A Novel Features Based Automated Gurmukhi Text Summarization System

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Computer Science & Engineering

Submitted By

Gurmeet Singh

(Roll No. 801232008)

Under the supervision of:

Mr. Karun Verma

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

May 2014

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "A Novel Features Based Automated Gurmukhi Text Summarization System", in partial fulfilment of the requirements for the award of degree of Master of Engineering in *Computer Science Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mr. Karun Verma* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(GURMEET SINGH)

801232008

ME (CSE)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Mr. Karun Verma)

Assistant Professor

CSED

Thapar University

Patiala

Countersigned by:

(Dr. Deepak Garg)

Associate Professor and Head

Computer Science and Engineering Department

Thapar University

Patiala

(Dr. S. K. Mohapatra)

Dean (Academic Affairs)

Thapar University

Patiala

Page | i

Acknowledgement

I would like to express my deep sense of gratitude to my supervisor, Mr. Karun Verma, Assistant Professor, Computer Science and Engineering Department, Thapar University, Patiala, for his invaluable help and guidance during the course of thesis. I am highly indebted to him for constantly encouraging me by giving his critics on my work. I am grateful to him for giving me the support and confidence that helped me a lot in carrying out the research work in the present form. And for me, it's an honor to work under him.

I also take the opportunity to thank Dr. Deepak Garg, Associate Professor and Head, Computer Science and Engineering Department, Thapar University, Patiala, for providing us with the adequate infrastructure in carrying the research work.

I would also like to thank my parents and friends for their inspiration and ever encouraging moral support, which went a long way in successful completion of my thesis.

Above all, I would like to thank the almighty God for His blessings and for driving me with faith, hope and courage in the thinnest of the times.

Gurmeet Singh

(801232008)

Abstract

Today it is very difficult, laborious and time consuming task to extract information manually from large amount of data available. A need of Automatic Text Summarization Tool is growing which will create complete and understandable summarized text of input text document. Many summarization techniques have been developed for English language but very less research work is carried out in the area of Punjabi Text Summarization. Mostly Gurmukhi script is used for writing Punjabi language. In this paper, the idea to summarize Gurmukhi text document using extraction technique is discussed which summarized text based on features extracted from document. New features namely Presence of URL's or Email Addresses, Presence of Brackets and Presence of Inverted Commas in sentences have been analyzed, proposed and compared with some older features available for text summarization in literature. The document summary generated by proposed system has been compared with summary of document generated by human experts. The comparison shows better results in the terms of Precision, Recall and F-score.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Today, Internet is used for almost everything, it is difficult or impossible to imagine life without internet. There is a large amount of data available over the internet in the form of multimedia. One such multimedia type is text. It is very difficult and time consuming task to extract information manually from large amount of data available. Therefore a need of Automated Text Summarization (ATS) is growing. ATS automatically convert big text file into summarized (abstract) form, which is easily readable, understandable and complete. Without ATS, it is very laborious job for human to read out whole Document to understand it. As an example, suppose president of country is on visit to some other country and he may want a very short summary of every email message on his mobile phone, so by reading that summary he may take decision that he has to pay attention to that mail instantly or not **[9]** and some other areas where ATS can be applied are Media (news headlines), Medical (medical chart of patient without reading his medical history), Education (review or headings), Research (abstracts of research papers), Entertainment (preview of movies), Sports (scoreboard showing statistic of game), Curriculum Vitae (CV) and Subject of an email or letter etc **[9][10].**

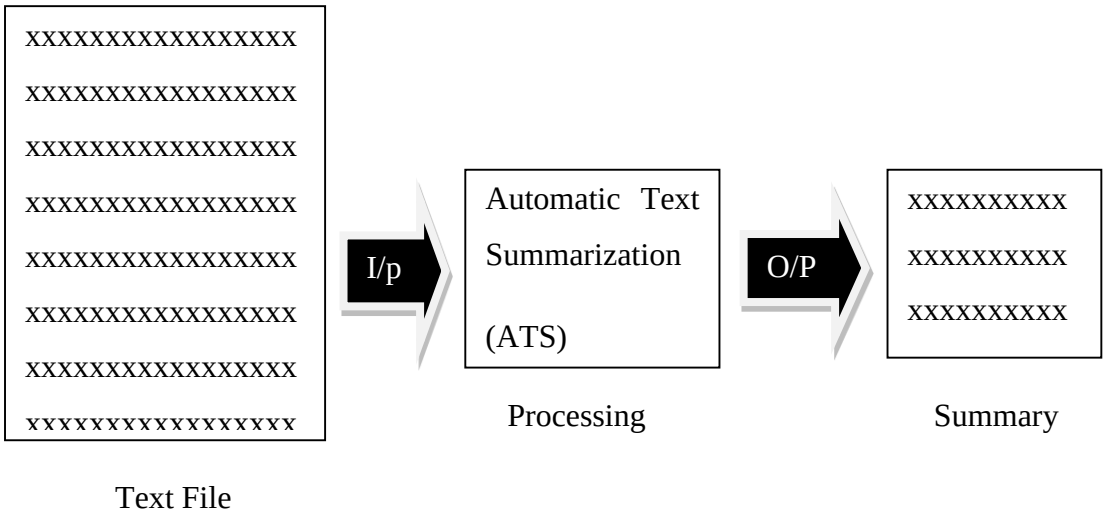


Figure 1: Block Diagram of Automatic Text Summarization

Automatic Text Summarization Technique is divided into two categories as shown in Figure 2:-

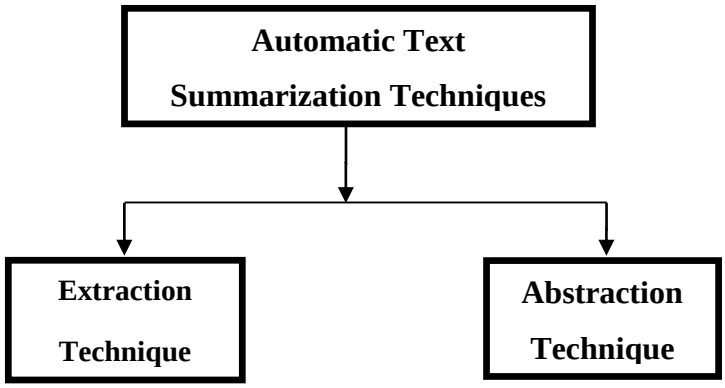


Figure 2: Block Diagram of Automatic Text Summarization Techniques

1) Extraction technique 2) Abstraction technique. Extraction technique is simply extracting important sentences into final summary, where importance of sentence is calculated based on weights assigned to sentences using statistical and linguistic features of text**[6][9]**. In contrast, Abstraction technique involves semantic based text summarization. In which input text file is readout and understand by the ATS and then rewrite it using natural language processing (NLP) by condensing its important information and overall meaning. Abstraction technique uses purely linguistic feature, which are very hard to calculate **[10].** Summary created by human is different from summary created by the ATS. As human brain can easily capture and understand the given task deeply on the basis of semantics, it is a big challenge for ATS to understood and generate summary as human do. Although ATS has advantages over manual summarization such as: - length of the summary needed is adjustable and it is very fastest process.

Chapter 2: Literature Work Overview

In this chapter we explain various researches and techniques used for automated text summarization.

H.p.luhn **[8]** is the father of automatic text summarization. He started creating abstract of technical literature by automatic means for the sake of quick and accurate identification of technical papers. The main goal of luhn was to save readers effort and time in finding relevant information from articles and reports. The complete text of a document was prepared in machine readable form and was scanned by IBM 704 machine and after processing statistical information is derived. Word frequency and distribution was used for computing relative measures, first for individual word and secondly for sentences.

P.B. Baxendale **[21]** in 1958 proposed new feature that is sentence position or sentence location in an input document. Baxendale analyzed that sentences which are located at the beginning or at the end of the document are important than other sentences contained into the document. He tested sentence location feature on 200 paragraphs and found that 85% of the paragraphs were from beginning and 7% of the paragraphs were from ending locations. A sentence location feature became an important feature for sentence extraction and it is used till now.

H.P. Edmundson **[4]** in 1969 described new typical structure for a text summarization. He proposed two new features and incorporates two old features explained above. Two new features proposed by Edmundson are:

1. **Cue Words**: presence of most indicative words into a document such as lastly, finally, in summary etc.
2. **Title or Heading Words**: An extra weight was assigned to a sentence, if sentences have heading words in it.

Edmundson evaluate his proposed structure with manual abstracts and found 44% of accuracy.

Paul C. Jacobs **[22]** in 1999 built a System for Conceptual Information Summarization, Organization and Retrieval (SCISOR) was a prototype system that performs two types of functions such as question answering and text analysis. SCISOR runs on real time data and runs on Sun™ workstation under UNIX™. SCISOR was able to process six stories every minute. Financial news was given as input to SCISOR. Topic-analyser selects and analyses stories about acquisitions and corporate mergers and performs lexical analysis of various inputs such as names, numbers, dates and on other information. A combination of language-driven and conceptual analysis process on the texts and was able to identify various major roles like price, target, suitor, company product, financing and other features. The result of language-driven and conceptual analysis was represented uniquely for every story and added to the central knowledge database. The conceptual component use information stored at central knowledge base and able to give the answers by analyzing English question given to SCISOR.

Julian Kupiec **[23]** in 1995 described a new technique of summarization using naive-Bayes classifier, the classification function classifies each sentence as it is extraction worthy or not. He described some new features like sentence length, presence of uppercase words, phrase structure and incorporate feature developed by Edmundson. Documents were inputted in the picture formats which were scanned by OCR to extract text data from it and then inputted to classifier. Every sentence in a document was given a score using naive-Bayes classifier and sentences having top score were extracted for the summary. To evaluate the system, a corpus of manually created abstracted were used in such a way that every sentence in manually created abstract, where matched with the actual document and mapping value was created. The mapping values then used to evaluate auto-abstract created by system.

Eduard Hovy and Chin-Yew Lin **[24]** in 1999 create a system named Summarist and was used to create summaries for multi-languages. Summarist was able to combine symbolic world knowledge and solid processing phase to decide relevance of the concept. Summarist was structured into three phases:

Summarist = identification of topic + interpretation + generation of summary

For identification, filtering of information was done in order to retain only most important topics. For interpretation, fusing of extracted topic was done into more briefly and was clearly expressed. In generation phase reformulation of the extracted material was done to form new text.

Chin-Yew Lin **[25]** in 1999 try to model system for sentence extraction using decision trees for generating informative generic/query-oriented extracts. Lin identify various features such as: baseline, title, tf-idf, position, query signature, IR signature, sentence length, lexical connectivity, numerical data, proper noun, pronoun and adjective, weekdays and month, quotation and first sentences. The score for all features were combined by the automated learning using decision trees and simple combination function. Lin conducts an in-depth study of various features effect through glass-box. The result of experiment shows that there was no single feature performs best overall for query-based summaries. Features like numerical data gives good results for query requires numerical value answer, while weekdays and month feature gives best result for (when?) queries.

Jade Goldstein **[26]** in 2000 developed a multi-document summarization system by extraction technique. The system was capable of summarizing complete set of documents that contains relevant information shared among all documents only once. He suggested that same techniques available for single document summarization can also be applied for multi document summarization besides only four key differences:

1. Degree of redundancy in multi-document summarization is very higher than the degree of redundancy in single document.
2. Compression ratio is very smaller in the case of multi-document summarization.
3. The problem of co-reference even big challenges for multi-document summarization.
4. A set of document may contain temporal dimension about an unfolding event. Here further information may override previous one.

