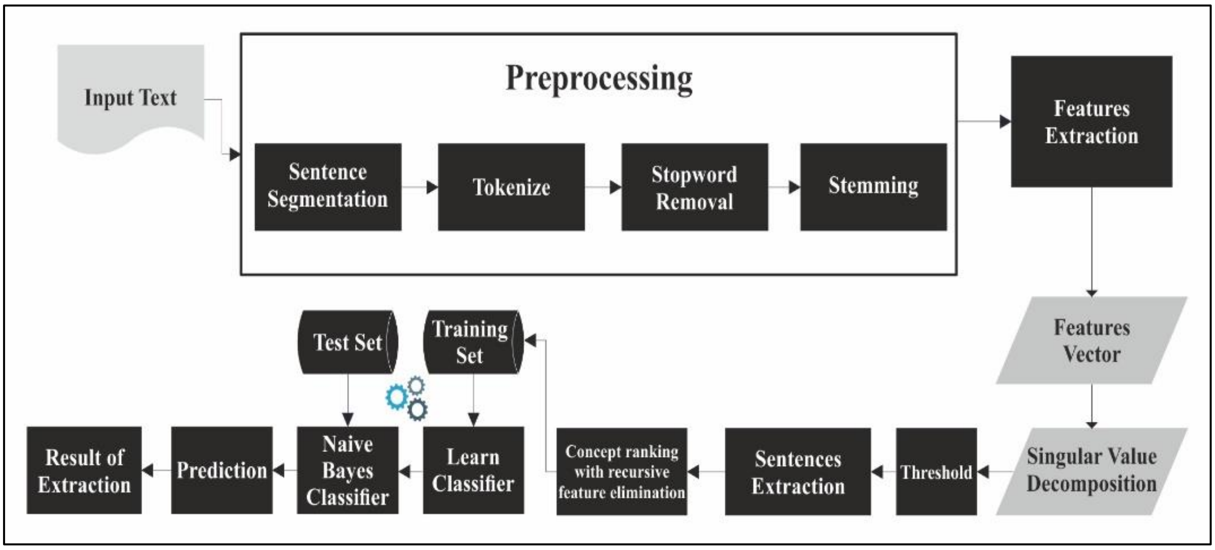


Figure 4.1. The Naïve Bays proposed Model

The aim of automatic text summarization is to generate important sentences for summaries. The proposed method uses statistical method i.e. Singular Value Decomposition (SVD), probability, feature ranking with recursive feature elimination on generated concepts on SVD, Naïve Bayes machine learning algorithm for training documents and prediction. The flow of proposed method is shown below.

Input: An input document

Output: A summarized text as per compression ratio.

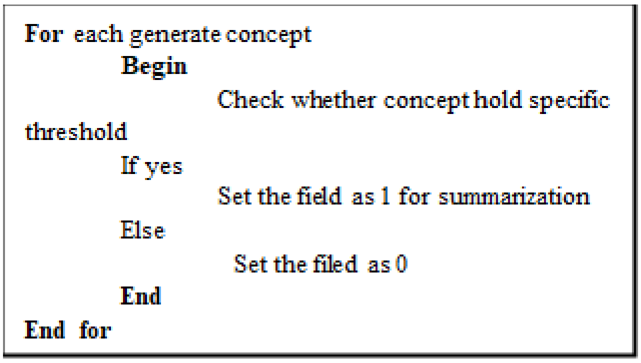
4.1. Preprocessing and Latent Semantic Analysis

As discussed in chapter-3 about preprocessing of text, model performs same text preprocessing as per above steps. Once it complete, it performs Latent Semantic Analysis on corpus and generate V_T matrix.

4.2 Generation of Summary

After performing SVD, VT matrix, the matrix of extracted sentences X concepts is use for picking the significant sentences. In VT matrix, row represents the importance of concepts. The cell values demonstrate the relationship between the sentence and the concept. A higher value indicates that concept is more relevant to concept.

The sentences are marked with 1 or 0 with specific threshold for prediction of extracted sentences.



```
For each generate concept
        Begin
                Check whether concept hold specific
threshold
        If yes
                Set the field as 1 for summarization
        Else
                 Set the filed as 0
        End
End for
```

Figure.4.2. Threshold Selection

4.3 Select important concepts on basis of Feature ranking with recursive feature elimination

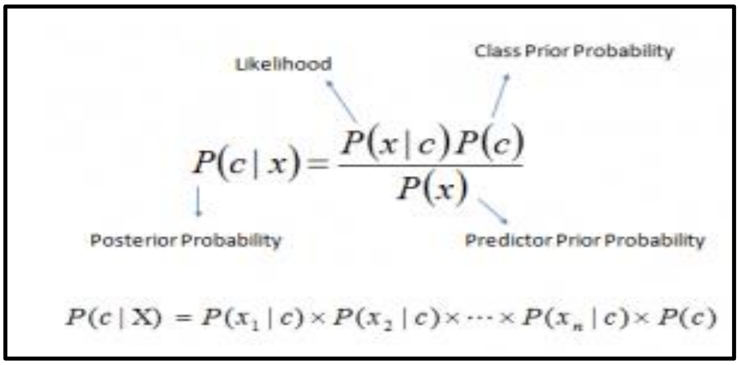
After performing SVD, it populates many concepts on each sentences. Many times, it is not necessary to select all concepts for prediction of summary. Since each concept does not have significant importance, it is necessary to select the important concepts only.

