TEXT SUMMARIZER

A Major Project Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

BY

SHABBIR SIDHPURWALA
EN16CS301239

SAIYAM JAIN
EN16CS301221

Under the Guidance of

Ms. Sushma Verma

Department of Computer Science and Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331

JUNE 2020

TEXT SUMMARIZER

A Major Project Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

BY

SHABBIR SIDHPURWALA
EN16CS301239

SAIYAM JAIN
EN16CS301221

Under the Guidance of

Ms Sushma Verma

Department of Computer Science and Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331

JUNE 2020

Report Approval

The project work **"TEXT SUMMARIZER"** is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approve any statement made, opinion expressed, or conclusion drawn there in; but approve the "Project Report" only for the purpose for which it has been submitted.

Internal Examiner:

Name:

Designation:

Affiliation:

External Examiner

Name:

Designation:

Affiliation:

Declaration

I hereby declare that the project entitled **"TEXT SUMMARIZER"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology of Computer Applications in 'Computer Science and Engineering' completed under the supervision of **Ms. Sushma Verma, Assistant Professor, Computer Science and Engineering Department,** Medi-Caps University Indore is an authentic work.

Further, I declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Shabbir Sidhpurwala	**Saiyam Jain**
EN16CS301239	**EN16CS301221**

Certificate

I, **Ms. Sushma Verma** certify that the project entitled "**Text Summarizer"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology by **Shabbir Sidhpurwala and Saiyam Jain** is the record carried out by them under my guidance and that the work has not formed the basis of award of any other degree elsewhere.

_____	_____
Mr. Sushma Verma	Dr. Suresh Jain
Assistant Professor	Head of the Department
Computer Science and Engineering	Computer Science and Engineering
Medi-Caps University, Indore	Medi-Caps University, Indore

Acknowledgements

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal**, who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Sunil K Somani**, Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank Prof. **(Dr.) D K Panda**, Dean, Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Prof. (Dr.) Suresh Jain** for his continuous encouragement for betterment of the project.

We express my heartfelt gratitude to my Internal Guide, Ms. Sushma Verma, Assistant Professor, Department of Computer Science and Engineering CSE, without whose continuous help and support, this project would ever have reached to the completion.

Moreover, we express our heartfelt gratification to our Project Co-Ordinator, Mr. Lakhan Singh and Mr. Arpit Deo, Department of Computer Science and Engineering, Medi-Caps University, without whose continuous help and support, this project would ever have reached to the completion.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

Shabbir Sidhpurwala
EN16CS301239
B.Tech. IV Year
Department of Computer Science
Faculty of Engineering
Medi-Caps University, Indore

Saiyam Jain
EN16CS301221
B.Tech. IV Year
Department of Computer Science
Faculty of Engineering
Medi-Caps University, Indore

Abstract

Today's world is all about information, with most of it online which enables anytime, anywhere, easy and unlimited access; participation & publishing of information has consequently escalated the suffering of 'Information Glut'. Assisting users' informational searches with reduced reading or surfing time by extracting and evaluating accurate, authentic & relevant information are the primary concerns in the present milieu. Automatic text summarization is the process of condensing an original document into shorter form to create smaller, compact version from the abundant information that is available, preserving the content & meaning such that it meets the needs of the user. Though many summarization techniques have been proposed but there are no 'silver bullets' to achieve the superlative results as of human generated summaries. Thus, the domain of text summarization is an active and dynamic field of study, practice & research with the continuous need toexpound novel techniques for achieving comparable & effectual results.

Fuzzy logic has appeared as a powerful theoretical framework for studying human reasoning and its application has been explored within the domain of text summarization in the past few years. One key aspect of text summarization is accurate identification of keywords from the given textual content. In this project, a new technique based on fuzzy logic has been proposed using two graph based techniques named as TextRank and LexRank and one semantic based technique named as Latent semantic analysis(LSA).In our work, we have also investigated their relative performance with the proposed method. All these methods used in developing hybrid model are of extractive summarization type. The techniques are evaluated on Opinosis data set using 'ROUGE-1'and 'time to extract the keywords'. The proposed technique has outperformed the existing techniques, wen compared with the results given by the original studies.

Keywords: Fuzzy Logic, Latent Semantic Analysis, Lex-Rank, Text-Rank.

Table of Contents

List of Figures

List of Tables

Abbreviations

Abbreviations	Description
IT	Information Technology
App	Application
Pvt.	Private
Ltd.	Limited
SDK	Software Development Kit
API	Application Program Interface
PDF	Portable Document File
XML	extensible Markup Language
CV	Curriculum Vitae

CHAPTER 1

INTRODUCTION

1.1 Introduction

According to www.worldwidewebsize.com, the indexed Web contains at least 6.08 billion pages (24 September, 2019). With the massive proliferation in the velocity, volume and variety of information accessible online and the consequent need to develop viable paradigms which facilitate better techniques to access this information, there has been a strong resurgence of interest in Web Information Retrieval (Web IR) research in recent years. The ultimate challenge of Web IR research is to provide improved systems that retrieve the most relevant information available on the web to better satisfy a user's information need. Moreover, with the transformation of Web into a customary decision support and recommendation tool, tackling the challenge of "information overload" on the Web has become increasingly vital. The Web IR research is typically organized in tasks with specific goals to be achieved. With the ease in availability of Internet and increased use of smart devices like mobile phone, laptop, tablets the access and storage of data has escalated rapidly, resulting in need of tools which can enhance the user's productivity and experience. A simple keyword search on the Internet results in hundreds of results in less than a second, some of which are not even relevant to the users' query and finding the pertinent information is a difficult and time consuming task.

Summarization has been identified as an effective Web IR task, which helps users to locate the right information at the right time thus facilitating timely decisions. Human's summarization can be biased, context-dependent and may vary with human cognition. Thus, suitable techniques & tools are needed to extort pertinent and imperative sections such that critical information in the form of summary is acquired; providing a machine generated summary free from bias. The idea is to create smaller, condensed versions from the abundant information that is available, preserving the content & meaning such that it meets the needs of the user.

Automatic text summarization techniques can be used for only extracting the keyword too. As, no matter what the intent, type or context of summary generated is, the primary objective is to assist users' informational searches with reduced reading/surfing time and also improve the document indexing efficiency at the same time. This will fasten the search process, as the relevance of an article to our interested topic can be deducted by the important keywords of the article. Extracting keywords using text summarization algorithm can optimize the search process.

A text summarization method has been proposed which is a hybrid of four techniques namely TextRank, LexRank, Latent Semantic Analysis and Fuzzy logic each having their own pros and cons. Unitedly, resulting in more confident results.