An ideal multi-document summarization system would able to show different levels of details as output, that was very difficult task without NLP understanding and

generation. Goldstein suggested some methods for multi-document summarization such as summary from common parts of documents, summary from common and unique parts of documents, centroid document summary, latest document plus outlier's summary. The system developed was emphasizing on "relevant novelty", which was a metric used for minimizing redundancy and maximizing relevancy and diversity. If we want to measure relevant novelty then firstly we have to measure relevance and novelty independently using metric "marginal relevance". The paragraph had high marginal relevance if it was having relevancy to query and usefulness for summary hence Goldstein label his method as "maximum marginal relevance multi-document" (MMR-MD). MMR-MD was defined using Sim_1 and Sim_2.

Sim_1 was computed by firstly computing cosine similarity, coverage score and information contents of a paragraph, finally temporal sequence of the document allowing recent information having more weights.

Sim_2 was computed by firstly computing cosine similarity, penalizes passage which were subset of clusters from which other passages was already chosen and penalizes documents from where selected passage was taken.

Hongyan jing et al **[27]** in 2000 proposed a novel system for text summarization. In which reduction of extra phrases was done from the extracted sentences for summary. The system used multiple sources for deciding which phrases may be removed without losing coherence of resultant summary. The reduction algorithm had five steps:

1. **Syntactic Paring**: Firstly input sentences were parsed using ESG parser.
2. **Grammar Checking**: Here every component of sentence was checked grammatically so that which component of sentence would be deleted without losing sentence grammar.
3. **Context Information**: Here decision about the sentence component was done, by deciding which component of sentence was mostly related to gist of the topic.

4. **Corpus Evidence**: Corpus was used which had a collection of sentences which were reduced by human and their corresponding original sentences to done computation by system that how likely human remove extra phrases by maintaining coherence.
5. **Final decision**: The final decision was taken based upon the result of previous steps. To decide which phrase was to be removed if and only if it was not grammatically wrong.

The output summary was evaluated against summary created by humans. The success rate achieved was 81.3% that was very good.

Conroy and O'leary **[28]** made a system for sentence extraction using two techniques named QR and HMM. In QR method of summarization was simple. In QR method the importance of sentence was measured and the sentence having higher importance was added to the summary. After that, relative score of remaining sentences were changed because some the remaining sentences were redundant. That process was done repeatedly until main idea would not capture. Second method was hidden markov model (HMM) that is sequential model for text summarization. In HMM only three features was used: sentence position, total no of terms in the sentence and likeness of sentence terms given document terms. The HMM was arranged as follows: HMM contained 2s+1 states, rotating between s (summary states) and s+1 (non-summary states). Some rule was applied that was skipping of next state only in summary states otherwise hesitation only in non-summary states. At last evaluation was done by comparing HMM created summary with human created summary.

D.R. Radev et al **[29]** in 2004 developed a system for multi-document summarization named centroid-based summarization (CBS). The first phase was topic detection, here topic related to same event were grouped together. An agglomerative algorithm was used over TF-IDF for accomplishing this task of topic detection. In second phase centroids values were used to identify sentences in every cluster that were central to the topic of whole cluster. Here two metric were defined by author that were used to resemble the two in the maximal marginal relevance (MMR). The first metric was cluster based relative utility (CBRU), which evaluate the relevance of a particular sentence to the general topic of whole cluster. The second metric was cross-sentence

informational subsumption (CSIS), which compute redundancy factor among sentences. Three features were used for calculating score of every sentence S_i.

1. **Centroid value**: The total of centroid values of all the words in a sentence is called centroid value.
2. **Sentence Position**: Starting and ending sentences were more important.
3. **First-sentence overlap**: The inner product between word occurrence array of S_i and the first sentence of a document.

The final score of every sentence was calculated using three score computed above minus redundancy penalty (R_s) for every sentence that overlapped higher score sentences.

S. P. Yong et al **[30]** in 2005 described an automated system that had learning ability by combining various recent approaches such as: statistical approach, neural network and sentences extraction. The system had extracted 83.03% of significant contents from the input document. The system was built of three phases:

1. **Pre-Processing Phase**: the system had two pre-processing methods that where applied, first was the removal of unnecessary words (stop words) and the second was stemming.
2. **Keywords Extraction phase**: Keywords were extracted using TF (term frequency) by IDF (inverse document frequency) and top rated list was taken as keywords.
3. **Summary Creation Phase**: The sentence was extracted for the summary which had more no of keywords present in it. It was suggested that again run through stop words because to ensure that no stop word was working as keyword.

Krysta M. Svore et al **[31]** in 2007 proposed an algorithm that was based on neural network and use they used third party datasets for doing text summarization. The dataset contained 1365 document collected from CNN.com, each document had title, timestamp, three- four human created story highlights and article text. Three highlights were created by the system and its evaluation was done using two metrics. The first metric, concatenate system generated highlights and human generated

highlights differently and does comparison among them. Second metric compared the sentences at individual level in an ordered way. The training of the corpus was done from labels and features for every sentence of a document. The ranking algorithm used was RankNet, a pair-base neural network algorithm that uses gradient decent procedure for training. ROUGE-1 was used for training set, that assigning score to similar value between human created highlights and sentences in a document. The proposed system had a feature which drives information from the queries logs of the Microsoft's search engine and Wikipedia entries. The author suggested that if the sentence contains keywords that were used in the search engine and Wikipedia were most important and had greater chances of being a sentence as highlight. The summaries were evaluated using ROUGE-1 and ROUGE-2 and showed significant improvements over the baseline.

Ladda Suanmali et al **[32]** in 2009 propose a system based on fuzzy logic. Fuzzy rules and fuzzy sets were used for extracting important sentences based on sentence features. Nine features were used to calculate sentence importance and a value was given between '0' and '1' if a particular feature was present in the sentence. Feature used were sentence centrality, title, word sentence, keyword, sentence length, similarity to first sentence, sentence position, numeric data and proper noun. The values of these features were given as input to fuzzy system, which produced an output based on features and IF-THEN rules defined for sentence extraction. The system was evaluated, by comparing results among Microsoft Word 2007 summarizer and baseline summarizer.

Li Chengcheng **[33]** in 2010 presented a novel automatic text summarization method based upon Rhetorical Structure Theory (RST). RST is the notations of rhetorical relation existing between two non-overlapping text called nucleus (N) and satellite (S). The empirical observations were used to express distinction between N and S, which was expressed by N that what is more important to the writer's purpose than S and the Rhetorical nucleus was comprehensible independent of the S. The full text was divided into small units called sentences based on delimiter (commas, full stop, any punctuation mark found between sentences). The whole procedure was built on a graph, sentences which were less important were deleted from the graph and

remaining sentences were summarized. The main procedure followed by author was divided into three steps:

1. Candiate sentence was analyzed
2. Rhetoric relations were founded
3. Essential parts of the sentences were decided for abstract creation.

Ladda Suanmali et al **[34]** in 2011 developed a system for sentence extraction based on fuzzy, genetics algorithm, sementics labeling and their collaboration for generating high quality summary. In pre procesing some features were used for calculating sentence score. After calculating sentence score, five different methods were used to extract important sentences in processing phase.

1. **General Statistical Method (GSM):** Here sentence score is calculated by adding weight of all features for sentence S_i.
2. **Fuzzy Logic Method**: The features ectracted were given as input to fuzzifier and some rules were defined here. At last defuzzification was done and final score was given.
3. **Genetic Algorithm Method**: The vectors of extracted features were given as input. Then feature scores where optimized using GA.
4. **Semantic role labeling Method**: sentence similarity value was calculated using sementic role labeling that capure semantic words in sentences.
5. The system was evaluated by taking six summarizer as benchmarks and proposed system outperforms when compared with others summarizers.

Vishal gupta **[6]** in 2012 prposed a system for punjabi text summarization using extraction technique. The proposed system had two main phases: PreProcessing Phase and Processing Phase. Preprocessing phase involves refinment of input document which involves sentence boundry identification, stopwords removal, stemmer and etc. In Processing phase , sentence weight was calculated using various features identified by the author and at last final score of every sentence was calculated, top ranked sentence where extracted as summary.

2.1 Automatic Text Summarization Using Extraction Technique

As we discussed above, an extraction technique of text summarization consists of selecting important sentences from source document and arrange them in the destination document. Our main focus is on extraction technique for text document summarization, overview of Extraction technique is shown in Figure 3:

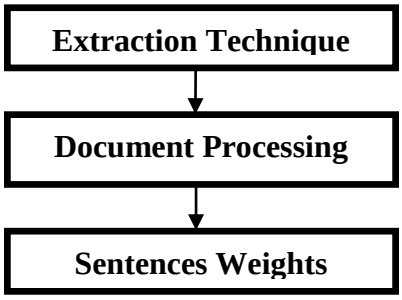


Figure 3: Overview of Extraction Technique

Usually, the information in a given document is not constant, which means that some parts of document are more important than others are less important. The main challenge is to identify important parts of document and extract them for final summary. Here most work presented on single-document summarization using extraction method. In this section, some extractive techniques are discussed briefly, which are applied for extraction of sentences for final summary.

Automatic text summarization using extraction technique is divided into three steps as shown in Figure 4:

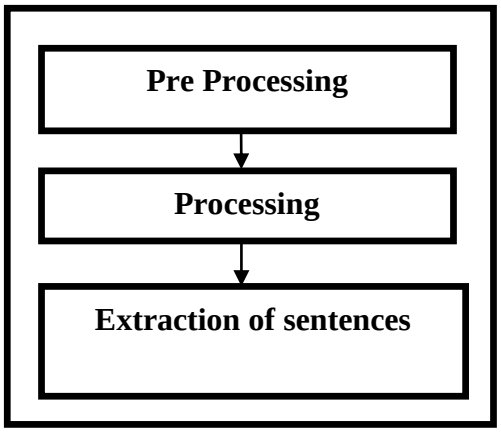


Figure 4: Block Diagram of Extraction Technique

2.1.1 Pre Processing

In pre processing phase of ATS, we break the text document into sentences, sentences are further broken into words and after that stop words are removed. Preprocessing phase involves four steps 1) Segmentation 2) Tokenization 3) Stop words removal 4) Root Word Identification as shown in Figure 5:

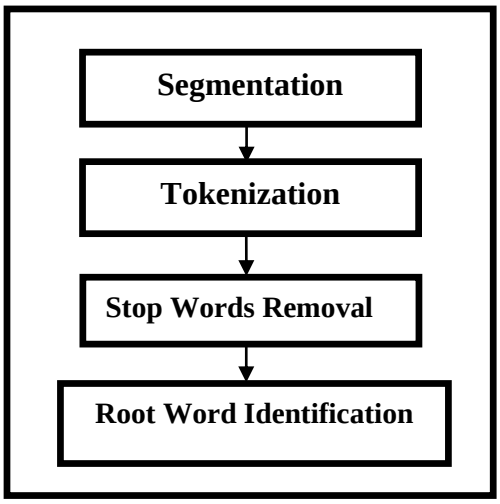


Figure 5: Block Diagram of Pre Processing Phase

2.1.1.1 Segmentation

In segmentation phase, sentences are segmented based upon sentence boundary. In Punjabi language sentence boundary is identified by "।" called "dandi". On every sentence boundary, the sentence are broken and put into list of lists (data structure available in python). Sentence boundary in Punjabi text is marked by circle as shown in Figure 6. The output of sentence segmentation phase is collection of sentences that are further processed in next phases.