Here, we use recursive feature elimination; it is a comprehensive optimization algorithm, which aims to find the best subset on basis of performance. It iterates through entire list and prepares a model. It keeps aside best and worst performing feature at each repetition. Then it builds next model with rest concept/features until all concepts are exhausted. Then it ranks the concepts based on their order of elimination.

4.4 Naïve Bayes for training & prediction model

Naïve Bayes algorithm is based on Bayes' Theorem. It assumes features are independent. It acquires prior probability and conditional probability on each features. It predicts the class label by highest probability.

As per Bayes' theorem provides a way of calculation posterior probability P (c|x), p(x) and p(x|c).



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 4.3 Naïve Bayes Equation

As per above step of recursive feature selection, dataset divides into sets i.e. training set and test. Then Gaussian Naïve Bayes is applied on training set and train model for prediction. Using train model, we predict on test data for summarization against generate on basis of Singular Value Decomposition (SVD).

4.5 Result Evaluation

Mainly, there are three types of criteria that can be used for evaluation of summaries. (1) Co-selection, (2) Content-based similarity and (3) Relevance-correlation. Co-selection includes precision, recall and F-measure [6]. A co-selection works only on extractive summary. Content-based similarity will check similarity measure in document. It uses word overlap, longer common subsequence, and cosine similarity [5].

ROUGE 2.0 evaluation toolkit works on criteria of intrinsic summarization. It is used to calculate the ratio of how the reference summary overlaps the system summary. ROUGE evaluation measures generate three types of value for each summary. Average precision, average recall and average F-measure.

Precision (Rijsbergen, 1979) defined how many retrieved selected sentences are relevant to user's information.

$$precision = \frac{\left|\{\text{relevant sentences}\} \cap \{\text{retrieved sentences}\}\right|}{\left|\{\text{retrieved sentences}\}\right|} \qquad (4)$$

Recall (Rijsbergen, 1979) defined how many relevant sentence are selected and successfully retrieved.

$$recall = \frac{|\{\text{relevant sentences}\} \cap \{\text{retrieved sentences}\}|}{|\{\text{relevant sentences}\}|} \quad (5)$$

F-measure is consider as harmonic mean (Uddin and Khan, 2007) of precision and recall. F-measure or F-score is define as.

$$F = \frac{(2.\text{precision.recall})}{(\text{precision} + \text{recall})} \quad (6)$$

The text corpus used in this project includes 10 articles from different sources, such as yoga, sports article, movie review, story, etc. The statics of the documents is given below in Table 4.1.

Table 4. 1. Statistics of Documents

Number of Documents	10
Average number of sentences per document	31.5
Minimum number of sentences per document	21
Maximum number of sentences per document	56
Summary of document length (%)	50%
Maximum number of sentences per summary	28
Minimum number of sentences per summary	8

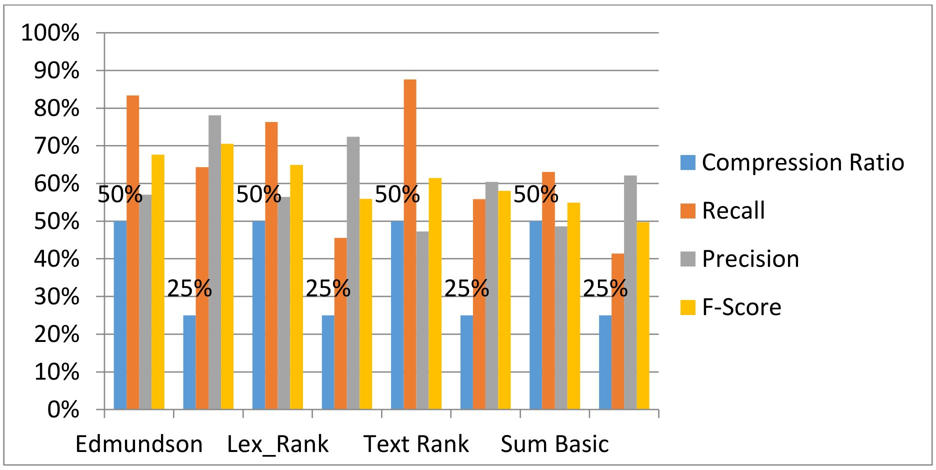


Figure 4.4. Comparison of Different Summarization Tools

Above chart explains the comparison of different algorithms of text summarization. Different algorithms are used and tested on specific documents for evaluation. As per chart, we can see that Text Rank is showing better performance in all parameter such as Precision, recall and F-score. As per figure, it is clearly seen that Text Rank is showing better performance in all cases and next Edmundson is showing great variation for evaluation. We have tested our model against existing techniques and evaluated with different evaluation criteria, which is given in the Table 4.2. The details about the content of the documents in shown in Table 4.1 and the formulas for calculating the Precision, Recall and F-Score is explained in detail before the Table 4.1 is displayed.