1.2 Literature Review

In this paper, an automated method for text assessment that starts with text summarization and Text Summarization using fuzzy rules then compares the automatically generated summaries with reference texts provided by domain experts. The proposed summarization method is a fuzzy rule-based system that identifies and selects the most informative sentences and concepts in each text document. The proposed method can combine features extracted from the text by using a multivariate linear regression model. It reduces the number of fuzzy rules without performance degradation. Also read some research paper for this which includes text summarizer with help of Fuzzy logic, lexRank, Textrank and Latent semantic analysis. All are generating it own result independently. I will be going to discuss some standard paper Like in fuzzy logic we get how this algorithm worked on it and produce the result. First, we talk about fuzzy logic. Text summarization has become a necessary and efficient tool for interpreting huge amount of digital information. The text summarization system under discussion makes use of fuzzy logic approach to generate the summary. The term fuzzy logic was introduced with by [Zad]. It is a mathematical tool used for dealing with uncertainty, imprecision, ambiguity and vagueness. Fuzzy logic is a form of many-valued logic that deals with approximate reasoning rather than fixed and exact reasoning.

According to Timothy Ross[Tim], Fuzzy logic is a mathematical model in which truth can be partial i.e. it can have value between 0 and 1 as compared to traditional binary logic where variables have only two values i.e. 0 or 1. Fuzzy logic is used to handle the concept of partial truth where its truth value ranges between completely true and completely false. For e.g. consider a variable such as age which may have a non-numeric value such as young or old. Such variables are called linguistic variables.

The main feature of fuzzy logic in text summarization is that it is able to deal with impreciseness and uncertainty of features in deciding the importance of sentences to be extracted into the summary.

We read another research paper like latent semantic analysis and Lexrank algorithm and identified how this algorithm helps me in my project or idea.

1.3 Objectives

To implement the problem statement, the following objectives have been defined:

- Use best combination of minimum features in feature extraction
- Integrate lexical chaining to capture semantic meaning or the central theme of the document achieving cohesiveness in the summary.
- To capture correlation of sentences using anaphora resolution to achieve coherence in the summary.
- To use intrinsic method of evaluation as it is more suitable for extractive summarization.
- Use Java version of ROUGE tool (n-gram) for evaluation of summary generated in terms of performance metrics precision, recall & F-measure.
- The summary generated should contain the significant information along with the topic coverage which quickly enables the user to quickly comprehend vast amount of information.

1.3.1 Research Gap

After going through the literature survey in automatic text summarization, it is observed that till now most of the authors have used single approach to perform automatic text summarization using extraction. They are either Statistical approaches or linguistic approaches. Statistical approaches are TF-IDF, MMR method, Naive-Bayesian method and other machine learning approaches. Linguistic approaches such as RST, textual entailment, graph-based methods such as LexRank and TextRank algorithms have been used. There are very few methods that have used the combination of both the approaches.

1.3.2 Statistical methods

Statistical approaches copy the information deemed most important by the system to the summary.

- It relies on the statistical distribution of certain features.
- It derives the rank of the sentence based on total weight of the sentence
- The statistical approaches are faster in working, but have an inherent upper bound on performance.
- They are simple to implement, understand and require less time.
- This technique is flexible in the sense that wide range of information can be collected.
- They are standardized; they are relatively free from several types of errors.

1.3.2.1 Limitations of statistical methods

Though statistical methods are faster in working, they have certain limitations:

- Statistical methods do not give a satisfactory result, because the output summary is not very coherent.
- The statistical approach summarizes without understanding the meaning and the relationship between the textual units of the document.
- Summary generated using this technique lacks cohesiveness. Hence the quality of summary is not good.
- It does not identify correlating sentences.

1.3.2.2 Linguistic methods:

The linguistic approach for summarization needs the support of external knowledge-base also known as lexical database. This approach has the following advantages

- It produces meaningful summary.
- Linguistic approaches consider term semantics and yield better results. These techniques can identify relation between sentences and use co-reference resolution techniques

1.3.2.3 Limitations of Linguistic methods

- This approach works faster on small-sized documents but not on large-sized documents
- The linguistic methods do not consider domain-specific features in text summarization whereas these features can be considered only in statistical methods.

In our research study we propose new model that uses combination of statistical and shallow linguistic features in feature extraction, fuzzy logic framework and co-reference resolution. Fuzzy logic is rule-based unsupervised method for generating summary of large documents. The decision module can be designed efficiently using fuzzy rules. Lexical chain is NLP (Natural Language Processing) technique that focuses on semantic analysis of the document. According to our research, fuzzy logic framework with shallow features extraction has proved to be efficient in generating summary as compared to other summarizers that use only statistical methods. Techniques using combination of fuzzy logic and lexical chaining along with co- reference resolution has not come to publish as per authors knowledge in literature survey.

1.3.3 Aim of Thesis

The aim of thesis is to explore and implement an automatic text summarization using knowledge base and combination of various statistical and linguistic features. The proposed system uses basically extraction method and single document summarization using fuzzy logic and semantic analysis. The fuzzy logic helps to extract significant sentence and word features from the given document and decision module determines the degree of importance of each sentence based on its feature scores. The decision making is based on rule-base module of fuzzy system. The degree of importance of sentences determines the summary of the given document. The dataset consists of news articles in particular domain. The work done in this thesis is subdivided into following parts.

1. Study and define the text features for deciding and extracting the relevant sentences for summarization.
2. Design the framework for summarization process.
3. Generate the summary

1.3.4 Problem Statement

"An efficient domain-specific Text summarization technique using Knowledge-base and combined Statistical & Linguistic methods".

1.4 Significance

Most of the current automated text summarization systems use extraction method to produce a summary. Sentence extraction techniques are commonly used to produce extraction summaries. One of the methods to obtain suitable sentences is to assign some numerical measure of a sentence for the summary called sentence scoring and then select the best sentences to form document summary based on the compression rate. In the extraction method compression rate is an important factor used to define the ratio between the length of the summary and the source text. As the compression rate increases, the summary will be larger, and more insignificant content is contained. While the compression rate decreases the summary to be short, more information is lost. In fact, when the compression rate is 5-30%, the quality of summary is acceptable.

Large information can be restated briefly in your own words which give the total meaning of the document in concise format. The total information is reduced or condensed into another version of the same document called as summary. The summary comprises of very important elements and it will have the essence of the original information. Summary will focus on the main ideas that describe the complete document. It also consists of important details that are supporting the main

ideas. The summary should be short in size but should be able to ingest all the main points without fail. This is because summary acts as a mirror for the total document and the reader should be able to scan the main ideas of the whole document in the summary.

Summary is very important to exist in a large document or any lecture as it helps the reader to learn all the aspects discussed in the document. Summary is useful to exist in academic matters. Summarizing helps the student for his academic purposes when he is studying about something. Writing a summary for a set of large information after reading it will help the student to remember well what he has studied. It is necessary to write summaries in the college assignments and in written tests. In most of the academic projects, summarizing will help to shrink the source information into a small paragraph in a research project thesis.

Summarizing is important as it will offer profit to both the teacher and student. For the student summary helps to tell others important points, to test his or her understanding, to gain practice in decision making and arranging of points in a sequence. For the teacher, summary helps to assess the capacity of the student to gather significant points, to identify the comprehensive form of the total information, to look at the ability of the student to organize and prioritize the points.