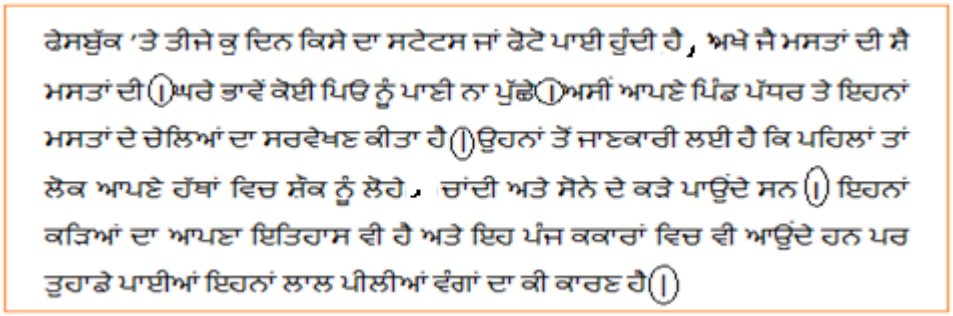

ਫੇਸਬੁੱਕ 'ਤੇ ਤੀਜੇ ਕੁ ਦਿਨ ਕਿਸੇ ਦਾ ਸਟੇਟਸ ਜਾਂ ਫੋਟੋ ਪਾਈ ਹੁੰਦੀ ਹੈ, ਅਖੇ ਜੈ ਮਸਤਾਂ ਦੀ ਸ਼ੈ ਮਸਤਾਂ ਦੀ ।ਘਰੇ ਭਾਵੇਂ ਕੋਈ ਪਿਓ ਨੂੰ ਪਾਣੀ ਨਾ ਪੁੱਛੇ।ਅਸੀਂ ਆਪਣੇ ਪਿੰਡ ਪੱਧਰ ਤੇ ਇਹਨਾਂ ਮਸਤਾਂ ਦੇ ਚੇਲਿਆਂ ਦਾ ਸਰਵੇਖਣ ਕੀਤਾ ਹੈ।ਉਹਨਾਂ ਤੋਂ ਜਾਣਕਾਰੀ ਲਈ ਹੈ ਕਿ ਪਹਿਲਾਂ ਤਾਂ ਲੋਕ ਆਪਣੇ ਹੱਥਾਂ ਵਿਚ ਸ਼ੌਕ ਨੂੰ ਲੋਹੇ, ਚਾਂਦੀ ਅਤੇ ਸੋਨੇ ਦੇ ਕੜੇ ਪਾਉਂਦੇ ਸਨ। ਇਹਨਾਂ ਕੜਿਆਂ ਦਾ ਆਪਣਾ ਇਤਿਹਾਸ ਵੀ ਹੈ ਅਤੇ ਇਹ ਪੰਜ ਕਕਾਰਾਂ ਵਿਚ ਵੀ ਆਉਂਦੇ ਹਨ ਪਰ ਤੁਹਾਡੇ ਪਾਈਆਂ ਇਹਨਾਂ ਲਾਲ ਪੀਲੀਆਂ ਵੰਗਾਂ ਦਾ ਕੀ ਕਾਰਣ ਹੈ।

Figure 6: Segmentation Phase

2.1.1.2 Tokenization

Tokenization is the process of braking down the sentences into words. In Gurmukhi, sentences are tokenized by identifying the space and comma between the words. So the list of lists is created in which each list contains elements as words or also called tokens which are maintained for further processing.

2.1.1.3 Stop Words Removal

Most commonly or frequently used words are called stop words. Stop words are meaningless and does not have any importance into the sentences. So these types of words should be removed from input document, otherwise the sentence containing more no of stop words could have higher weight. We analyzed that every Gurmukhi text document must contains minimum 30 percent or more stop words. A list of some Gurmukhi stop words is shown in Table 1.

Table 1: Some Gurmukhi Stop Words

ਹੁਣ	ਨੂੰ	ਕੀਤਾ	ਸੀ
ਕਿ	ਤੇ	ਹਨ	ਕੀ
ਅਤੇ	ਜੋ	ਦੇ	ਦੀ
ਜੋ	ਹੇ	ਹਾਂ	ਪਰ
ਹੁੰਦੀ	ਰਹਿ	ਮੇਰੀ	ਲਈ

2.1.1.4 Root Word Identification

Root word identification is the process of identifying and converging words towards their root (stem). In most of the cases, variants of words having similar meaning when we interpret them. In Gurmukhi, root word for "ਮੁੰਡੇਆਂ", "ਮੁੰਡੇ" is "ਮੁੰਡਾ". An improved root word identification algorithm for Gurmukhi is developed and implemented in proposed system by taking [5] as reference algorithm.

2.1.2 Processing

Processing phase is the heart of GTSS, here detailed analysis on text document is done. In processing phase, feature value for every sentence is calculated. Some features in Punjabi language are different from other languages. In GTSS some old features and some new features are used for calculating sentence score are shown below:-

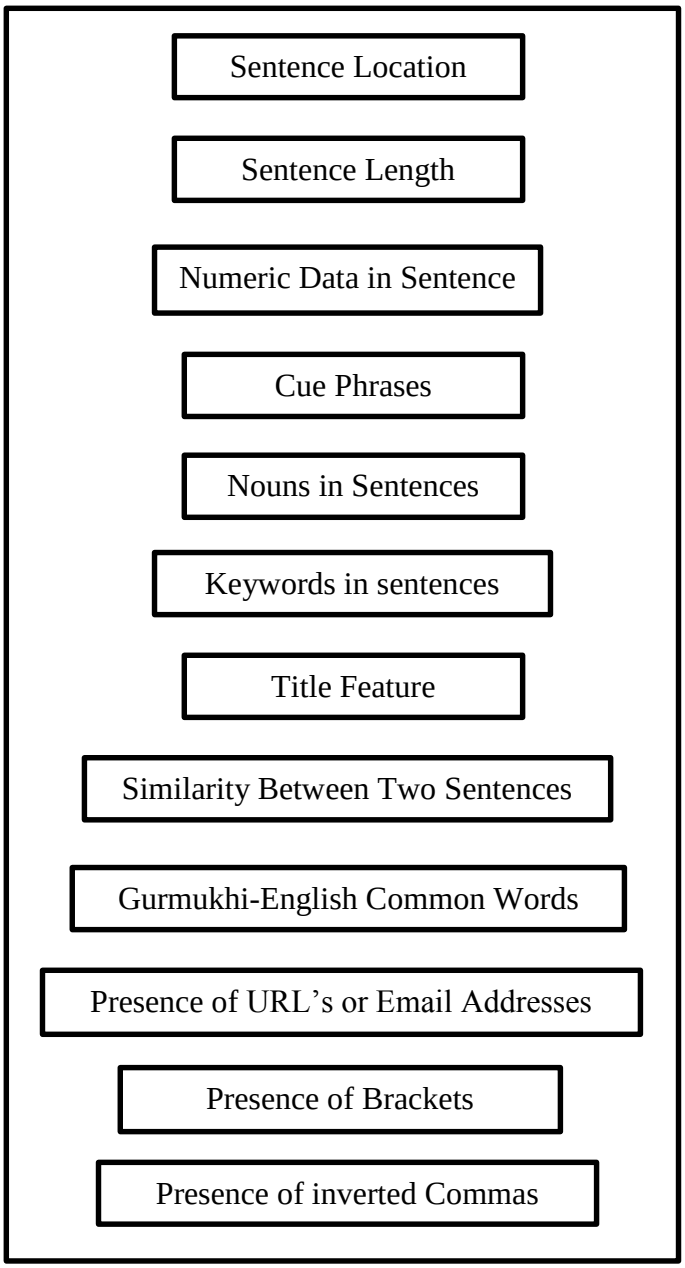


Figure 7: Block Diagram of Features for Sentences Extraction

2.1.2.1 Sentence Location (f_1)

Location of sentence tells its importance in a text document. Starting sentences are important in almost all the cases because they express theme of the document and has higher probability to be extracted for the summary. Sentence location value is calculated in such a way that, higher values are assigned to the starting sentences and lower values are assigned to ending sentences. Sentence location values are calculated using equation 1.

$$\boldsymbol{Sloc(S_i) = \frac{n - i}{n}} \quad (\mathbf{1})$$

Where S_i is the i^{th} sentence in a text document, n= Total no of sentences in a text document and i ranges from 0 to n.

2.1.2.2 Sentence Length (f_2)

Sentences which are shorter in length may not represent theme of a text document because of fewer words contained in it, although selecting longer length sentences are also not good for summary. So sentence length values are calculated in such a way that, shorter and longer sentences are assigned lower values. Sentence length values are calculated using equation 2**.**

$$\boldsymbol{Slen(S_i) = \frac{W(S_i)}{LongSen}} \quad (\mathbf{2})$$

Where W(S_i)= Total no of words in sentence S_i and LongSen= Total no of words in a longest sentence.

2.1.2.3 Numeric Data in Sentence (f_3)

A numeric data in Gurmukhi Sentence represent some important information regarding some date, age, rupees, address etc. In Gurmukhi numeric data can be represented in Roman, Digits and Gurmukhi fonts such as (XIV/I/MMXIV // 16/01/2014 // ੧੬/੦੧/੨੦੧੪) date, (XXIII// 23 //੨੩) age, (MMMMM //5000 // ੫੦੦੦) rupees etc. Numeric data values are calculated using equation 3.

$$\boldsymbol{Ndata(S_i) = \frac{nd(S_i)}{SenLen(S_i)}} \quad (\mathbf{3})$$

Where nd(S_i)= Total no of numeric data in sentence S_i and SenLen(S_i)= Total no of words in sentence(S_i).

2.1.2.4 Cue phrases (f_4)

Cue phrases are the most indicative words such as lastly, finally, in summary, on the other hand, for example and anyway etc. Sentences containing cue words are important and have higher probability to be extracted for the summary. In Gurmukhi cue phrases are ਨਤੀਜਾ, ਨਤੀਜੇ, ਨਿਚੋੜ, ਅੰਤ ਵਿੱਚ, ਕੁਜ ਸ਼ਬਦਾ ਵਿੱਚ, ਗੱਲ ਨਬੇੜਦੇ ਹੋਏ and ਸਿੱਟਾ etc.

2.1.2.5 Nouns in Sentence (f_5)

A noun is the name of the person, place or thing. Sentences containing nouns are important and have higher probability to be extracted for the summary. In Gurmukhi nouns are ਮਨਵੀਰ, ਸੱਚੇਨਦਰਾ, ਯੋਗੀ, ਸਨਦੀਪ, ਮਯੂਰ, ਸ਼ਿਵਾਂਗੀ, ਚਾਰੂ, ਥਾਪਰ and ਪਟਿਆਲਾ etc.

Nouns are calculated using equation 4.

$$\boldsymbol{Nouns(S_i) = \frac{Ns(S_i)}{SenLen(S_i)}} \quad (\mathbf{4})$$

Where Ns(S_i) = Total no of nouns in sentence S_i.

2.1.2.6 Keywords in Sentence (f_6)

Keywords are words that appear with unusual frequency (very high) in a text document. Keywords identification and calculation is very important feature and it helps in deciding sentence importance. In proposed system top 10% words having higher frequencies are taken as keywords. Keywords in sentences are calculated using equation 5.

$$\boldsymbol{Key(S_i) = \frac{Keywords(S_i)}{SenLen(S_i)}} \quad (\mathbf{5})$$

Where key(S_i) = Total no of keywords in sentence S_i.

2.1.2.7 Title Feature (f_7)

Words present in Title are the heart of the matter contained in text document. So if a Gurmukhi sentence S_i is highly associated with title then sentence S_i has higher

probability to be extracted for the summary. Title feature is calculated using equation 6.

$$TF(S_i) = \frac{TitleMatch(S_i)}{Wt} \quad (6)$$

Where TitleMatch(S_i) = Total no of (S_i) words matched with title words and Wt = Total no of words in title.

2.1.2.8 Similarity Between Two Sentences (f_8)

Similarity between two sentences is used to determine whether two sentences are semantically equal or not. Root words are used for determining similarity between two Sentences. If two Sentences having maximum root words match then they have higher probability of being similar.