Table 4.2 Scores of Each Document

Document	**Precision**	**Recall**	**F-Score**
Document 1	0.83	0.84	0.80
Document 2	0.69	0.83	0.76
Document 3	0.94	0.92	0.92
Document 4	0.95	0.91	0.86
Document 5	0.87	0.80	0.80
Document 6	0.69	0.83	0.76
Document 7	0.80	0.67	0.62
Document 8	0.94	0.90	0.88
Document 9	0.85	0.89	0.82
Document 10	0.88	0.86	0.79

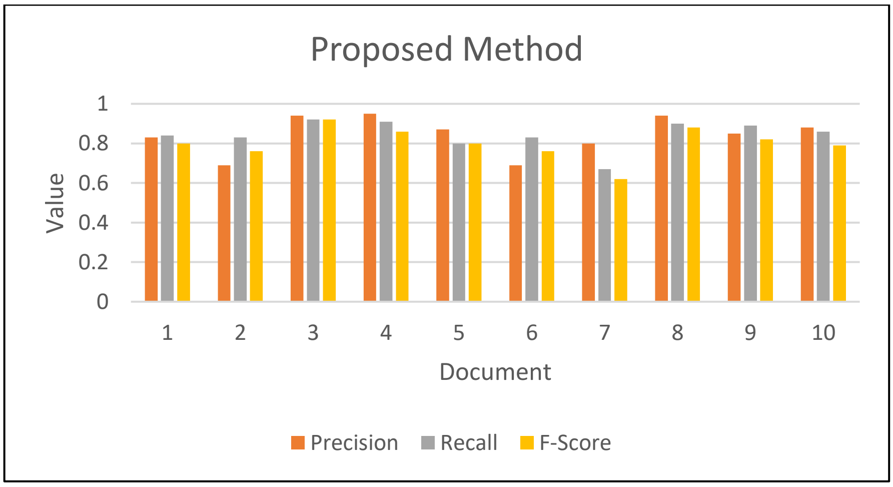


Figure 4.5 Scores of Each Document

As per above figure and comparison with existing algorithms as per Figure 4.4, it is clearly seen that in most document, the results are showing a better performance in comparison with the existing algorithms. Document 3, 4 & 8, achieved more than 80% precision, recall and F-score where as in few cases, performance of model shows less performance in document 2 and 7.

Chapter 5
A Hybrid Approach of Text Summarization Using Latent Semantic Analysis and Deep Learning

5 A Hybrid Approach of Text Summarization Using Latent Semantic Analysis and Deep Learning

The Hybrid model has been divided into number of steps. The algorithm of this model is described as follows:

Step 1: Analyze the text

- Sentence segmentation, based on the boundary
- Tokenize, broken into words
- Stop word removal
- Stemming

Step 2: Feature Extraction & Feature Vector

- Compute frequency of occurrence(fi) of each term (ti) which is appearing in the document (TF)
- Computer Inverse document frequency (IDF)

Step 3: Latent Semantic Analysis

- Input matrix creation, where columns are sentences and rows are words
- Calculate Singular Value Decomposition, $A=U\sum VT$,

where A is input matrix, U is Extracted Concepts X Words, ∑ Scaling values. V_T Sentences X Extracted Concepts

Step 4: Threshold

- Apply specific threshold on V, where $t < 0.5$

Step 5: Self Organizing Maps

- Initialize each node's weights
- Select random vector
- Consider Best Matching Unit (BMU), using Euclidean distance find similarity between two setsD = i=0i=n(Vi-Wi)2
- Deterring the BMU neighborhood t=0(-t)
- Modify Node's weights

$W(t+1)=W(t)+ \odot(t)L(T)(V(t)-W(t))$

Step 6: Mapping

- Returns a dictionary Wm where Wm[(i,j)] is a list with all the patterns that have been mapped in the position(i,j)

Step 7: Artificial Neural Network

- Randomly initialize weights, where weight < 0

- Input the observation
- Forward propagation
- Generate error
- Back propagation

Step 8: Sentence Score

- Sort the sentences with the score in descending order

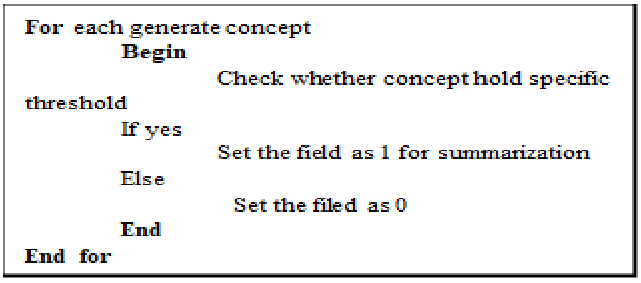
5.1 Preprocessing and Latent Semantic Analysis

As discussed in chapter-3 about preprocessing of text, model performs same text preprocessing as per above steps. Once it complete, it performs Latent Semantic Analysis on corpus and generate V_T matrix.

5.2 Generation of Summary

After performing SVD, V_T matrix, the matrix of extracted sentences X concepts is use for picking the significant sentences. In VT matrix, row represents the importance of concepts. The cell values demonstrate the relationship between the sentence and the concept. A higher value indicates that concept is more relevant to concept.

The sentences are marked with 1 or 0 with specific threshold for prediction of extracted sentences.



```
For each generate concept
        Begin
                Check whether concept hold specific
threshold
        If yes
                Set the field as 1 for summarization
        Else
                 Set the filed as 0
        End
End for
```

Figure 5.1. Threshold Selection

5.3 The SOM Algorithm

The Self-Organizing Maps, which was first introduced by Kohonen [16], SOM is an unsupervised deep learning algorithm which is use in Text Mining, data mining, visualization for data, image & speech recognition, medical & medicine industry and natural language processing [17]. The SOM, that maps M-dimensional input vector a_j to two-dimensional neurons or maps as per their features. It converts the high-dimensional data into the map which groups the similar data together, that help us to explain high-dimensional data. The SOM has two layers. In first, it includes input space whereas in second, which consist output space.