Summarizing helps us to gather the original text properly which indicates at the end that the reader understood the concept clearly. The knowledge we get by summarizing will help us to analyze and criticize the original document.

1.5 Research Design

We have some work on this field by applying different algorithm and then merge them and produce more optimize result. By this we find out which algorithm produce more optimize result in which state and which type of data or text

Like we apply TextRank, LexRank, Latent semantic and Fuzzy Logic and tried to find out which algorithm are best. Now a day Fuzzy Logic is immerging the best algorithm for text summarization. But we try to do by merging all four algorithm and find the optimize solution.

Now we discuss some pre-written research paper on TextRank summarization. Automatic Text Summarization gained attention as early as the 1950’s. A research paper, published by Hans Peter Luhn in the late 1950s, titled “The automatic creation of literature abstracts”, used features such as word frequency and phrase frequency to extract important sentences from the text for summarization purposes.

Another important research, done by Harold P Edmundson in the late 1960's, used methods like the presence of cue words, words used in the title appearing in the text, and the location of sentences, to extract significant sentences for text summarization. Since then, many important and exciting studies have been published to address the challenge of automatic text summarization.
Text summarization can broadly be divided into two categories — Extractive Summarization and Abstractive Summarization.

1 **Extractive Summarization:** These methods rely on extracting several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary. Therefore, identifying the right sentences for summarization is of utmost importance in an extractive method.
2 **Abstractive Summarization:** These methods use advanced NLP techniques to generate an entirely new summary. Some parts of this summary may not even appear in the original text.

In this article, we will be focusing on the extractive summarization technique

Let's understand the TextRank algorithm, now that we have a grasp on PageRank. I have listed the similarities between these two algorithms below:

- In place of web pages, we use sentences
- Similarity between any two sentences is used as an equivalent to the web page transition probability
- The similarity scores are stored in a square matrix, similar to the matrix M used for PageRank

TextRank is an extractive and unsupervised text summarization technique. Let's take a look at the flow of the TextRank algorithm that we will be following:

- The first step would be to concatenate all the text contained in the articles
- Then split the text into individual sentences
- In the next step, we will find vector representation (word embeddings) for each and every sentence
- Similarities between sentence vectors are then calculated and stored in a matrix
- The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation
- Finally, a certain number of top-ranked sentences form the final summary

1.6 Source of Data

We read some research paper for this. Take some healthy Discussion with my mentors and she gave me some new approaches and help me when I would be stuck. We are also read some pre-existence program and saw which algorithm they use and how they apply this algorithm.

We read different algorithm like fuzzy logic and latent semantic analyses and took a look how they work and then I read some research paper where other apply this algorithm in text summarization.

We read research paper like:

- A text summarization method based on fuzzy rules and applicable to automated assessment
- Abstractive Text Summarization based on Improved Semantic Graph Approach
- Automatic Text Summarization Based on Word-Clusters and Ranking Algorithms
- Abstractive Text Summarization based on Improved Semantic Graph Approach

CHAPTER 2

PROBLEM DOMAIN

2.1: Problem Definition

2.1.1 Media monitoring

The problem of information overload and “content shock” has been widely discussed. Automatic summarization presents an opportunity to condense the continuous torrent of information into smaller pieces of information.

2.1.2 Newsletters

Many weekly newsletters take the form of an introduction followed by a curated selection of relevant articles. Summarization would allow organizations to further enrich newsletters with a stream of summaries (versus a list of links), which can be a particularly convenient format in mobile.

2.1.3 Search marketing and SEO

When evaluating search queries for SEO, it is critical to have a well-rounded understanding of what your competitors are talking about in their content. This has become particularly important since Google updated its algorithm and shifted focus towards topical authority (versus keywords). Multi-document summarization can be a powerful tool to quickly analyze dozens of search results, understand shared themes and skim the most important points.

2.1.4 Internal document workflow

Large companies are constantly producing internal knowledge, which frequently gets stored and under-used in databases as unstructured data. These companies should embrace tools that let them re-use already existing knowledge. Summarization can enable analysts to quickly understand everything the company has already done in a given subject, and quickly assemble reports that incorporate different points of view.

2.1.5 Financial research

Investment banking firms spend large amounts of money acquiring information to drive their decision-making, including automated stock trading. When you are a financial analyst looking at market reports and news every day, you will inevitably hit a wall and won’t be able to read everything. Summarization systems tailored to financial documents like earning reports and financial news can help analysts quickly derive market signals from content.

2.1.6 Legal contract analysis

Related to point 4 (internal document workflow), more specific summarization systems could be developed to analyze legal documents. In this case, a summarizer might add value by condensing a contract to the riskier clauses, or help you compare agreements.

2.1.7 Social media marketing

Companies producing long-form content, like whitepapers, e-books and blogs, might be able leverage summarization to break down this content and make it sharable on social media sites like Twitter or Facebook. This would allow companies to further re-use existing content.

2.1.8 Question answering and bots

Personal assistants are taking over the workplace and the smart home. However, most assistants are fairly limited to very specific tasks. Large-scale summarization could become a powerful question answering technique. By collecting the most relevant documents for a particular question, a summarizer could assemble a cohesive answer in the form of a multi- document summary.

2.1.9 Video scripting

Video is becoming one of the most important marketing mediums. Besides video-focused platforms like YouTube or Vimeo, people are now sharing videos on professional networks like LinkedIn. Depending on the type of video, more or less scripting might be required. Summarization can get to be an ally when looking to produce a script that incorporates research from many sources.

2.1.10 Medicases

With the growth of tele-health, there is a growing need to better manage medical cases, which are now fully digital. As telemedicine networks promise a more accessible and open healthcare system, technology has to make the process scalable. Summarization can be a crucial component in the tele-health supply chain when it comes to analyzing medical cases and routing these to the appropriate health professional.

2.2 Experimental Set-up

This project focuses on the Fuzzy logic extraction approach for text summarization and the graph-based approaches TextRank and LexRank for extracting the keyword using structure of the article and the semantic approach of text summarization using Latent Semantic Analysis. Our hybrid model consists of four components: TextRank, LexRank, LSA,

Fuzzy Logic. In this model, each text summarization method is used to extract the keywords, each method has ranked the keywords based on their importance (recurrence, centrality, noun etc). These keywords with their scores are then given as input to the final keyword extractor. In this phase, the keywords occurring in the final result of all the four methods are taken into final keyword list. Then the remaining keywords of all the methods are arranged in the descending order of the scores. From this list the top m keywords are selected in the final output. Keyword extraction can be done to select as many keywords we want, but, usually 1% of the total keywords (except stop words) are enough to represent the central idea of the document. Initially the document is passed through a pre-processing phase, to get only the required data. Like in the previous line, the keywords like is, the, a, to, only are not going to be the keywords representing the idea of the document. So, we first remove the unwanted stuff from the document and then also add some words which semantically mean the same, to get better semantic understanding. For example, the keywords like improving, enhancing are typically mean the same, so we can replace one with the other, so the actual occurrence of the word could be identified.