2.1.2.9 Gurmukhi-English Common Words (f_9)

Gurmukhi sentences may contain some Gurmukhi-English common words such as ਸਕੂਲ, ਯੂਨੀਵਰਸਿਟੀ, ਐਡਿਟ, ਸ਼ੇਵ and ਕਰੋੜ etc. Sentences containing Gurmukhi-English common words are important and have higher probability to be extracted for the summary. Gurmukhi-English common words are calculated using equation 7.

$$GE(S_i) = \frac{GurEng(S_i)}{SenLen(S_i)} \quad (7)$$

Where GurEng(S_i)= Total no of Gurmukhi-English common words in sentence S_i.

2.1.2.10 Presence of URL's or Email Addresses (f_{10})

Internet is important and widely used application now days. Text document may have URL's or Email Addresses present in it, which provides more information about the document in process. After doing analysis of various Gurmukhi newspapers and Gurmukhi documents it has been found that this feature has very high importance than other and needs to be extracted for the summary.

2.1.2.11 Presence of Brackets (f_{11})

Sometimes sentences may contain brackets such as ( ) parentheses, {} curly brackets etc. mostly braces contains material which could be omitted without destroying or altering sentence meaning. After doing analysis it has been found that brackets do not contain important information and has lower probability to be included for the summary. Presence of brackets in sentence is calculated using equation 8.

$$\boldsymbol{BK(S_i) = \frac{SenLen(S_i) - BracLen(S_i)}{SenLen(S_i)}} \quad (\mathbf{8})$$

Where BracLen(S_i)= Total no of words within brackets in sentence(S_i).

2.1.2.12 Presence of Inverted Commas (f_{12})

In Gurmukhi (“ ”,‘ ’)quotation marks or inverted comma surrounding quotations, direct speech, literal title or name etc. contains important information. After doing analysis it has been found that an inverted comma has higher probability to be included for the summary. Presence of inverted commas is calculated using equation 9.

$$\boldsymbol{IC(S_i) = \frac{QuoteWords(S_i)}{SenLen(S_i)}} \quad (\mathbf{9})$$

Where QuoteWords(Si)= Total no of (Si) words between quotation marks**.**

2.1.3 Sentence-Extraction Phase

In Sentence-Extraction phase firstly final weight of every sentence is calculated using Weight-Ranking equation given in equation 10. After calculating final weight of every sentence, extraction of sentences is done according to compression ratio required.

$$\boldsymbol{SenWeight(S_i) = f_1(S_i) + f_2(S_i) + \cdots + f_{12}(S_i)} \quad (\mathbf{10})$$

Where SenWeight(S_i) is a final weight of sentence(S_i) and f_1,f_2.... f_{12} are features which are computed above.

In order to make summary more reliable, accurate, complete and less redundant following process is applied:

1. Final weight of every sentence using weight-ranking equation is computed.
2. Final weights are sorted in reverse order.
3. Top weighted sentences are selected for summary according to compression ratio required.
4. Selected sentences for summary are shown in same sequence as they appeared in input text document.

Chapter 3: Motivation

A survey of existing systems discloses the fact that there is no such system for Punjabi Text Summarization which is more accurate and has low redundancy. So it is very difficult for some people to use internet without any language barrier. Punjabi is the 9th most widely spoken language worldwide and most commonly Gurmukhi script is used for writing Punjabi. Today there is lot of data available over the internet which is in Punjabi. So, Gurmukhi Text Summarization System (GTSS) will help those people to read out text documents more precisely without reading whole document and also to summarize large file of Gurmukhi text for various types of data analysis. In this dissertation, we will discuss the idea to summarize Gurmukhi text document using extraction technique.

Chapter 4: Problem Statement

Today there is only one system available for Punjabi Text Summarization over the internet, which is not perfect and has low quality of the summary. So a lot work needs to be done to improve the quality of the summary by developing new features and techniques for sentences extraction. Sometimes it may happen that the theme sentence of a document does not have enough weight to be included in the summary. If this issue is not handled carefully then summary may be ambiguous and it is not useful.

In this dissertation, a system is developed which will summarize Gurmukhi text document using extraction technique and has better accuracy then the existing system in terms of precision, recall and f-score. In proposed system some new features such as Emails and Web-Address, which are very important features of text and can be included in the summary. Other important features are quotations (sayings of peoples) and brackets ( ), these features can be included in the final summary.

So, work is done to resolve these types of issues in GTSS and to improve the quality of the summary that covers maximum theme with less redundancy.

Chapter 5: Proposed System Architecture

Goal of ATS is to select the most important sentences of the text document. The proposed method uses some statistical features to find most important sentences. As shown in Figure 8, system consists of 3 major blocks, Pre Processing phase, Processing phase and Sentence-Extraction phase.

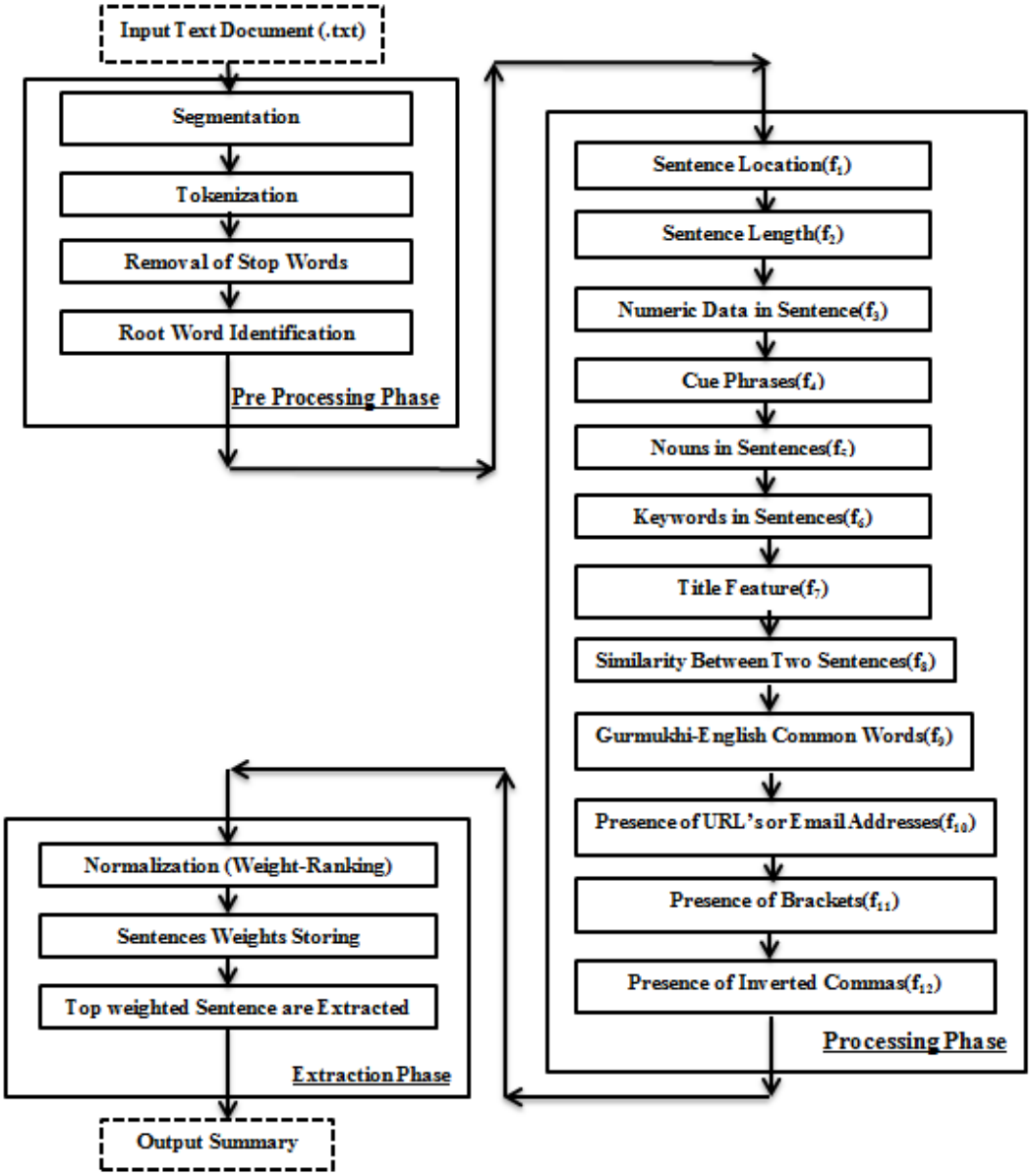


Figure 8: Proposed System Architecture

Chapter 6: Implementation

In this section we discuss how to summarize Punjabi text document and make summary according to need of user say 20%, 40%, 70% etc. Python 3.3.3 is used for implementation and detailed implementations of every step with snapshots are given in this chapter. Here are some brief steps that are involved implementation are shown below: -

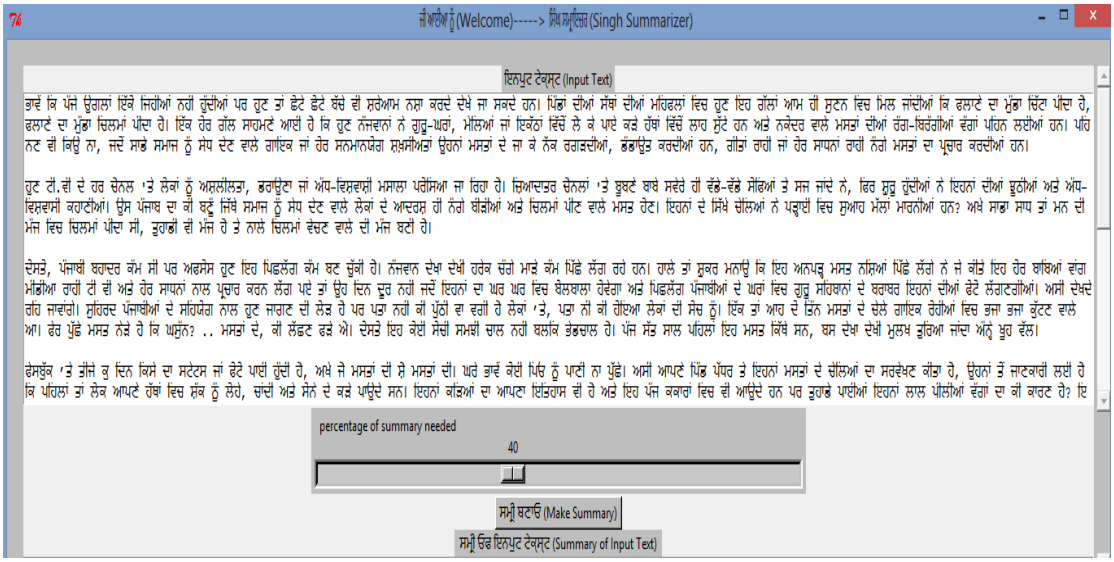
Steps involved in implementation are: -

1. Open input file in reading mode.
2. Store whole file in a list (data structure in python).
3. Close input file.
4. Now break list into list of lists by identifying delimiter such as **|** (dandi).
5. After step 4, break list of lists further into elements of lists by identifying delimiter such as (space between two words).
6. Remove stop words from list of lists.
7. After removing stop words remaining words are stemmed towards their root.
8. Calculate weights for 12 features and create new list for every feature and store value for every feature into new lists.
9. Normalize weights using Weight-Ranking equation.
10. Sort sentence weight in decreasing order.
11. Vary scale according to need of summary in percentage (%).
12. Extract sentence from main list which is created in step 2.
13. Put extracted sentences into new list in same order as they are in main list.
14. Show summary to user.