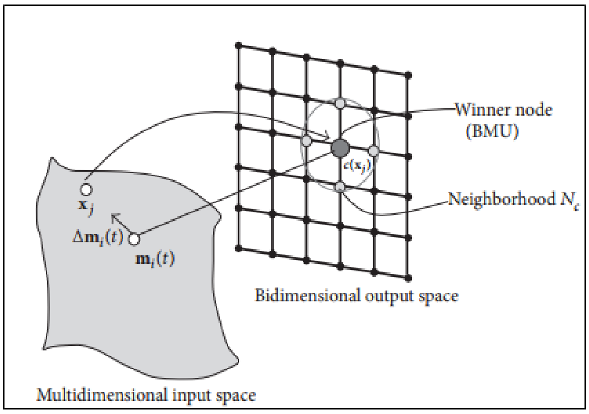


Figure. 5.2 SOM training algorithm

Above figure 5.2. explains the SOM with output nodes in two-dimensional grid view. SOM includes P units, index i of each unit is associates with M-dimensional vector mi in the input space and vector on a low-dimensional regular grid, ri, in the output space. SOM algorithm steps are given below.

1. Initialize each node weights at random value. In absence of prior information, the value of node can be random or linear and are adjusted after network learns.
2. Choose arbitrary vector from given training data and present to Self-Organizing Maps (SOM)
3. Find the Best Matching Unit (BMU), calculate distance between node weights (w1, w2, w3 ….,wk) and input vector (V1,V2, V3 …, Vj). Vector aj is compared to all possible vectors aj and Index c(ai) of the Best Matching Unit(BMU). The find smallest Euclidian Distance (measurement of similarity between two dataset) of BMU, that is, chose original vector nc which is closet to aj as follows

$$\left\| a_j - n_c \right\| = \min_i \left\| a_j - n_i \right\| \qquad (7)$$

4. Determining the BMU neighborhood, on every iteration an exponential decay function shrinks the size of neighborhood until it becomes BMU itself.
5. Update the Best Matching Unit (BMU) and its neighbors. Tuning of vector and winning node and its neighbor updates as

$$n_i\left(t+1\right) = n_i(t) + \Delta n_i(t) \qquad (8)$$

Where $t = 0,1,2....$ is an index of time. The value of $\Delta n(t)$ as computed as follow.

$$\Delta n_i(i) = \alpha(t) h_{ci}(t)(a_j(t) - n_i(t)) \qquad (9)$$

Where (t) is the learning rate and hci(t) the neighborhood function. The learning rate is remain between 0 and 1 and it will decreased during the learning phase. The hci(t) neighborhood function determines the distance between nodes and indexes in the output layer.

During the training phase of SOM. Step between 2 and 4 will repeat iterations until the Vector ai represent. The input patterns that are closer to node for two-dimensional map. After initialization of model, it can be trained either sequential or batch manner [18]. Here, sending all data of vectors to the map for weight updating. One data vector sent to model. After SOM is ready, one vector is mapped to one neuron of the map, which will reduce high dimension (input) space to low dimension (output) space.

SOM-Based Text Summarization

Text summarization is an unsupervised method used to extract sentences from document based on similarity between documents. Suppose α = {d1, d2, … , dN} be a collection of N sentences to be summarized. The purpose of text summarization is extracting meaningful sentences from corpus.

Text summarization using SOM can be divided into two main stages [19,20]. The first stage is document preprocessing and second, text summarization.

Document preprocessing: - preprocessing of document is very important for text summarization. With help of preprocessing, unnecessary characters, token will have removed from document. Segmentation, stop word removal and tokenize can perform in preprocessing stages which is explain in preprocessing stage.

In second stage, TF-IDF(Term Frequency-Inverse Document Frequency) will calculate for document. Once it is completed, SVD (Singular Value Decomposition) is perform to extract sentences from corpus. Details of TF-IDF and SVD is explained in above section.

Training SOM: - after obtaining feature vector aj from calculating Singular Value Decomposition (SVD) and threshold on concept selection associated with text document dj, on the selected feature vectors, SOM algorithms can be applied as discuss in above section. Here, in text summarization, MiniSom[28] is used to initializes a Self-Organizing Maps(SOM). System has to use 10 X 10 dimensions of the SOM. Also in SOM, need to provide number of elements of the vector to be trained for SOM, that numbers can be random as per selection of Vector aj. Another argument, that is, sigma – spread of the neighborhood function (Gaussian), needs to be adequate to the dimensions of the map.

At the iteration t we have sigma(t) = sigma / (1 + t/T) where T is #num_iteration/2. Learning_rate - initial learning rate (at the iteration t we have learning_rate(t) = learning_rate / (1 + t/T) where T is #num_iteration/2) decay_function, function that reduces learning rate and sigma at each iteration.

SOM Visualization

SOM is very useful tool for visualizing high-dimensional data to low-dimensional data. For Visualizing SOM, it uses both the matrix i.e. Unified distance matrix [21] and Component Planes [22]. The distance between neighboring maps can be achieved by U-Matrix, these distance can be visualizing using different color scale on the map.