2.3 Procedures Adopted

Each of the four techniques were first implemented separately, then their output is given as the input to the algorithm to extract more effective keywords, for determining the central idea of the document. Following figure shows the detailed flow chart of the proposed method containing step-wise procedure of all the four methods.

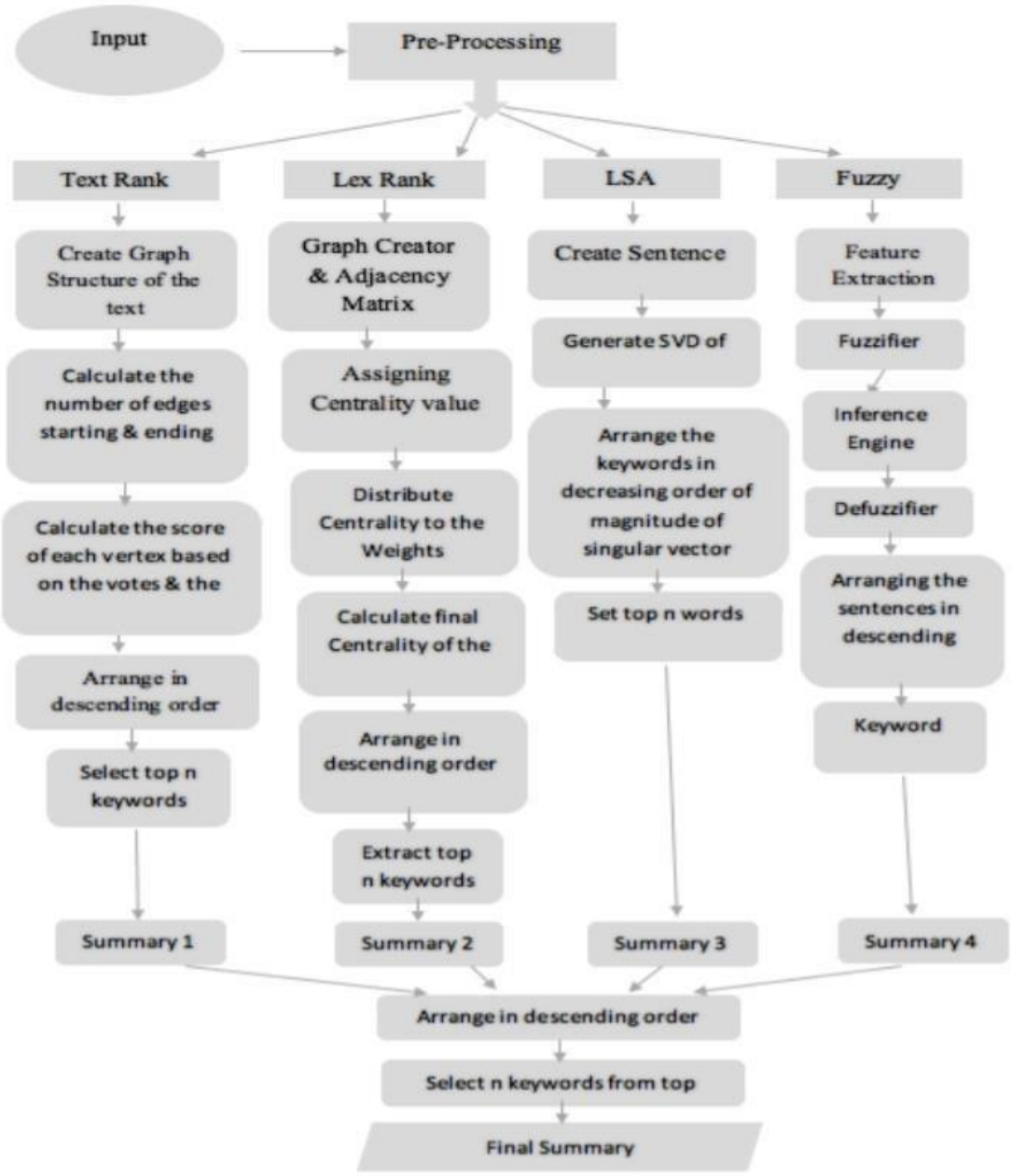


Fig 2.3 Proposed Algorithm

CHAPTER 3

LITERATURE SURVEY

3.1 Overview: A text summarization method based on fuzzy rules and applicable to automated assessment

3.1.1 Introduction

In this paper, an automated method for text assessment that starts with text summarization and

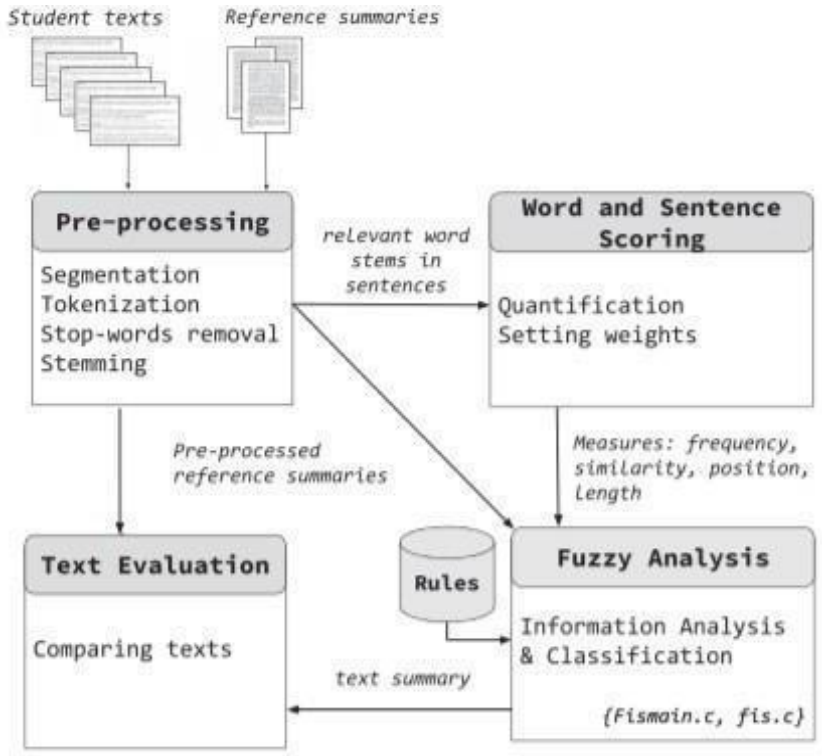


Fig 3.1.1 Text Summarization using fuzzy rules

Fig. 2.3, Text Summarization using fuzzy rules then compares the automatically generated summaries with reference texts provided by domain experts. The proposed summarization method is a fuzzy rule-based system that identifies and selects the most informative sentences and concepts in each text document. The proposed method can combine features extracted from the text by using a multivariate linear regression model. It reduces the number of fuzzy rules without performance degradation.