6.1 Reading, Storing and Segmenting Of Input File

Steps involved in section 6.1 are:-

1. User select input file (.txt)
2. Selected file is opened in read mode and stored in a list (data structure in python) by the GTSS.
3. After storing file contents in a list, segmentation on list is performed.
4. Segmented output is stored in a list of lists as shown in Figure 9.




```
Preprocessing phase


segmentation phase output

[['\ufeffਭਾਵੇਂ ਕਿ ਪੰਜੇ ਉਂਗਲਾਂ ਇੱਕੋ ਜਿਹੀਆਂ ਨਹੀਂ ਹੁੰਦੀਆਂ ਪਰ ਹੁਣ ਤਾਂ ਛੋਟੇ ਛੋਟੇ ਬੱਚੇ ਵੀ ਸ਼ਰੇਆਮ ਨਸ਼ਾ ਕਰਦੇ ਦੇਖੇ ਜਾ ਸਕਦੇ ਹਨ।', ' ਪਿੰਡਾਂ ਦੀਆਂ ਸੱਥਾਂ ਦੀਆਂ ਮਹਿਫਲਾਂ ਵਿਚ ਹੁਣ ਇਹ ਗੱਲਾਂ ਆਮ ਹੀ ਸੁਣਨ ਵਿਚ ਮਿਲ ਜਾਂਦੀਆਂ ਕਿ ਫਲਾਣੇ ਦਾ ਮੁੰਡਾ ਚਿੱਟਾ ਪੀਂਦਾ ਹੈ, ਫਲਾਣੇ ਦਾ ਮੁੰਡਾ ਚਿਲਮਾਂ ਪੀਂਦਾ ਹੈ।', ' ਇੱਕ ਹੋਰ ਗੱਲ ਸਾਹਮਣੇ ਆਈ ਹੈ ਕਿ ਹੁਣ ਨੌਜਵਾਨਾਂ ਨੇ ਗੁਰੂ-ਘਰਾਂ, ਮੱਲਿਆਂ ਜਾਂ ਇਕੱਠਾਂ ਵਿੱਚੋਂ ਲਾਹ ਸੁੱਟੇ ਹਨ ਅਤੇ ਨਕੇਦਾਰ ਵਾਲੇ ਮਸਤਾਂ ਦੀਆਂ ਰੰਗ-ਬਿਰੰਗੀਆਂ ਵੰਗਾਂ ਪਹਿਨ ਲਈਆਂ ਹਨ। ਪਹਿਨਣ ਵੀ ਕਿਉਂ ਨਾ, ਜਦੋਂ ਸਾਡੇ ਸਮਾਜ ਨੂੰ ਸੇਧ ਦੇਣ ਵਾਲੇ ਗਾਇਕ ਜਾਂ ਹੋਰ ਸਨਮਾਨਯੋਗ ਸ਼ਖ਼ਸੀਅਤਾਂ ਉਹਨਾਂ ਮਸਤਾਂ ਦੇ ਜਾ ਕੇ ਨੱਕ ਰਗੜਦੀਆਂ, ਭੱਭਾਉਂਦ ਕਰਦੀਆਂ ਹਨ, ਗੀਤਾਂ ਰਾਹੀਂ ਜਾਂ ਹੋਰ ਸਾਧਨਾਂ ਰਾਹੀਂ ਨੰਗੀ ਮਸਤਾਂ ਦਾ ਪ੍ਰਚਾਰ ਕਰਦੀਆਂ ਹਨ।', ' ਹੁਣ ਟੀ.ਵੀ ਦੇ ਹਰ ਚੈਨਲ 'ਤੇ ਲੋਕਾਂ ਨੂੰ ਅਸ਼ਲੀਲਤਾ, ਭਰਾਉਂਦਾ ਜਾਂ ਅੰਧ-ਵਿਸ਼ਵਾਸ਼ੀ ਮਸਾਲਾ ਪਰੋਸਿਆ ਜਾ ਰਿਹਾ ਹੈ। ਜ਼ਿਆਦਾਤਰ ਚੈਨਲਾਂ 'ਤੇ ਬੂਬਣੇ ਬਾਬੇ ਸਵੇਰੇ ਹੀ ਵੱਡੇ-ਵੱਡੇ ਸੰਗਿਆਂ ਤੇ ਸਜ ਜਾਂਦੇ ਨੇ, ਫਿਰ ਗੁਰੂ ਹੁੰਦੀਆਂ ਨੇ ਇਹਨਾਂ ਦੀਆਂ ਝੂਠੀਆਂ ਅਤੇ ਅੰਧ-ਵਿਸ਼ਵਾਸੀ ਕਹਾਣੀਆਂ। ਉਸ ਪੰਜਾਬ ਦਾ ਕੀ ਬਣੂੰ ਜਿੱਥੇ ਸਮਾਜ ਨੂੰ ਸੇਧ ਦੇਣ ਵਾਲੇ ਲੋਕਾਂ ਦੇ ਆਦਰਸ਼ ਹੀ ਨੰਗੇ ਥੱਿਕੀਆਂ ਅਤੇ ਚਿਲਮਾਂ ਪੀਣ ਵਾਲੇ ਮਸਤ ਹੋਣ। ਇਹਨਾਂ ਦੇ ਸਿੱਖੇ ਚੇਲਿਆਂ ਨੇ ਪੜ੍ਹਾਈ ਵਿਚ ਸੁਆਹ ਮੱਲਾਂ ਮਾਰਨੀਆਂ ਹਨ? ਅਖੇ ਸਾਡਾ ਸਾਧ ਤਾਂ ਮਨ ਦੀ ਮੰਜ ਵਿਚ ਚਿਲਮਾਂ ਪੀਂਦਾ ਸੀ, ਟੁਹਾਨੂੰ ਵੀ ਮੰਜ ਹੈ ਤੇ ਨਾਲੇ ਚਿਲਮਾਂ ਵੇਚਣ ਵਾਲੇ ਦੀ ਮੰਜ ਬਣੀ ਹੈ।', ' ਦੇਸਤੇ, ਪੰਜਾਬੀ ਬਹਾਦਰ ਕੌਮ ਸੀ ਪਰ ਅਫ਼ਸੋਸ ਹੁਣ ਇਹ ਪਿਛਲੱਗ ਕੌਮ ਬਣ ਚੁੱਕੀ ਹੈ। ਨੰਜਵਾਨ ਦੇਖਾ ਦੇਖੀ ਹਰੇਕ ਚੰਰੀ ਮਾੜੇ ਕੰਮ ਪਿੱਛੇ ਲੱਗ ਰਹੇ ਹਨ। ਹਾਲੇ ਤਾਂ ਸੁਕਰ ਮਨਾਉ ਕਿ ਇਹ ਅਨਪੜ੍ਹ ਮਸਤ ਨਸ਼ਿਆਂ ਪਿੱਛੇ ਲੱਗੇ ਨੇ ਜੇ ਕੱਲ੍ਹ ਇਹ ਹੋਰ ਬਾਬਿਆਂ ਵਾਂਗ ਮੰਡ੍ਹੀਆ ਰਾਹੀਂ ਟੀ ਵੀ ਅਤੇ ਹੋਰ ਸਾਧਨਾਂ ਨਾਲ ਪ੍ਰਚਾਰ ਕਰਨ ਲੱਗ ਪਏ ਤਾਂ ਉਹ ਦਿਨ ਦੂਰ ਨਹੀਂ ਜਦੋਂ ਇਹਨਾਂ ਦਾ ਘਰ ਘਰ ਵਿਚ ਬੇਲਬਾਲਾ ਹੋਵੇਗਾ ਅਤੇ ਪਿਛਲੱਗ ਪੰਜਾਬੀਆਂ ਦੇ ਘਰਾਂ ਵਿਚ ਗੁਰੂ ਸਹਿਬਾਨਾਂ ਦੇ ਬਰਾਬਰ ਇਹਨਾਂ ਦੀਆਂ ਫੋਟੋ ਲੱਗਣਗੀਆਂ। ਅਸੀਂ ਦੇਖਦੇ ਰਹਿ ਜਾਵਾਂਗੇ। ਸੁਹਿਰਦ ਪੰਜਾਬੀਆਂ ਦੇ ਸਹਿਯੋਗ ਨਾਲ ਹੁਣ ਜਾਗਣ ਦੀ ਲੋੜ ਹੈ ਪਰ ਪਤਾ ਨਹੀਂ ਕੀ ਪੁੱਠੀ ਵਾ ਵਗੀ ਹੈ ਲੋਕਾ 'ਤੇ, ਪਤਾ ਨੀ ਕੀ ਹੋਇਆ ਲੋਕਾ ਦੀ ਸੋਚ ਨੂੰ।', ' ਇੱਕ ਤਾਂ ਆਹ ਦੇ ਤਿੱਨ ਮਸਤਾਂ ਦੇ ਚੇਲੇ ਗਾਇਕ ਰੋਗੀਆਂ ਵਿਚ ਭਜਾ ਭਜਾ ਕੁੱਟਣ ਵਾਲੇ ਆ।', ' ਫੇਰ ਪੁੱਛੋ ਮਸਤ ਨੰਤ੍ਰੇ ਹੈ ਕਿ ਘਸੁੰਨ? .. ਮਸਤਾਂ ਦੇ, ਕੀ ਲੱਛਣ ਫੜ੍ਹ ਐ।', ' ਦੇਸਤੇ ਇਹ ਕੋਈ ਸੱਚੀ ਸਮਝੀ ਚਾਲ ਨਹੀਂ ਬਲਕਿ ਭੱਭਚਾਲ ਹੈ। ਪੰਜ ਸੱਤ ਸਾਲ ਪਹਿਲਾਂ ਇਹ ਮਸਤ ਕਿੱਥੇ ਸਨ, ਬਸ ਦੇਖਾ ਦੇਖੀ ਮੁਲਖ ਭੁਹਿਆ ਜਾਂਦਾ ਅੰਨ੍ਹੇ ਖੂਹ ਵੱਲ।', ' ਫੇਸਬੁੱਕ 'ਤੇ ਤੰਜ਼ੇ ਕੁ ਦਿਨ ਕਿਸੇ ਦਾ ਸਟੇਟਸ ਜਾਂ ਫੋਟੋ ਪਾਈ ਹੁੰਦੀ ਹੈ, ਅਖੇ ਜੇ ਮਸਤਾਂ ਦੀ ਸੈ ਮਸਤਾਂ ਦੀ।', ' ਘਰੇ ਭਾਵੇਂ ਕੋਈ ਪਿਓ ਨੂੰ ਪਾਣੀ ਨਾ ਪੁੱਛੇ।', ' ਅਸੀਂ ਆਪਣੇ ਪਿੰਡ ਪੱਧਰ ਤੇ ਇਹਨਾਂ ਮਸਤਾਂ ਦੇ ਚੌਲਿਆਂ ਦਾ ਸਰਵੇਖਣ ਕੀਤਾ ਹੈ, ਉਹਨਾਂ ਤੋਂ ਜਾਣਕਾਰੀ ਲਈ ਹੈ ਕਿ ਪਹਿਲਾਂ ਤਾਂ ਲੋਕ ਆਪਣੇ ਹੱਥਾਂ ਵਿਚ ਸੁੱਕ ਨੂੰ ਲੈਹੇ, ਚਾਂਦੀ ਅਤੇ ਸੋਨੇ ਦੇ ਕੜੇ ਪਾਉਂਦੇ ਸਨ। ਇਹਨਾਂ ਕੜਿਆਂ ਦਾ ਆਪਣਾ ਇਤਿਹਾਸ ਵੀ ਹੈ ਅਤੇ ਇਹ ਪੰਜ ਕਕਾਰਾਂ ਵਿਚ ਵੀ ਆਉਂਦੇ ਹਨ ਪਰ ਟੁਹਾਡੇ ਪਾਈਆਂ ਇਹਨਾਂ ਲਾਲ ਪੀਲੀਆਂ ਵੰਗਾਂ ਦਾ ਕੀ ਕਾਰਣ ਹੈ? ਇ', ' ਦੇਸਤੇ ਇਹ ਕੋਈ ਸੱਚੀ ਸਮਝੀ ਚਾਲ ਨਹੀਂ ਬਲਕਿ ਭੱਭਚਾਲ ਹੈ।', ' ਪੰਜ ਸੱਤ ਸਾਲ ਪਹਿਲਾਂ ਇਹ ਮਸਤ ਕਿੱਥੇ ਸਨ, ਬਸ ਦੇਖਾ ਦੇਖੀ ਮੁਲਖ ਭੁਹਿਆ ਜਾਂਦਾ ਅੰਨ੍ਹੇ ਖੂਹ ਵੱਲ।', ['\n\nਫੇਸਬੁੱਕ 'ਤੇ ਤੰਜ਼ੇ ਕੁ ਦਿਨ ਕਿਸੇ ਦਾ ਸਟੇਟਸ ਜਾਂ ਫੋਟੋ ਪਾਈ ਹੁੰਦੀ ਹੈ, ਅਖੇ ਜੇ ਮਸਤਾਂ ਦੀ ਸੈ ਮਸਤਾਂ ਦੀ।', [' ਘਰੇ ਭਾਵੇਂ ਕੋਈ ਪਿਓ ਨੂੰ ਪਾਣੀ ਨਾ ਪੁੱਛੇ।', ' ਅਸੀਂ ਆਪਣੇ ਪਿੰਡ ਪੱਧਰ ਤੇ ਇਹਨਾਂ ਮਸਤਾਂ ਦੇ ਚੌਲਿਆਂ ਦਾ ਸਰਵੇਖਣ ਕੀਤਾ ਹੈ, ਉਹਨਾਂ ਤੋਂ ਜਾਣਕਾਰੀ ਲਈ ਹੈ ਕਿ ਪਹਿਲਾਂ ਤਾਂ ਲੋਕ ਆਪਣੇ ਹੱਥਾਂ ਵਿਚ ਸੁੱਕ ਨੂੰ ਲੈਹੇ, ਚਾਂਦੀ ਅਤੇ ਸੋਨੇ ਦੇ ਕੜੇ ਪਾਉਂਦੇ ਸਨ।', [' ਇਹਨਾਂ ਕੜਿਆਂ ਦਾ ਆਪਣਾ ਇਤਿਹਾਸ ਵੀ ਹੈ ਅਤੇ ਇਹ ਪੰਜ ਕਕਾਰਾਂ ਵਿਚ ਵੀ ਆਉਂਦੇ ਹਨ ਪਰ ਟੁਹਾਡੇ ਪਾਈਆਂ ਇਹਨਾਂ ਲਾਲ ਪੀਲੀਆਂ ਵੰਗਾਂ ਦਾ ਕੀ ਕਾਰਣ ਹੈ? ਇਹਨਾਂ ਦਾ ਕੀ ਇਤਿਹਾਸ ਹੈ? ਤਾਂ ਉਹਨਾਂ ਨੂੰ ਇਸ ਬਾਰੇ ਕੋਈ ਜਾਣਕਾਰੀ ਨਹੀਂ ਸੀ।', [' ਸਣੇ ਇਹ ਸਭ ਇੱਕ ਦੂਜੇ ਦੀ ਦੇਖਾ ਦੇਖੀ ਕਰ ਰਹੇ ਸਨ।', [' ਵੰਗਾ ਛੱਡੋ, ਹੁਣ ਤਾਂ ਇਹ ਮਸਤ ਦੇ ਚੇਲੇ ਆਪਣੇ ਪੈਰਾ ਵਿਚ ਘੁੰਗਰੂ ਵੀ ਪਾਉਣ ਲੱਗ ਪਏ ਹਨ।', [' ਇਹਨਾਂ ਘੁੰਗਰੂਆਂ ਬਾਰੇ ਇਹਨਾਂ ਦੇ ਵਿਚਾਰ ਹਨ ਕਿ ਸਾ ਭ ਸਾਈਂ ਦੇ ਸਭ ਤੋਂ ਵੱਡੇ ਚੇਲੇ ਮਹਾਨ ਗਾਇਕ ਦੇ ਪੈਰਾ ਵਿਚ ਇਹ ਘੁੰਗਰੂ ਪਏ ਹੁੰਦੇ ਨੇ, ਉਸਦੀ ਦੇਖਾ ਦੇਖੀ ਅਸੀਂ ਵੀ ਪਾ ਲਏ ਅਤੇ ਸਾਈਂਆਂ ਦੀ ਕੱਜਰੀ ਬਣ ਗਏ।', [' ਅਖੇ ਇਸੇ ਲਈ ਤਾਂ ਸਾਡੇ ਤੇ ਗੀਤ ਆਇਆ ਹੈ ਕਿ "ਮੈਂ ਸਾਈਆਂ ਦੀ ਕੱਜਰੀ ਹਾਂ"'], [' ਇਹਨਾਂ ਚੇਲਿਆਂ ਨੂੰ ਸਾਈਆਂ ਦੇ ਭੰਗਿਆਂ ਦੇ ਇਤਿਹਾਸਿਕ ਪਿਛੋਕੜ ਬਾਰੇ ਉੱਕਾ ਹੀ ਜਾਣਕਾਰੀ ਨਹੀਂ ਸੀ।', [' ਉੱਤੋਂ ਸਿਤਮ ਇਹ ਹੈ ਕਿ ਇਹਨਾਂ ਕਦੇ ਸਾਈਆਂ ਦੇ ਭੰਰੇ ਵੀ ਜਾ ਕੇ ਨਹੀਂ ਦੇਖਿਆ।', [' ਇਹਨਾਂ ਦੇ ਸਿਰ ਤੇ ਸਾਈਆਂ ਦਾ ਭੂਤ ਸਿਰਫ ਦੇਖਾ ਦੇਖੀ ਆਸੇ ਪਾਸ ਿਰਿ ਸੁਣ ਕੇ ਜਾ ਮਸਤਾਂ ਦੇ ਆ ਰਹੇ ਗੀਤਾਂ ਕਰਕੇ ਚੜ੍ਹਿਆ ਹੈ।', ['\n\nਮੁੱਕਦੀ ਗੱਲ, ਸਾਡੀ ਅਣਗਹਿਲੀ ਕਰਕੇ ਇਹ ਸਾਈਂਪੁਣਾ ਜਾਂ ਮਸਤਪੁਣਾ ਇਨੇ ਦਿਨ ਬਹੁਤ ਖ਼ਤਰਨਾਕ ਤਰੀਕੇ ਨਾਲ ਫੈਲ ਰਿਹਾ ਹੈ।', [' ਸਾਡੇ ਪੰਜਾਬੀਆਂ ਦੀ ਇਹ ਫਿਤਰਤ ਰਹੀ ਹੈ ਕਿ ਜਿੱਨਾ ਚਿਰ ਅੱਗ ਸਾਡੇ ਘਰ ਤੱਕ ਨਹੀਂ ਪਹੁੰਚਦੀ ਉੰਨਾ ਚਿਰ ਸਾਨੂੰ ਜਾਗ ਨਹੀਂ ਆਉਂਦੀ ਜਾਂ ਅਸੀਂ ਮੌਸਟ ਬਣ ਕੇ ਗਵਾਂਢੀਆਂ ਦੀ ਲੱਗੀ ਅੱਗ ਵੇਖ ਕੇ ਅੰਦਰ ਵੜ ਜਾਂਦੇ ਹਾਂ।', [' ਪਰ ਪਤਾ ਉਦੋਂ ਲੱਗਣਾ ਹੈ ਜਦੋਂ ਖੁਦ ਨੂੰ ਸੇਕ ਲੱਗਿਆ।', [' ਜੇ ਇਵੇਂ ਹੀ ਸ਼ਰੀਕਾਂ ਦੀ ਨਿੱਕਲੀ ਕੁੜੀ ਨੂੰ ਵੇਖ ਵੇਖ ਝਾੜੀਆਂ ਮਾਰਦੇ ਰਹੇ ਤਾਂ ਉਹ ਦਿਨ ਦੂਰ ਨਹੀਂ ਜਦੋਂ ਸਾਡੇ ਆਪਣੇ ਜਵਾਕਾ ਨੇ ਕੰਨ ਪੜ ਕੇ ਮੁੰਦਰਾਂ ਪਾ ਲੈਣੀਆਂ ਹਨ।', [' ਲਾਲ ਪੀਲੀਆਂ ਵੰਗਾ ਪਾ ਕੇ ਪੈਰਾ ਵਿਚ ਘੁੰਗਰੂ ਅਤੇ ਹੱਥਾ ਵਿਚ ਚਿਲਮਾਂ ਫੜ ਲੈਣੀਆਂ ਹਨ।', [' ਜਦੋਂ ਸਾਡੇ ਆਪਣੇ ਘਰਾਂ ਵਿਚ 'ਮਸਤ ਮਸਤ' ਹੋਣ ਲੱਗ ਪੈਣੀ ਹੈ ਫਿਰ ਅਸੀਂ ਪਿੰਟਣਾ ਹੈ ਪੈਣਾ ਤੇ ਹੱਥ ਮਾਰ ਮਾਰ ਕੇ।', [' ਜੇ ਇਸੇ ਤਰ੍ਹਾਂ ਹੀ ਰਿਹਾ ਤਾਂ ਪਿੰਛ ਇਹਨਾਂ ਮਸਤਾਂ ਦੇ ਭੰਰੇ ਬਣਦੇ ਦੇਖ ਲਿਓ।', ['\n\nਜਲੰਧਰ ਨਕੇਦਰ ਵੱਲ ਤਾਂ ਇਹ ਕੰਮ ਸ਼ੁਰੂ ਹੋ ਗਿਆ ਹੈ ', [' ਉੱਪਰ ਤਾਂ ਲੋਕਾਂ ਸੰਤ ਸ਼ਾਂ ਅਕਾਲ ਅਤੇ ਨਮਸਤੇ ਬੁਲਾਉਣੀ ਛੱਡ ਕੇ 'ਜੈ ਮਸਤਾਂ ਦੀ' ਕਹਨੀ ਸ਼ੁਰੂ ਕਰ ਦਿੱਤੀ ਹੈ।', [' ਆਪਣੇ ਵੀ ਦਿਨ ਨੇੜੇ ਹੀ ਹਨ।', [' ਸਮਝਣਾ ਆਪਾਂ ਨੂੰ ਹੀ ਪੈਣਾ ਹੈ, ਸਿਆਣੇ ਆਪਾਂ ਨੂੰ ਹੀ ਹੋਣਾ ਪੈਣਾ ਹੈ।', [' ਆਪਣੇ ਬੱਚਿਆਂ ਨੂੰ ਸਮਝਾਉਣਾ ਪੈਣਾ ਹੈ।', ['\nਲੇਖਕ, ਕਲਾਕਾਰੀ, ਭਰੇ ਨਾ, ਬਾਹਰ ਆਓ, ਐਵੇਂ ਨਾ ਦੇਖਾ ਦੇਖੀ ਭੀੜ ਦਾ ਹਿੱਸਾ ਬਣੀਂ ਜਾਓ।', ['\n**\n(6315)\nਕਰਨ ਬਰਾੜ (ਹਰੀਕੇ ਕਲਾ)\n9988040642']]
```