The U-matrix technique is a single graph that shows the group borders according to the differences between neighbor's units. The distance range of the U-matrix that can be displayed on the map is represented by different shades of gray. Large distances depicted by white color, that is, large gap exist between the vector values in input space. Whereas grey color represents small distance, that is, units are closed together. Visualizing the input data without having prior information about the input, U-matrix is versatile tool for visualizing.

Another component of visualizing input is Component Planes, which is a grid whose cell contain the value of input vector that is, displayed by different type of colors. Which through, analyze of contribution of individual variable in summarization and correlation between different variables.

After applying SOM Algorithms with specific arguments, Self-Organizing map is ready with plotting.

Mapping

During mapping phase, the relation between input vector and output layer nodes can be determined, after that all input vector or pattern can be mapped on to the output layer nodes, after training stage of SOM.

For SOM method, it requires pre-defined size and structure of the network. Various methods are available to achieve same purpose. To prepare network more enhance for the simulation of input space so system have to add rows or columns dynamically to the network.

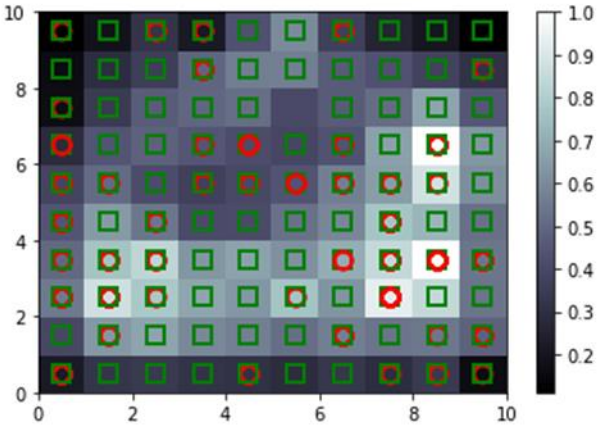


Figure 5.3. SOM Visualization

As shown in figure 5.4, green square indicates that sentences are part of summary whereas red circle shows that it is not part of summary. However, as per figure, there are sentences that come together in same area, which means, green squares and red circles are falling in one common area. As per strategy, system has to mapped those sentences for re-training.

For re-training, supervise method of deep learning that is ANN (Artificial Neural Network) is used.

5.4 Artificial Neural Network

Neural Network Training

During first stage, neural network learns for which type of sentences are include in summary. This task is complete by training network with sentence, where sentence is check to add as part of summary or not. It is achieving by taking inputs mapping which is done in self-organizing maps and extracting sentences from after performing SVD, in another words, re-training of all input vectors. Then neural network learns from input vector and that inherent sentences should be included in summary or not. This model, use three layer to feed network to prove approximation of universal function [23]. This Artificial Neural Network (ANN) uses 8 input layers, 6 hidden layers and 1 output layer to predict sentence as part of summary or not. In addition, model use Stochastic Gradient Descent function where cost function is combination of actual value and predicted value. The goal of training is to minimize loss and improve accuracy of function. Adding penalty feature units, the associated weights unnecessary connections to very small values, while strengthening the rest of the connections. Unnecessary connections and neurons are pruned from network without affecting the performance of network. Following steps are algorithm of Artificial Neural Network with Stochastic Gradient Descent.

1. Initialize the weight with small random value, which is close to 0 and 1.
2. First observation of dataset input in the input layer, each feature is one input node.
3. Forward-propagation: from left to right, neurons are activating; effect of neuron's activation is restricted by weights. To get predicted result, propagate the activation function.
4. Compare actual result and predicted result to generate error and measure error.
5. Back-propagation: from right to left, back propagate the error. Change the value according to error. Through learning rate, it decides how much update the value.
6. Repeat step 1 to 5 and update the weights after each observation
7. Epochs, when the whole training set pass through the Artificial Neural Network (ANN). Redo more epochs.

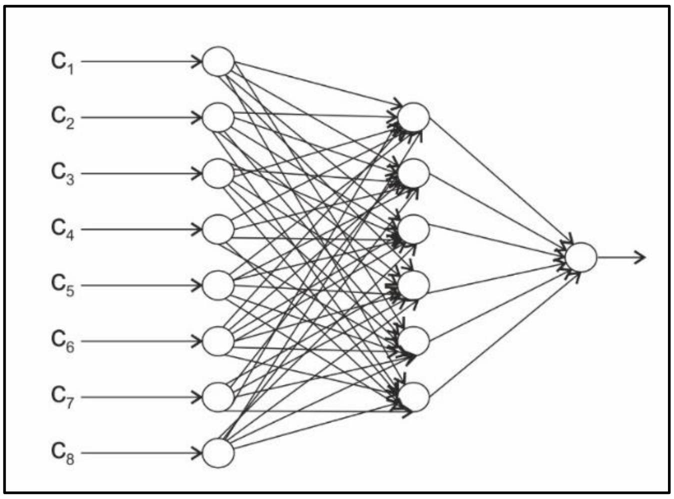


Figure.5.4 The Neural Network After Training

Feature Fusion

Once the model learns, we need to find the trend and relationship features that characteristic in the majority of sentences. Feature fusion completed by this task, which has two steps: 1) removing uncommon feature 2). Collapsing the effect of common feature

Model is train using a cost function in stage 1. Connection pruned from network if it has a very small weighs, which does not effect on the performance of network. As an outcome, the neurons as input and hidden layer network, which has a small value in connection will pruned from network without affecting the performance of the network.

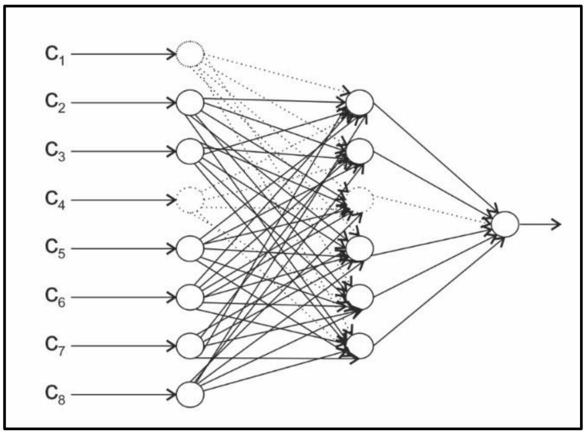


Figure 5.5. The Neural Network before Pruning

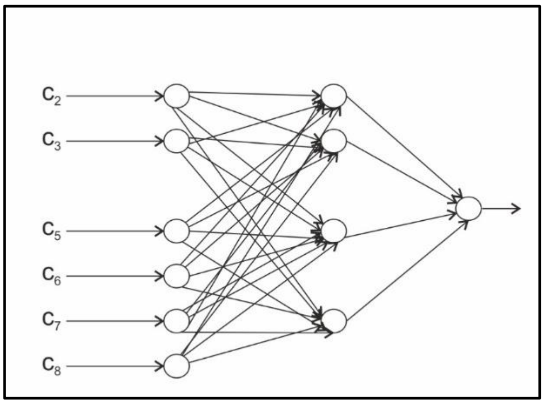


Fig. 5.6 Neural Network after Pruning

The hidden layer transforms the input into some meaningful that output layer can use for prediction. The hidden layer activation values for each hidden layer neuron are clustered using an adaptive clustering technique. The mixture of two steps, which corresponds to generalize effect of features and providing control parameter for sentence ranking.

Sentences Selection & Ranking

Sentence ranking is to consider to part of extracting text summarization. Our model ranks the sentences based on relevance and semantically similar to context previously selected sentences. When the model is trained, pruned and simplify, it uses for determine that whether sentence is part of summary or not. The model will return array of probabilities for each sentences, according to probabilities of sentence, the model will add sentence into summary. For multiclass classification, the array need to transfer to single class prediction class using argsort function.

5.5 Evaluation of Output

The proposed system of single document text summarization was evaluated on standard corpus of Opinosis [24]. The corpus contains 51 documents along with 5 human summaries from different sources.

To evaluate the quality of generated summaries various parameters are used. There are multiple types of way to evaluate the quality of a summarization system (Jing et al., 1998; Neto et al., 2000; Santos et al., 2004). Two types of approaches are available for evaluation: intrinsic and extrinsic. In intrinsic, the sentences summary is evaluated based on the content summary's analysis whereas in extrinsic summary, summarization quality is checked based on task-based, also, it checks the usefulness. We have used Recall Oriented Understudy for Gisting Evaluation (i.e. ROUGE) toolkit to compare human and system generated summary.

ROUGE 2.0 is an intrinsic evaluation toolkit for summarization. It compares an automatically system generated summary against reference summary (human summary). In toolkit, the human summary is taken as reference summary and generated summary is considered as system summary. The ROUGE evaluation parameter (version 2.0.) generates precision, recall and F-score for each summary.

Precision (Rijsbergen, 1979) is defined as how many retrieved selected sentences are relevant to user's information.

$$precision = \frac{|\{relevant\ sentences\} \cap \{\ retrieved\ sentnecnes\}|}{|\{retrieved\ sentences\}|} \qquad (10)$$

Recall (Rijsbergen, 1979) is defined as how many relevant sentences are selected and successfully retrieved.

$$recall = \frac{|\{relevant\ sentences\} \cap \{\ retrieved\ sentnecnes\}|}{|\{relevant\ sentences\}|} \qquad (11)$$

F-measure is considered as mean (Uddin and Khan, 2007) of precision and recall. F-measure is defined as:

$$F - measure\ = \frac{(2.precision.recall)}{(precision+recall)} \qquad (12)$$

These parameters help us to compute how closely similar are the system summary and human summary. (Garcia-Hernandez et al., 2009; Lin, 2004). For text summarization evaluation, ROUGE (http://rxnlp.com/rouge-2-0/#.WtgygIhubIU) is the main metric. The number of overlapping words between system summary and human summary to be evaluated by the counting measures. In our experiment we have used ROUGE-1,2 & ROUGE-L, SU4 to computer average recall, average precision and average F-measure. ROUGE-1 and ROUGE-2 work well with single document summarization. Unigram is considered as type of ROUGE – 1 and bigram as ROUGE - 2. The quantity of words in P (reference summary) that also available in Q (system summary) is called Unigram recall. Whereas unigram precision is that quantity of words Q also available in P. ROUGE-L is called the LCS - Longest common subsequence. It takes longest length in the given P and Q, and finds similarity and recognizes longest co-occurring sequence in n-grams. ROUGE-SU is a skip-bigram plus unigram based co-occurring statistics.