3.1.2 Methods

Proposed method employs measures founded on fuzzy logics for ATS, including concrete or objective attributes (features observable in a text) and metrics for more abstract, higher-level, or somewhat subjective attributes (fuzzy metric). The method is composed of four major tasks executed in subsequent stages:

- **Pre-processing:** The pre-processing module includes: the identification of sentences segmentation; breaking the continuous stream of characters in words or terms, also called tokens tokenization; the elimination of words that do not add relevant information such as, articles, prepositions, adverbs, numbers, pronouns and punctuation stop word removal; and a linguistic normalization in order to obtain the radical of each word stemming. In English, for example, the tokens: lies, lying and lie can be transformed to the stem lie after removing the prefix and suffix. The pre-processing is an important step, as it reduces the amount of information to analyze and can contribute for the next steps to generate better results.
- **Word and sentence scoring:** The word and sentence scoring module extracts the features that express the relevance of the words and sentences - quantification; and assigns weights based on measures - setting weights. The text goes through several steps in order to prepare it for the next module, the fuzzy analysis. o Measures The first step in the word and sentence scoring module is to identify the textual features that can be taken into account when determining the importance of sentences. The scoring technique uses statistical or empirical methods for the summary considering textual features such as the position and length of the sentence, word frequency, and the distance between sentences. Through the analysis of textual features, it is possible to derive measures that recalls the relevance of a sentence. A = {frequency, similarity, position, length}. Finally, all measures are normalized between 0 and 1 for improving the fuzzy analysis process and be treated with equal proportion.
- **Fuzzy analysis:** The fuzzy module does the informativeness classification. We can say that the dataset itself is its own most informative description, and any other summary implies a loss of information. Thus, informativeness is a measure that represent to what extent a particular summary is informative. After all measures are computed, the scores are used to model words through the fuzzy set theory. Formally, a fuzzy set on X, denoted by A, is defined as $A = \{x, \mu A(x) | x \in X\}$, where $\mu A(x)$ is the membership function of x.

In the fuzzy module the rule antecedents are concatenated by and connectives that involve the operator MIN and all outputs of the rules are grouped by the MAX method. We produced 27 rules (by the combination all input variables) in the rule base and the fuzzy sets were represented with the bell membership function. The parameter values of the bell function were predicted with manual experimentation and a priori knowledge. The reason we choose the bell membership function is because the number of fuzzy rules is small and the value of the fuzzy output score of sentences is unrepeatable. Note that the fuzzy module output is a metric over the space of texts

that produces values in the range [0, 1]. There is an equivalence between the triangle inequality and fuzzy points to measure the distances.

3.1.3 Features

- The fuzzy summarization not only improves informativeness of the summary but also helps to identify the concepts that are suitable to guide study by teachers and students
- The fuzzy summarization gives more precise summary with less training dataset

3.1.4 Limitations

- In fuzzy summarization the quantity of measures used to identify the main concepts can be a problem since the greater the number of measures, the higher the number of fuzzy rules to express knowledge.
- The overall performance of our method depends crucially on the efficiency of the measures and the text evaluation. In the text evaluation unigrams, answers containing few words provide more accurate results than those with many words. Therefore, to improve the performance of our method, it is necessary to deepen the analysis with semantic verification of the terms.

3.2 SummCoder: An Unsupervised Framework for Extractive Text Summarization Based on Deep Auto-encoders

3.2.1 Overview

Automatic text summarization is an important branch of natural language processing that aims to represent long text documents in a compressed way so that the information can be quickly understood and readable for end users. We can broadly classify text summarization techniques into two categories; extractive and abstractive text summarization. Extractive text summarization concatenates the most relevant sentences from the document to produce the summary. It usually comprises of three major steps: intermediate representation of input text, sentence scoring, and sentence selection. Conversely, abstractive text summarization generates the summary by paraphrasing the main contents of the document using natural language generation techniques. The summarization can also be categorized as single document and multi-document depending on the number of input documents given. Similarly, the summarization can be either generic or query-based. The generic summarization provides an overall idea of the document content whereas, query-focused summarization presents the relevant content of the document according to the user

given queries. In this paper, the method is based on the task of generic extractive text summarization on single documents. At present, the biggest challenge in applying supervised deep learning approaches for extractive text summarization is the unavailability of large-scale extractive summaries created manually that are needed as ground truth for training the networks. This paper addresses this shortcoming by exploiting techniques, which do not require labeled data for training, specifically an unsupervised deep learning approach based on autoencoders and sentence embedings.

3.2.2 Methods

This method is comprised of following steps like:

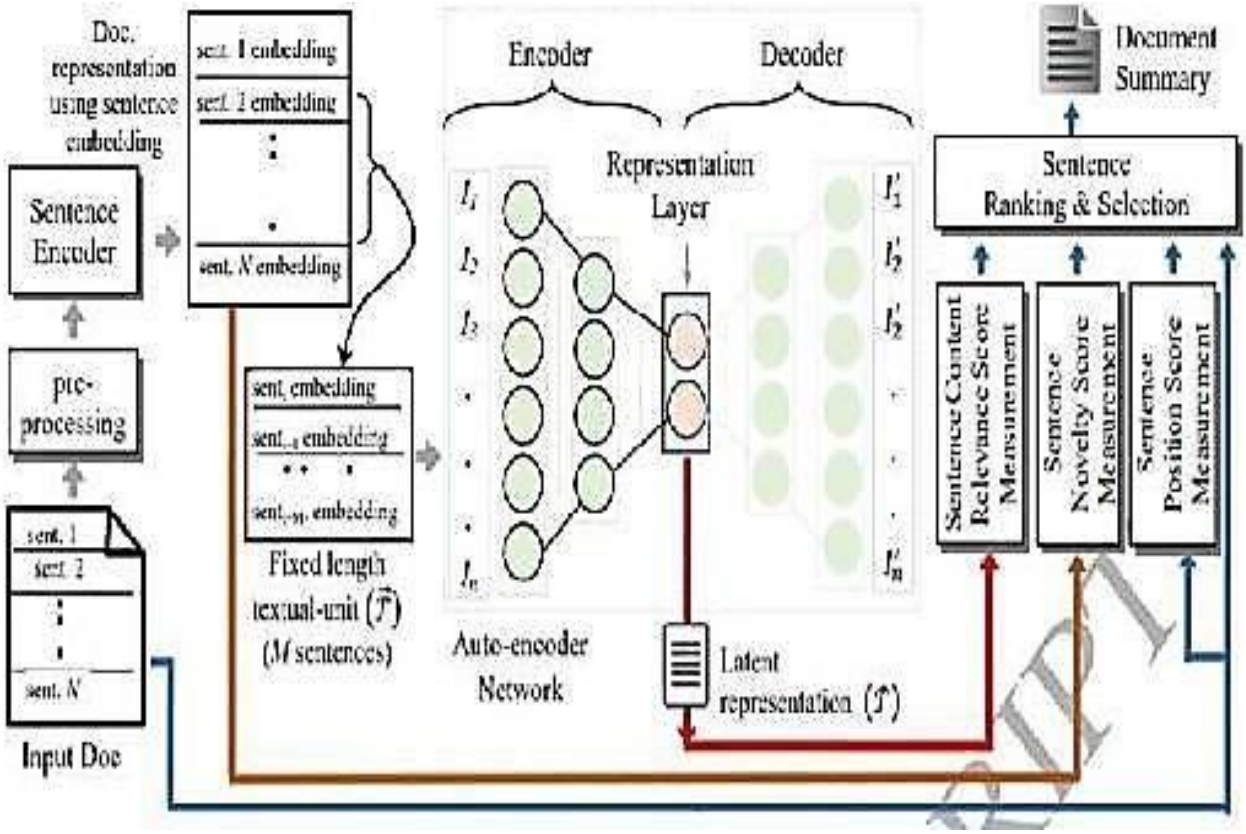


Fig 3.2 SummCoder: Text Summarizer based Deep Auto- encoders

- **Preprocessing:** Firstly, we pre-processed each document before passing it to the next stage of the sentence encoder. We extracted text from the documents by removing any XML/HTML tags and metadata and then tokenized the text into sentences. We cleaned the document by removing special characters, redundant whitespaces, numbers, URLs and email addresses. All the words were converted to lower case and sentences containing less

than five words were omitted while training the network. For pre-processing, we used Natural Language ToolKit5 (NLTK) library and regular expressions.

- **Sentence Encoder (Vector Representaion):** After pre-processing of the input document, the next step is to map each sentence into a fixed length vector of real numbers, and we used skip-thoughts model to obtain sentence embeddings. It is an unsupervised approach (the model is termed as "skip-thoughts") to train a generic, distributed sentence encoder which models sentences into vectors called skip-thought vectors.

The skip-thoughts model is inspired by word vectors learning, abstracted to the sentence level in the sense that sentence vectors are learned using surrounding sentences. It considers that nearby sentences provide rich semantic and contextual information. The model is based on encoder-decoder networks such as an RNN encoder with Gated Recurrent Units (GRUs) activations and an RNN decoder with conditional GRUs. The purpose of an encoder network is to map an English sentence into a vector and then the decoder part tries to map encoded vector into surrounding sentences. The encoder-decoder architecture pushes the model to select and abstract some important features and creates the connection between a source and a target. The complete network is trained to reconstruct the surrounding sentences to map sentences, which have semantic and syntactic similarity into identical vector representations. The unidirectional skip- thoughts model has been trained on unlabeled Book Corpus dataset and achieved state-of-the-art accuracies on various tasks such as semantic relatedness, paraphrase detection, image-sentence ranking, or classification.

- **Summary Generator:** The extractive summarization problem can be defined as a sentence sequence ranking or sentence selection problem in a document. To determine which sentences should be included in the summary, we propose three criteria:
- **Sentence Content Relevance Metric (scoreConstR):** To measure the relevance of a sentence in the document based on its content, we first compute a latent document representation. Next, we remove the current sentence from the input document and generate another latent representation with remaining sentences in the document, which we will refer to as modified latent representation. More precisely, to generate the modified latent representation we replace the embedding vector of the considered sentence from the input textual-unit embedding with a vector of the same size containing all zeroes and fed it to auto-encoder to obtain the modified latent document representation. To estimate the sentence content relevance score, for each sentence we generate the modified latent

document representation for all the sentences, based on the aforementioned process and then compute the cosine similarity between the original latent document representation and each modified latent representation. The value of score is bounded in [0,1] and a higher value represents a more relevant sentence. It is reasonable to assume that, in any document, important sentences contain a higher number of important words or phrases compared to less important sentences.

- **Sentence Novelty Metric (scoreNov):** This parameter computes the novelty metric for each sentence in a document. The aim is to assign a low score when the sentence is redundant or repetitive and high score when it is novel. We obtain the similarity between two sentences and as the cosine similarity of their corresponding embedding vectors. To obtain the global novelty of a sentence in the document, we compute its similarity with remaining sentences and if the similarity is found below a pre-determined threshold, then it is considered novel. Moreover, with this strategy, when two sentences are found similar to each other beyond the predetermined threshold (0.85), we assign a higher novelty score to the sentence, which has a higher value for content relevance measure scoreConstR. The value of scoreNov parameter is in the range [0, 1].
- **Sentence Position Relevance Metric(scorePosR):** The sentence position is an early sentence-based heuristic for text summarization. We devised sentence position relevance metric, scorePosR as another important parameter which contributes to the overall sentence ranking in the document. The value of the proposed parameter scorePosR is bounded in the range [0.5, 1.0]. The sentence position relevance score is higher for sentences located at the beginning of the document and it gets, stable at a value of 0.5 after a given number of sentences depending on the total number of sentences of the document N.
- **Sentence Ranking and Selection:** We define the final sentence score of a sentence S1 in a document D, scoref(D,S1) , using weighted score fusion of the three criteria, i.e. sentence content relevance score , sentence novelty score and sentence position relevance score as follows: scoref(D,S1) = a*scoreContR(D,S1) + b*scoreNov(D,S1) + c*scorePosR(S,S1) where s,b,c belongs to [0,1] with a+b+c=1 , are fusion weights and assigns relative weights to different parameters while calculating the final sentence selection score.

3.2.3 Features

- The selection of sentence is pretty straight forward
- The auto encoder is trained though unsupervised learning method means we don’t need

- Large labeled datasets

3.2.4 Limitations

- This method is not so intelligent; in the future we would like to improve summarization by considering phrases as a smaller unit than a sentence as taken in current work, looking this way to improve the summarization.
- It can be extended for query-based text summarization and multi-document text summarization.

3.2.5 Abstractive Text Summarization based on Improved Semantic Graph Approach

The goal of abstractive summarization of multi-documents is to automatically produce a condensed version of the document text and maintain the significant information. Most of the graph-based extractive methods represent sentence as bag of words and utilize content similarity measure, which might fail to detect semantically equivalent redundant sentences. On other hand, graph based abstractive method depends on domain expert to build a semantic graph from manually created ontology, which requires time and effort. This work presents a semantic graph approach with improved ranking algorithm for abstractive summarization of multi-documents. The semantic graph is built from the source documents in a manner that the graph nodes denote the predicate argument structures (PASs)—the semantic structure of sentence, which is automatically identified by using semantic role labeling; while graph edges represent similarity weight, which is computed from PASs semantic similarity. In order to reflect the impact of both document and document set on PASs, the edge of semantic graph is further augmented with PAS- to-document and PAS-to-document set relationships. The important graph nodes (PASs) are ranked using the improved graph ranking algorithm. The redundant PASs are reduced by using maximal marginal relevance for re-ranking the PASs and finally summary sentences are generated from the top ranked PASs using language generation.