Figure 9: Sorting Weights and Percentage of Summary Needed

6.2 Tokenization

Steps involved in section 6.2 are:-

1. List of lists in which segmented data is there, is further tokenized on the basis of space between two tokens (words).
2. Split() function is used for tokenization.
3. Tokenized output is stored in list of lists as shown in Figure 10.

Figure 10: Tokenization

6.3 Stop Words Removal

Steps involved in section 6.3 are:-

1. The output of tokenization phase stored in list of lists is further used by stop words removal phase.
2. From list of lists, stop words are removed by GTSS.
3. After removing stop words, approximately 45% words are left for processing phase, the words left are useful words as shown in Figure 11.
4. The output of stop words removal phase is stored in list of lists for further processing.

Figure 11: Stop Words Removal

6.4 Root Words Identification

Steps involved in section 6.4 are:-

1. Output of stop words removal phase is further processed by stemmer.
2. Output list of stop words removal phase is used, where words are stemmed towards their corresponding root as shown in Figure 12.
3. The output of stop words removal phase is stored in list of lists for further processing.

Figure 12: Root Word Identification

6.5 Calculating Features Weights

Steps involved in section 6.5 are:-

1. After pre-processing phase, list of lists is further used for processing phase.
2. In processing phase, a weight is calculated for every feature and new vector is created for every feature as shown in Figure 13.
3. A positive or negative weight is assigned to every sentence according to the selected feature, which are explained in previous chapters.

```
Processing phase

1. feature sentence position  [0.9761904761904762, 0.9523809523809523, 0.9285714285714286, 0.9047619047619048, 0.8809523809523809, 0.8571428571428571, 0.83333333333333
34, 0.8095238095238095, 0.7857142857142857, 0.7619047619047619, 0.7380952380952381, 0.7142857142857143, 0.6904761904761905, 0.6666666666666666, 0.6428571428571429, 0.6
190476190476191, 0.5952380952380952, 0.5714285714285714, 0.5476190476190477, 0.5238095238095238, 0.5, 0.47619047619047616, 0.4523809523809524, 0.42857142857142855, 0.4
047619047619047 7, 0.38095238095238093, 0.35714285714285715, 0.3333333333333333, 0.30952380952380953, 0.2857142857142857, 0.2619047619047619, 0.23809523809523808, 0.214
2857142857142 7, 0.19047619047619047, 0.16666666666666666, 0.14285714285714285, 0.11904761904761904, 0.09523809523809523, 0.07142857142857142, 0.047619047619047616, 0.0
23809523809523808, 0.0]

2. feature sentence lenght  [0.3157894736842105, 0.42105263157894735, 0.05263157894736842, 0.07894736842105263, 0.05263157894736842, 0.5526315789473685, 0.710526315789
4737, 0.18421052631578946, 0.13157894736842105, 0.18421052631578946, 0.13157894736842105, 0.15789473684210525, 0.18421052631578946, 0.23684210526315788, 0.342105263157
89475, 0.15789473684210525, 0.2631578947368421, 0.2631578947368421, 0.21052631578947367, 0.23684210526315788, 0.2894736842105263, 0.42105263157894735, 0.368421052631578
89, 0.42105263157894735, 0.23684210526315788, 0.34210526315789475, 0.18421052631578946, 0.3157894736842105, 0.47368421052631576, 0.6052631578947368, 0.236842105263157
8, 0.23684210526315788, 0.3157894736842105, 0.2894736842105263, 0.34210526315789475, 0.2894736842105263, 0.631578947368421, 0.4473684210526316, 1.0, 0.5, 0.57894736842
10527, 0.39473684210526316]

3.feature presence of numerical data in sentences  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0.14285714285714285]

4. feature presence of cue phrases in sentences  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0]

5. feature presence of nouns in sentences  [0.13636363636363635, 0, 0.058823529411764705, 0.02631578947368421, 0.058823529411764705, 0, 0.043478260869565216, 0.0285714
2857142857, 0.07142857142857142, 0, 0.10526315789473684, 0, 0.21428571428571427, 0.0625, 0, 0, 0.058823529411764705, 0.05, 0.125, 0.14285714285714285, 0.025, 0, 0, 0.0
2631578947368421, 0, 0.13333333333333333, 0, 0.08695652173913043, 0, 0.10810810810810811, 0.1, 0.03333333333333333, 0.06666666666666667, 0.09523809523809523, 0, 0.1, 0
.09090909090909091, 0, 0, 0.16666666666666666, 0.06666666666666667, 0.2857142857142857]

6.1. keywords in Input Text Document  ['ਮਸਤ', 'ਇਹਨ', 'ਸਾਡਾ', 'ਦੇਖਾ']

6.2. feature presence of keywords words in sentences  [0.0, 0.0, 0.047619047619047616, 0.1111111111111111, 0.0, 0.0625, 0.08333333333333333, 0.08695652173913043, 0.0,
0.1111111111111111, 0.07894736842105263, 0.0, 0.0, 0.09090909090909091, 0.18181818181818182, 0.0, 0.15384615384615385, 0.16666666666666666, 0.0, 0.11764705882352941, 0
.15789473684210525, 0.2, 0.1111111111111111, 0.18181818181818182, 0.125, 0.1111111111111111, 0.3333333333333333, 0.23076923076923078, 0.0, 0.041666666666666664, 0.0, 0
.05555555555555555, 0.0, 0.07142857142857142, 0.2, 0.0, 0.125, 0.0, 0.0, 0.0, 0.07692307692307693, 0.0]

7. Punjabi Title feature  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

8. Cosine Similarity Between Two Sentences  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0]

9. feature Punjabi-English Common Words in sentences  [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.08333333333333333, 0.0, 0.1111111111111111, 0.0, 0.05263157894736842, 0.0, 0.066
66666666666667, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.041666666666666664, 0.0, 0.0, 0.0, 0.0, 0.0, 0.2, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0]

10. feature presence of web address in sentences  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0]

11. feature presence of brackets in sentences  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 2]

12. feature presence of quotes in sentences  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0]
```

Figure 13: Calculating Features Weights

6.6 Normalization Phase

Steps involved in section 6.6 are:-

1. After processing phase, 12 vectors of features are needed to be normalized.
2. In normalized phase, weight of every sentence for every feature is calculated and normalized as shown in Figure 14.

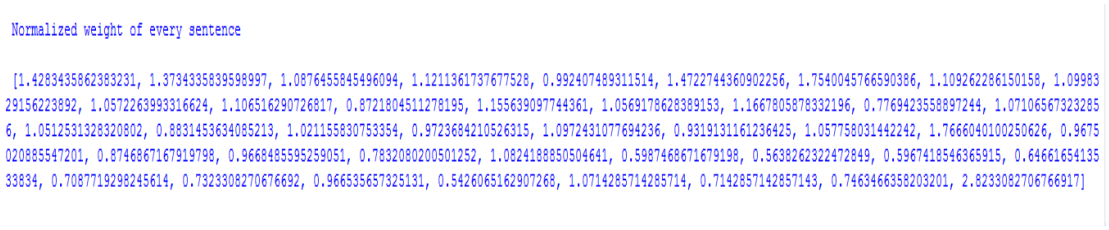


Figure 14: Normalization Phase

6.7 Sorting Weights and Percentage of Summary Needed

Steps involved in section 6.7 are:-

1. Normalized weights are sorted in decreasing order.
2. Sentences having weights on the top of the list are most important sentences.
3. Higher weighted sentences are extracted for summary according to the requirement of the user, it may be 20%, 45% etc.
4. Sentences are extracted in same manner in which they appeared in original document, in order to maintain unambiguous summary as shown in Figure 15.

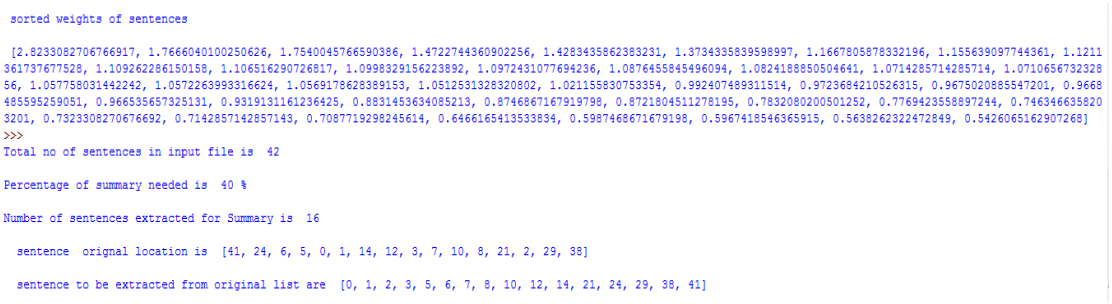


Figure 15: Sorting Weights and Percentage of Summary Needed

6.8 Summary Creation

Steps involved in section 6.8 are:-

1. Sentences are extracted from original list, which is created in chapter 3.1.
2. Extracted sentences are shown in GUI as shown in Figure 16.

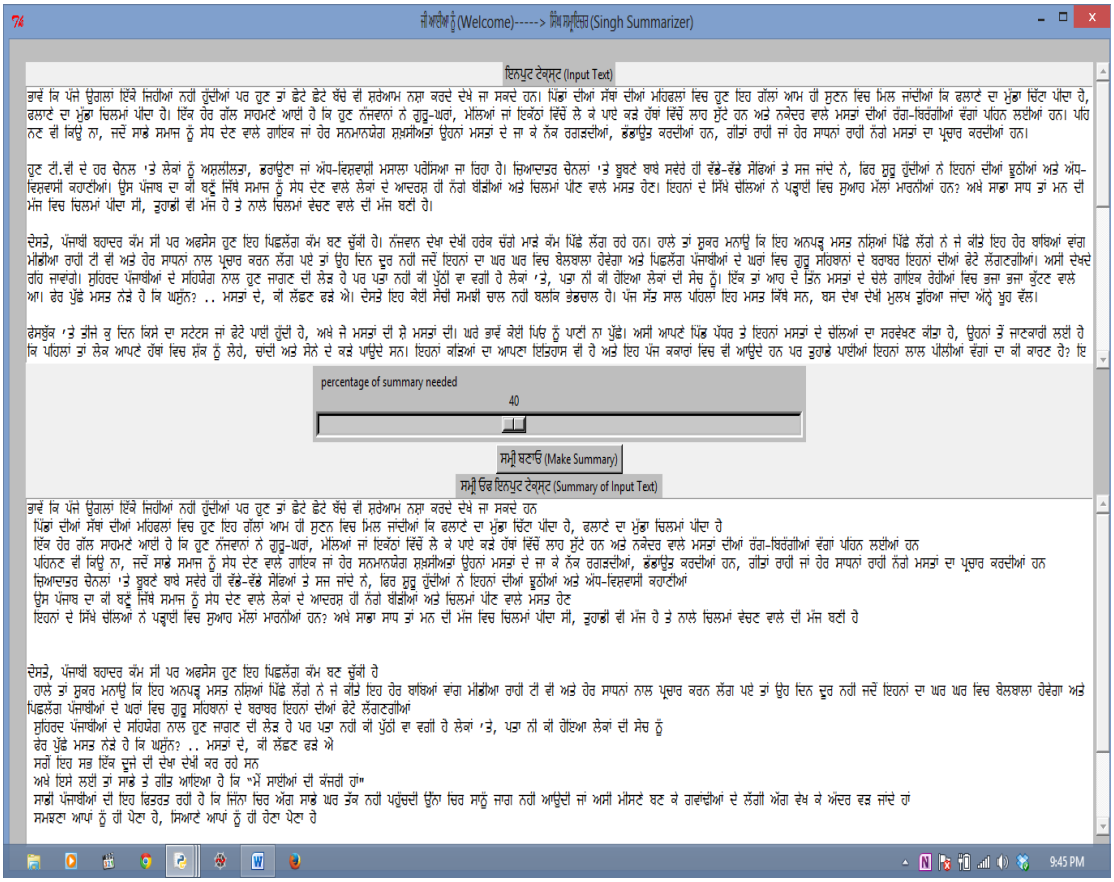


Figure 16: Summary (GUI)

Chapter 7: Evaluation of Summary

It is necessary to check accuracy, relevance and usefulness of the summary created by GTSS. In this paper, Precision, Recall and F-score are used to evaluate proposed system.

7.1 Precision

Precision is no of correct sentences divided by no of all sentences extracted.

$$P = \frac{\{\text{Relevant Sentences}\} \cap \{\text{Extracted Sentences}\}}{\{\text{Extracted Sentences}\}}$$

7.2 Recall

Recall is no of correct sentences divided by no of sentences that should have been extracted.

$$R = \frac{\{\text{Relevant Sentences}\} \cap \{\text{Extracted Sentences}\}}{\{\text{Relevant Sentences}\}}$$

7.3 F-score

F-score is the harmonic mean of precision and recall.

$$F = \frac{\{2 * P * R\}}{\{P + R\}}$$

Where Extracted Sentences are extracted by the system and Relevant Sentences are identified by human.