Experiment Results

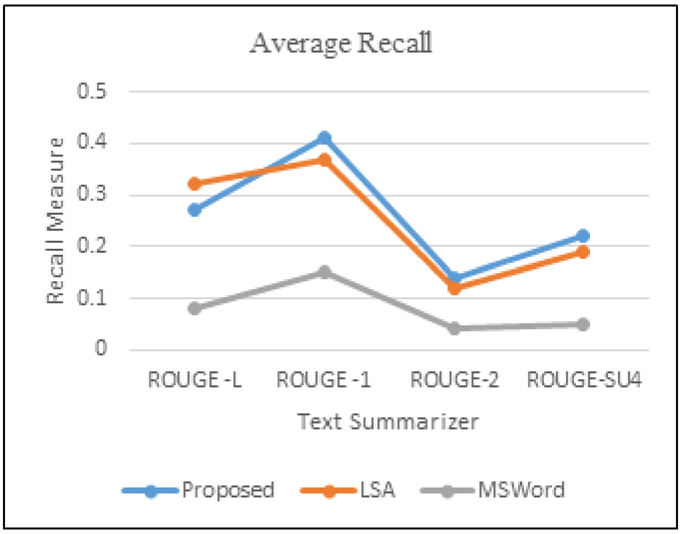


Figure 5.7. Comparative chart for average recall score obtained by different summarization tools

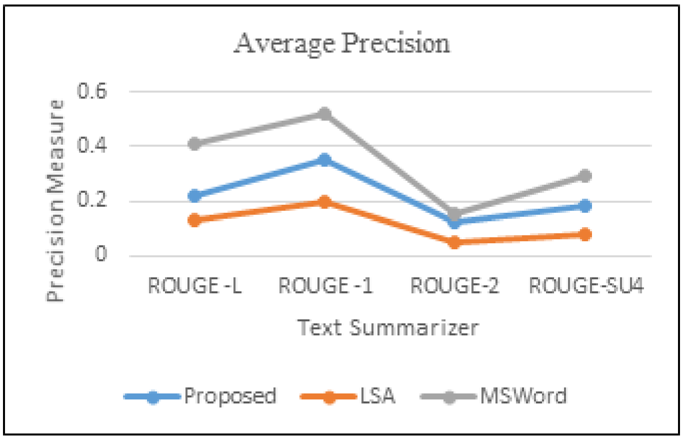


Figure 5.8. Comparative chart for average precision score obtained by different summarization tools

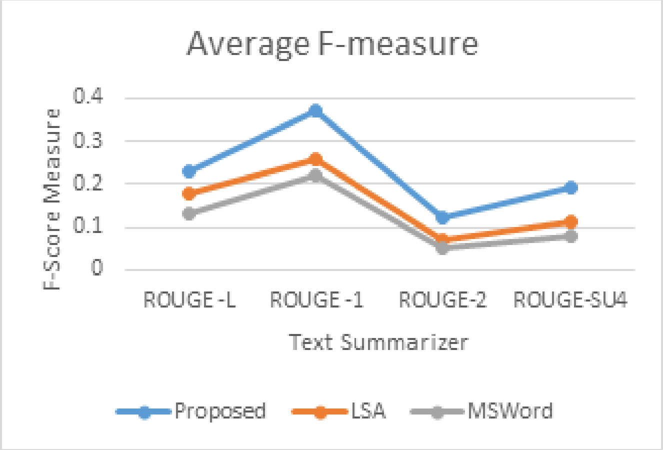


Figure 5.9. Comparative chart for average f-measure score obtained by different summarization tools

Table 5.1. Comparison of different summarization tool: F-Measure

AVERAGE F-MEASURE

	Proposed	LSA	MSWord
ROUGE -L	0.23	0.18	0.13
ROUGE -1	0.37	0.26	0.22
ROUGE-2	0.12	0.07	0.05
ROUGE-SU4	0.19	0.11	0.08

According result of recall obtained for ROUGE -1 & 2, ROUGE-L and ROUGE-SU4 shown in Figure 5.8. Our proposed method shows excellent performance but LSA summarizer shows better result compare to our proposed method for ROUGE-1. For ROUGE-2 and ROUGE-SU4, shows highest average recall for proposed method.

As show in figure 5.8, the difference between precision values between proposed method and MS Word summarizer marginally increased for ROUGE-L and ROUGE-1. Whereas with ROUGE-2 in proposed method, precision values reduced compare to ROUGE-L and ROUGE-1. In ROUGE-SU4, precision value increase in compare to our proposed method. In all cases, LSA shows worst performance in compare to MS Word summarizer and proposed method. The variation occurs between few parameters because of length of summary which is not equal to reference summary depending upon the length of summary which is included in the final summary. Also human summary which is provided in corpus and generated from MS Word summarizer, which is not purely extracted because MS Word summarizer merge sentences from corpus and add into final summary. Whereas proposed system, pick original sentences from document and include into summary.

It is clearly seen after analyze figure-5.9 that for average f-measure, our proposed method performs excellent in compare to LSA summarizer and MS Word summarizer for all parameters i.e. ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4. Our proposed methods show great result for f-measure for all evaluation parameters.