3.3.6 Method

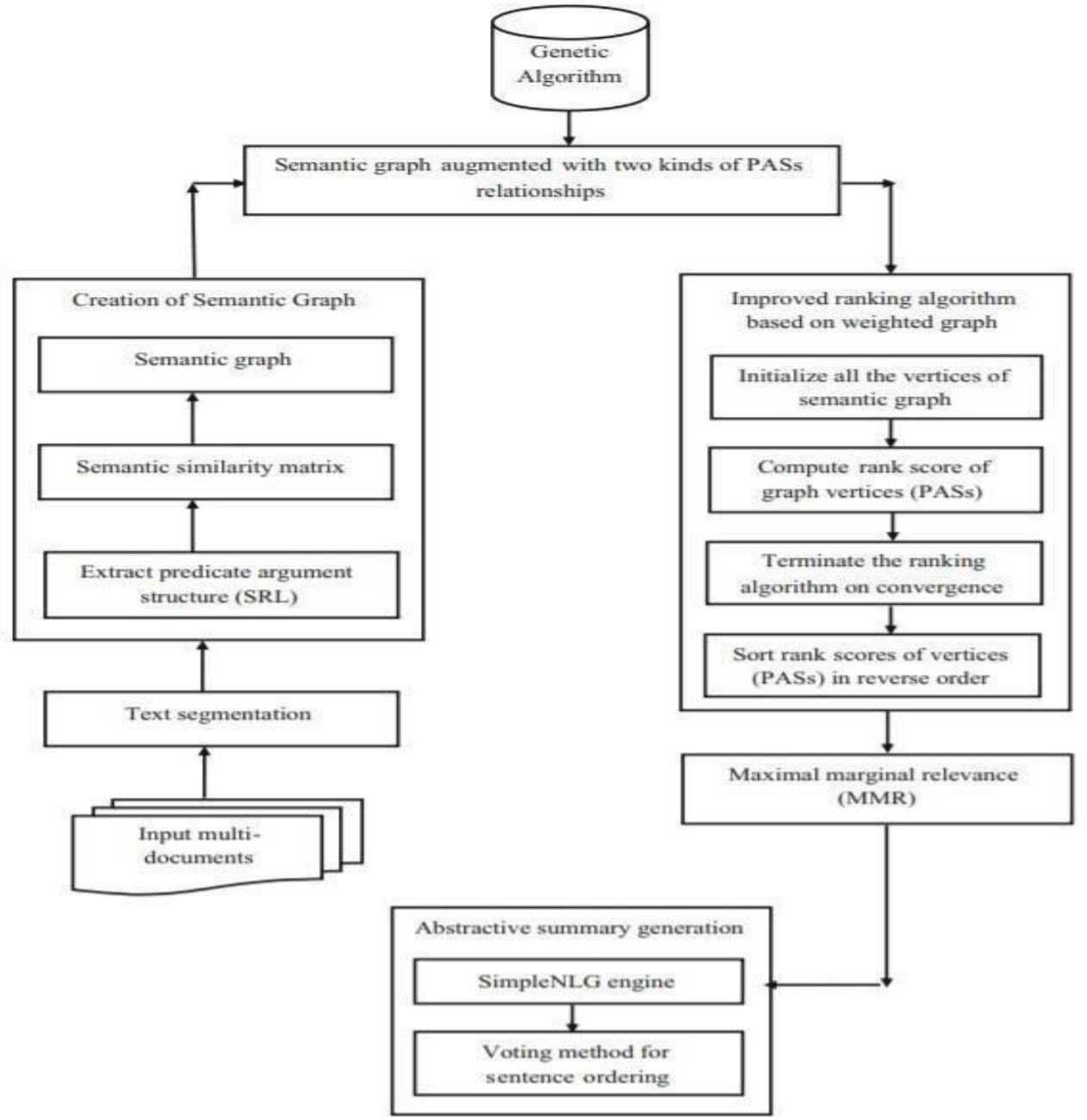


Fig 3.2.5 Abstractive Text Summarization Based on Improved Semantic Graph Approach

Creation of Semantic Graph: This phase aims to build semantic graph from the document collection. The steps involved in this phase are as follows:

- **Semantic Role Labelling:** The goal of this step is to parse each sentence in the document collection and extracts PAS from them. We begin by segmenting the document collection into sentences. Each sentence is given a key based on its document number and location

number. As profound semantic text analysis is carried out in abstractive summarization, therefore, we employ SENNA semantic role parser to determine PAS from the sentence collection by labeling the semantic phrases appropriately. Example 1 Consider the following two sentences represented by simple predicate argument structures. S1: "Eventually, a huge cyclone hit the entrance of my house". S2: "Finally, a massive hurricane attacks my home". After applying semantic role labeling to sentences S1 and S2, the corresponding simple predicate argument structures P1 and P2 are extracted are as follows: P1: [AM-TMP: Eventually] [A0: a huge cyclone] [V: hit] [A1: the entrance of my house] P2: [AM-DIS: Finally] [A0: a massive hurricane] [V: attack] [A1: my home] When we achieve PASs, they are segmented into significant tokens/words, then stop words are taken out.

- **Semantic Similarity Matrix:** This step intends to calculate pair-wise semantic similarity scores PASs and then build a matrix from it. The verb, noun, time and location arguments of each PAS are compared with other PASs to compute pair wise similarities. Empirical findings revealed that Jiang similarity measure was found to have close agreement with humans as compare to other similarity measures. Consequently, this research make use of Jiang's measure to find semantic similarity between PASs. This measure supposes that each concept that is present in the WordNet own some information.
- **Semantic Graph:** A semantic graph (an undirected weighted graph) is created from the similarity matrix in a manner that if similarity weight sim (pi, pj) between PASs (vertices) pi and p j is greater than zero, then link is set up between PASs; otherwise a link is not created. So the link or edges of graph represent similarity weight between PASs. This study defines a semantic similarity threshold i.e. $0 < \beta \leq 0.5$ based on empirical analysis. Consequently, a link is created between PASs (vertices), if semantic similarity is within the range; otherwise no link is set up if similarity is outside the range.
- **Semantic Graph Analysis:** Augmentation with PAS Relationships We assume that the PASs which are more related with the document/document set, are more reasonable to be included in summary. Based on text summarization literature, we employ text features that are more relevant to represent predicate argument structure to document relationship, which includes: title feature, location and PAS to document semantic similarity. Similarly, we also use text features that are more appropriate to signify the correlation/relationship of predicate argument structure to document set, and are given as follows: term weight and frequent term and PAS to document set semantic similarity. Summary quality is vulnerable