Chapter 8: Results and discussion

GTSS has been tested over twenty Gurmukhi Documents consists of news, stories and articles etc. taken randomly from Internet. In our experiment, we use most common sentences of summaries created by four human experts manually at compression ratios 20%, 40% and 70% are taken as benchmark summaries (reference summaries).

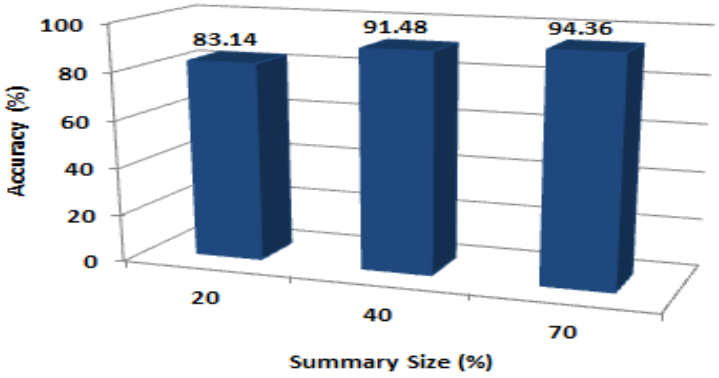


Figure 1: Summary Accuracy vs. Summary Size

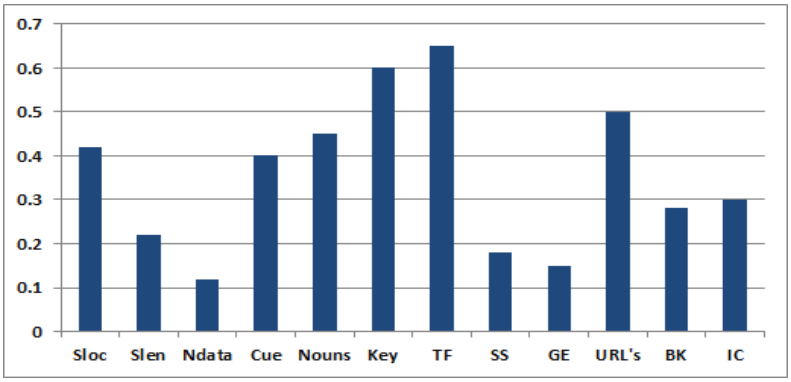


Figure 2: Features Weights Comparison

It has been noticed that GTSS performs well and achieves high rates over 90% accurate. In the category of news articles for compression ratio of 20% the accuracy achieved was 82.04%, and for 40% and above it touched the 92% mark in terms of

accuracy. For stories, GTSS achieves highest accuracy for compression ratio 40% and above in the range above 94%. In case of article published online the accuracy achieved by GTSS was around 91% for a compression ratio of 40% and above. In conclusion, GTSS achieved maximum accuracy in terms of F-Score for compression ratio of 40% and above for news, stories and articles etc. and perform fairly at 20% compression ratio, due to filtration of maximum information. More interestingly, it can also be concluded that the accuracy of GTSS increases as summary size increases as shown in Figure 1. Figure 2 shows the average weights of all the features identified in the surveyed text documents including news, stories and articles. The proposed URL feature emerges as an important feature for summarization in context to the amount of Gurmukhi text available online after doing analysis of various Punjabi documents.

Chapter 9: Conclusion

This thesis discusses single document automated Gurmukhi text summarization system which is implemented using Python 3.3.3. In Proposed system most of the used resources such as list of Gurmukhi nouns, Gurmukhi-English Common Words list, Stop Word list and Gurmukhi Root Word Identification algorithm etc. were developed and implemented. There are no ready resources available openly over the internet because of very less research work is carried out for Gurmukhi text summarization system. After doing analysis of various Gurmukhi newspapers and articles some new features are also developed such as Presence of URL's or Email Addresses, Presence of Brackets and Presence of Inverted Commas in sentences etc. which are also important features of text and shows good results by including them.

Chapter 10: Future Scope and Challenges

10.1 Future scope

In the future, GTSS may further be extended to multi document summarization. Quality of the summary may further be improved by implementing optimization techniques, some semantic features and some others features given below:-

1. Text summarization tools can be linked with various applications available online for summarizing data.
2. Abstractive approaches to text summarization can be added to improve quality of the summary to large extent.
3. Summarization approaches can be implemented for other multimedia such as audio, video etc.
4. Text summarization task can be extended to other Indian languages also.

10.2 Challenges

There are a lot of challenges which needs to be resolved such as: -

1. A corpus for Punjabi stop words is not available.
2. Features for Punjabi language are different and are very difficult to process.
3. Pronoun level ambiguity is very difficult to remove.

Chapter 11: References

[1] Vinod Chandra SS and Achuth Sankar S Nair and Vrinda V Nair and Mahalekshmi T. A Linguistic Coloring Editor for Processing Multilingual Text Corpora using Hidden Markov Models. *Journal of Computer Society of India (ISSN 0254-7813)*, 37(3):8–11, 2007.

[2] Achuth Sankar S Nair and Vrinda V Nair and Vinod Chandra SS. Hidden Markov Model Based Identification of Transliterated Regional Language Words in Text Documents. *Twentieth Interna-tional Joint Conference on Artificial Intelligence* , pages 87–91, 2007.

[3] Dipanjan Das and André FT Martins. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, 4:192–195, 2007.

[4] Harold P Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.

[5] Vishal Gupta and Gurpreet Singh Lehal. Punjabi language stemmer for nouns and proper names. In *Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), IJCNLP*, pages 35–39, 2011.

[6] Vishal Gupta and Gurpreet Singh Lehal. Automatic punjabi text extractive summarization system. In *COLING (Demos)*, pages 191–198, 2012.

[7] Hongyan Jing and Kathleen R McKeown. Cut and paste based text summarization. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 178–185. Association for Computational Linguistics, 2000.

[8] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[9] Inderjeet Mani. *Automatic summarization*, volume 3. John Benjamins Publishing, 2001.

[10] Inderjeet Mani and Mark T Maybury. *Advances in automatic text summarization*. the MIT Press, 1999.

[11] Kamal Sarkar. Bengali text summarization by sentence extraction. *arXiv preprint arXiv:1201.2240*, 2012.

[12] Ladda Suanmali, Naomie Salim, and Mohammed Salem Binwahlan. Fuzzy genetic semantic based text summarization. In *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, pages 1184–1191. IEEE, 2011.

[13] Chetana Thaokar and Latesh Malik. Test model for summarizing hindi text using extraction method. In *Information & Communication Technologies (ICT), 2013 IEEE Conference on*, pages 1138–1143. IEEE, 2013.

[14] Rene Arnulfo Garcia-Herandez and Yulia Ledeneva, "Word Sequence Models for Single Text Summarization," IEEE, 44-48, 2009.

[15] R. C. Balabantaray, D. K. Sahooo, B. Sahoo and M. Swain, "Text Summarization using Term Weights,'' International Journal of Computer Applications, vol. 38, no. 1, pp. 10-14, 2012.

[16] Ramakrishna Varadarajan and Vagelis Hristidis, "Structure-Based Query-Specific Document Summarization", in proceedings of CIKM'05, ACM, Bremen, Germany, 2005.

[17] Hal Daum´e III and Daniel Marcu, "Bayesian Query-Focused Summarization", Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, 305–312, Sydney, July 2006.

[18] Feng Jin, Minlie Huang and Xiaoyan Zhu, " A Queryspecific Opinion Summarization System", in proceedings ofICCI '09, 8th IEEE international conference on cognitive informatics, Kowloon, Hong Kong, 428-433, 2009.

[19] Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslami and Pooya Khosravyan Dehkordy, "Optimizing Text Summarization Based on Fuzzy Logic", In proceedings of Seventh IEEE/ACIS International Conference on Computer and Information Science, IEEE, University of Shahid Bahonar Kerman, UK, 347-352, 2008.

[20] S Mangairkarasi and S Gunasundari, "Semantic based Text Summarization using Universal Networking Language," *International Journal of Applied Information Systems (IJAIS), vol.3, No.8, 2012.*

[21] P. B. Baxendale, "Machine-made Index for Technical Literature -An Experiment," *Journal of Research and IBM Development*, vol. 2, no. 4, pp. 354-361, October 1958.

[22] Paul S Jacobs and Lisa F Rau, "SCISOR: Extracting information from on-line," *Communications of the ACM*, Vol.33, no.11, pp. 88-97, 1990.

[23] Julian Kupiec, 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, pp. 68-73, 1995.

[24] Eduard Hovy and Chin-Yew Lin, "Automated text summarization and the SUMMARIST system," Proceedings of a workshop on held at Baltimore, pp. 197-214, 1998.

[25] Chin-Yew Lin, "Training a selection function for extraction," Proceedings of the eighth international conference on Information and knowledge management, ACM, pp. 55-62, 1999.

[26] Jade Goldstein, Vibhu Mittal, Jaime Carbonell and Mark Kantrowitz, "Multi-document summarization by sentence extraction," Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization, vol.4, pp. 40-48, 2000.

[27] Hongyan Jing, "Sentence Reduction for Automatic Text Summarization," In Proceedings of the 6th Applied Natural Language Processing Conference, Seattle, USA, pp. 310-315, 2000.

[28] M.J Conroy and O'leary, "Text summarization via hidden markov models," In Proceedings of SIGIR '01, pp. 406-407, 2001.

[29] Dragomir R Radev, Hongyan Jing and et al, "Centroid-based summarization of multiple documents," *Information Processing & Management*, Elsevier, vol.40, no.6, pp.919-938, 2004.

[30] Md Haque, Suraiya Pervin and others, "Literature Review of Automatic Single Document Text Summarization Using NLP," *International Journal of Innovation and Applied Studies*, vol.3, no.3, pp. 857-865, 2013.

[31] Krysta Marie Svore, Lucy Vanderwende and Christopher JC Burges, "Enhancing Single-Document Summarization by Combining RankNet and Third-Party Sources," EMNLP-CoNLL, pp. 448-457, 2007.

[32] Ladda Suanmali, Mohammed Salem Binwahlan and Naomie Salim, "Sentence features fusion for text summarization using fuzzy logic," Ninth International Conference on Hybrid Intelligent Systems, IEEE, vol.1, pp.142-146, 2009.

[33] Li Chengcheng, "Automatic Text Summarization Based On Rhetorical Structure Theory," International Conference on Computer Application and System Modeling (ICCASM), vol. 13, pp. 595-598, October 2010.

[34] Ladda Suanmali, Naomie Salim and Mohammed Salem Binwahlan, "Fuzzy genetic semantic based text summarization," Ninth International Conference on Dependable Autonomic and Secure Computing (DASC), IEEE }, pp. 1184-1191, 2011.

[35] Manuel Baena, and J.Carmona and others, "TF-SIDF: Term frequency, sketched inverse document frequency," 11th International Conference on Intelligent Systems Design and Applications (ISDA), IEEE, pp. 1044-1049, 2011.

[36] Mohamed Abdel Fattah and Fuji Ren, "Automatic Text Summarization," *International Journal of Computer Science*, vol.3, no.1, 2008.

[37] Inderjeet Mani and Eric Bloedorn, "Multi-document summarization by graph search and matching," arXiv preprint cmp-lg/9712004, 1997.

Chapter 12: List of Publications

[1] Gurmeet Singh and Karun Verma, "A Novel Features Based Automated Gurmukhi Text Summarization System," *International Conference on Advance in Computing, Communication and Information Science,* Elsevier, 2014. **(Accepted)**