6 References

1. R. Mihalcea and P. Tarau, 'Textrank: Bringing order into texts, " in Proceedings of EMNLP, vol. 4, Barcelona, Spain, 2004.
2. J. Zhang, L. Sun, and Q. Zhou, "A cue-based hub-authority approach for multi-document text summarization, " in Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on, pp. 642-645, IEEE, 2005.K. Elissa
3. S. Hariharan and R. Srinivasan, "Studies on graph based approaches for single and multi-document summarizations, " Int. 1. Comput. Theory Eng, vol. 1, pp. 1793-8201, 2009
4. K. S. Thakkar, R. V. Dharaskar, and M. Chandak, "Graph-based algorithms for text summarization, " in Emerging Trends in Engineering and Technology (lCETET), 2010 3rd International Conference on, pp.516- 519, IEEE, 2010.
5. S. S. Ge, Z. Zhang, and H. He, "Weighted graph model based sentence clustering and ranking for document summarization, " in Interaction Sciences (ICIS), 2011 4th International Conference on, pp. 90-95, IEEE, 2011
6. T.-A. Nguyen-Hoang, K. Nguyen, and Q.-V. Tran, "Tsgvi: a graphbased summarization system for vietnamese documents,"Journal of Ambient Intelligence and Humanized Computing, vol. 3, no. 4, pp. 305- 313, 2012.
7. J. D. Schlesinger, D. P. Oleary, and J. M. Conroy, "Arabic/English multi-document summarization with CLASSY the past and the future, " in Computational Linguistics and Intelligent Text Processing, pp. 568-581, Springer, 2008.
8. X.-c. Ma, G.-B. Yu, and L. Ma, "Multi-document summarization using clustering algorithm, " in Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on, pp. 1-4, IEEE, 2009.
9. V. K. Gupta and T. J. Siddiqui, "Multi-document summarization using sentence clustering, " in Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on, pp. 1-5, IEEE, 2012
10. G. Salton, “Automatic Text Processing: the transformation, analysis, and retrieval of information by computer,” AddisonWesley Publishing Company, USA, 1989.
11. S. Xiong and Y. Luo, "A New Approach for Multi-document Summarization Based on Latent Semantic Analysis," Computational Intelligence and Design (ISCID), 2014 Seventh International Symposium on, Hangzhou, 2014, pp. 177-180.
12. J. Steinberger and K. Jezek, “Using latent semantic analysis in text summarization and summary evaluation,” in Proc. ISIM ’04, 2004, pp. 93–100.
13. Kupiec, Julian, Jan Pedersen, and Francine Chen. "A trainable document summarizer." Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1995.
14. H. P. Edmundson. 1969. New Methods in Automatic Extracting. J. ACM 16, 2 (April 1969), 264-285. DOI=http://dx.doi.org/10.1145/321510.321519
15. Text summarization using Latent Semantic Analysis, Journal of Information Science - Makbule Gulcin Ozsoy, Ferda Nur Alpaslan, Ilyas Cicekli, 2011. [online] Available at: http://journals.sagepub.com/doi/10.1177/0165551511408848
16. T. Kohonen, “Self-organized formation of topologically correct feature maps,” Biological Cybernetics, vol. 43, no. 1, pp. 59–69, 1982
17. T. Kohonen, “Essentials of the self-organizing map,” Neural Networks, vol. 37, pp. 52–65, 2013
18. J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, SOM toolbox for Matlab 5, 2005, http://www.cis.hut.f/somtoolbox/
19. T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, “Websom— self-organizing maps of document collections,” in Proceedings of the Work shop on Self-Organizing Maps (WSOM ’97), pp. 310– 315, Espoo, Finland, June 1997.
20. S. Kaski, T. Honkela, K. Lagus, and T. Kohonen, “Websom— self-organizing maps of document collections,” Neurocomputing, vol. 21, no. 1–3, pp. 101–117, 1998.
21. A. Ultsch and H. P. Siemon, “Kohonen’s self organizing feature maps for exploratory data analysis,” in Proceedings of the Proceedings of International Neural Networks Conference (INNC ’90), pp. 305–308, Kluwer, Paris, France, 1990.
22. J. Vesanto, “SOM-based data visualization methods,” Intelligent Data Analysis, vol. 3, no. 2, pp. 111–126, 1999
23. M.R. Hestenes and E. Stiefel, “Methods of conjugate gradients for solving linear systems”, Journal of Research of the National Bureau of Standards, vol. 49, pp. 409-436, 1952

24. Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 340-348.

7 Publication

1. Shah, C. and Jivani, A., 2016, August. “Literature Study on Multi-document Text Summarization Techniques”. As a book chapter in Communications in Computer and Information Science (CCIS), Springer Series. (ISI, DBLP, EICompendex, SCOPUS)
2. Shah, C. and Jivani, A., 2017. “Multi-document summarization: study on existing techniques”. In International conference on Advances in Computing, Communication and Informatics (ICACCI’17)
3. Shah, C. and Jivani, A., 2017. “An Automatic Text Summarization on Naive Bayes Classifier Using Latent Semantic Analysis”. Published in Springer Contributed Volume series.
4. Shah, C. and Jivani, A., 2017. “A Hybrid Approach of Text Summarization Using Latent -Semantic Analysis and Deep Learning”. Selected for oral presentation in ICACCI’18 sponsored by IEEE, index by Scopus, DBLP and Google Scholar and EI Compendex and Web of Science (THOMPSON REUTERS Conference Proceeding Citation Index)