to text features i.e. different feature have not same importance in the process of producing summary. Therefore, assigning weights to features is essential for creating a good summary. This study makes use of genetic algorithm (GA) to get optimal weights for different features. GA is selected since it is an optimization technique which is robust, and employed in diverse disciplines of applications and research, especially to deal with the problems of optimization.]. GA is trained and tested on 59 data sets (fetched from DUC 2002), which are chosen by exploiting 10-fold cross validation. The GA procedure starts by setting the initial population to 50 chromosomes, which are assigned with the random real values in the range of 0 and 1. The fitness function indicates the average recall achieved from summaries of multiple documents. The fitness function computes the fitness values of chromosomes, which indicates how the good the chromosome is. When all the training multi-documents are summarized, then in this study, the best average recall will serve as a fitness function. In this study, we implemented scattered method to execute the crossover operation, and Gaussian mutation method to accomplish the mutation operation. The GA process is usually converged when 100 maximum generations are reached and is therefore terminated. The individual chromosome with maximum fitness value represents optimal weights for features. • Improved Weighted Graph-Based Ranking Algorithm Conventionally, Google's PageRank and HITS algorithm are ranking algorithms that have been effectively employed in link analysis and social networks. PageRank algorithm attempted on undirected graph gave the best performance in DUC 2002 extractive summarization task for multi-documents The PageRank is a ranking algorithm that offers a way for determining the significance of a vertex in a graph by assuming the global information from the whole graph. In other words, all sentences are rated without taking into consideration the sentence-to-document relationship and sentence-to-document set relationship. This work utilizes an improved ranking algorithm based on weighted graph (IWGRA). The ranking algorithm will consider edge weights in the importance analysis of vertices (PASs). The edge weight is computed from PAS-to-PAS semantic similarity, PAS-to-document and PASto- document set relationships.

Algorithm 1 Improved weighted graph-based ranking algorithm

Step	Main process	Process detail
1	*Initialization*	Initialize the rank scores of all vertices of graph to 1. Set the damping factor to 0.85.
2	*Computation of rank score*	*2.1* Retrieve the vertices of the graph.
		2.2 For each vertex of the graph, determine the vertices that are linked to it.
		2.3 For each linked vertex, determine its importance by finding the number of outgoing links from that vertex, and then sum up the weights associated with the outgoing links.
		2.4 Compute the rank score of the given vertex considered in Step 2.2, using Eq. (18).
		2.5 Update the rank score of the vertex under consideration.
		2.6 Repeat Steps 2.2-2.5 until convergence is attained. Note convergence is attained when the difference between any two successive vertices scores falls below the given threshold (0.0001).
3	*Sort the rank scores of graph vertices*	Sort the rank scores obtained for the graph vertices in reverse order.

Fig. 3.2.5 Abstractive Text Summarization Based on Improved Semantic Graph Approach

- **MMR to Control Redundancy:** The ranking algorithm may assign same rank score to PASs representing the same concept and hence the final summary may contain redundancy. Furthermore, the other concepts of the documents which represents group of least similar PASs may not be included in the final summary, and thus significant information may lost. In this study, we employ a modified version of MMR to reduce redundancy by reranking the PASs for inclusion in summary. A predicate argument structure is included in the summary generation list if it is not too similar to any existing PAS in the summary generation list. At first, the PAS with the maximum salience score is chosen and appended into the summary generation list. Then, the subsequent PAS having the higher salience score is selected and added into summary generation list. This process chooses PASs by taking into account both importance and redundancy and keeps on repeating until the compression rate of summary is met. . Once the graph vertices (predicate argument structures) are re-ranked using MMR, then the top scored vertices (PASs) are chosen based

on compression rate of summary and are given to next phase that employs a Simple NLG engine.

- **Summary Generation:** In this phase, we make use of Simple NLG along with simple heuristic rules defined in it, to produce summary sentences from the top scored representative PASs chosen in previous phase. Simple NLG engine utilizes simple English grammar rules to convert syntactical structures into sentences. The engine is robust i.e. it is not crashed, when partial syntactical structures are provided as input. The first heuristic rule states that "if the subjects in the predicate argument structures (PASs) refer to the same entity, then merge the predicate argument structures by removing the subject in all PASs except the first one, separating them by a comma (if there exist more than two PASs) and then combine them using connective " and The second rule states that "If PAS Pi subsumes a PAS Pj , then the subsumed PAS Pj is discarded in order to avoid redundancy". the summary sentences created from PASs will be the compressed version of the original sentences in document collection. The heuristic rule defined in Simple NLG merges the PASs that indicate the same subject. For instance, the following source input sentences: S1: "Hurricane Gilbert claimed to be the most intense storm on record in terms of barometric pressure". S2: "Hurricane Gilbert slammed into Kingston on Monday with torrential rains and 115 mph winds". S3: "Hurricane Gilbert ripped roofs off homes and buildings" After applying SENNA SRL: the corresponding three predicate argument structures P1, P2 and P3 are obtained as follows: P1: [A0: Hurricane Gilbert] [V: claimed] [A1: to be the most intense storm on record] P2: [A0: Hurricane Gilbert] [V: slammed] [A1: into Kingston] [AM-TMP: on Monday] P3: [A0: Hurricane Gilbert] [V: ripped] [A1: roofs off homes and buildings]

From the previous step, P1, P2 and P3are considered to be the top ranked PASs. After Applying the rule1 on the above example, we found that subject A0 is repeated and is ignored from all PASs apart from the first one. The Simple NLG engine utilizes first heuristic rule on the three PASs given in aforementioned example, and creates the summary sentence, which is the condensed version of the original sentence in the document. Summary Sentence: "Hurricane Gilbert claimed to be the most intense storm on record, slammed into Kingston on Monday with torrential rains and ripped roofs off homes and buildings". After the generation of summary sentences, then we propose a voting scheme to rearrange the summary sentences. The scheme operates in four steps:

- **Step 1:** The voting scheme is based on the idea of voting or recommendation i.e. one sentence vote another sentence, if it addresses the same concept as described in another sentence, which is determined based on different similarity relations between them. In this work, a well know similarity measure, Jaccard similarity measure [54] is employed to calculate the similarity between a summary sentence and source sentences in document set.
- **Step 2:** Once the similar source sentences (contained in the document set) to the corresponding summary sentence are identified, then the positions of the source sentences are averaged to give score to the corresponding summary sentence. We exploit position feature to determine the normalize position [0,1] of a source sentence in the document set
- **Step 3:** Repeat step 1 and step 2 until scores are assigned to all summary sentences.
- **Step 4:** Arrange the sentences in the summary according to ascending order of scores.

Even though abstractive summarization in real sense is hard to achieve, our semantic graph approach demonstrates the viability of this new track for summarization community.
Our approach improves graph ranking algorithm by incorporating PASs semantic similarity, and the two kinds of PAS relationships.
Experimental findings confirm that summary results are improved with the ranking algorithm.

3.3 Optimizing Text Summarization Based on Fuzzy Logic

3.3.1 Overview

In order to implement text summarization based on fuzzy logic, MATLAB is used, since it is possible to simulate fuzzy logic in this software. To do so; first, we consider each characteristic of a text such as sentence length, similarity to little, similarity to key word and etc., as the input of fuzzy system. Then, we enter all the rules needed for summarization, in the knowledge base of this system (All those rules are formulated by several expends in this field available in MATLAB). Afterward, a value from zero to one is obtained for each sentence in the output based on sentence characteristics and the available rules in the knowledgebase. The obtained value in the output determines the degree of the importance of the sentence in the final summary.

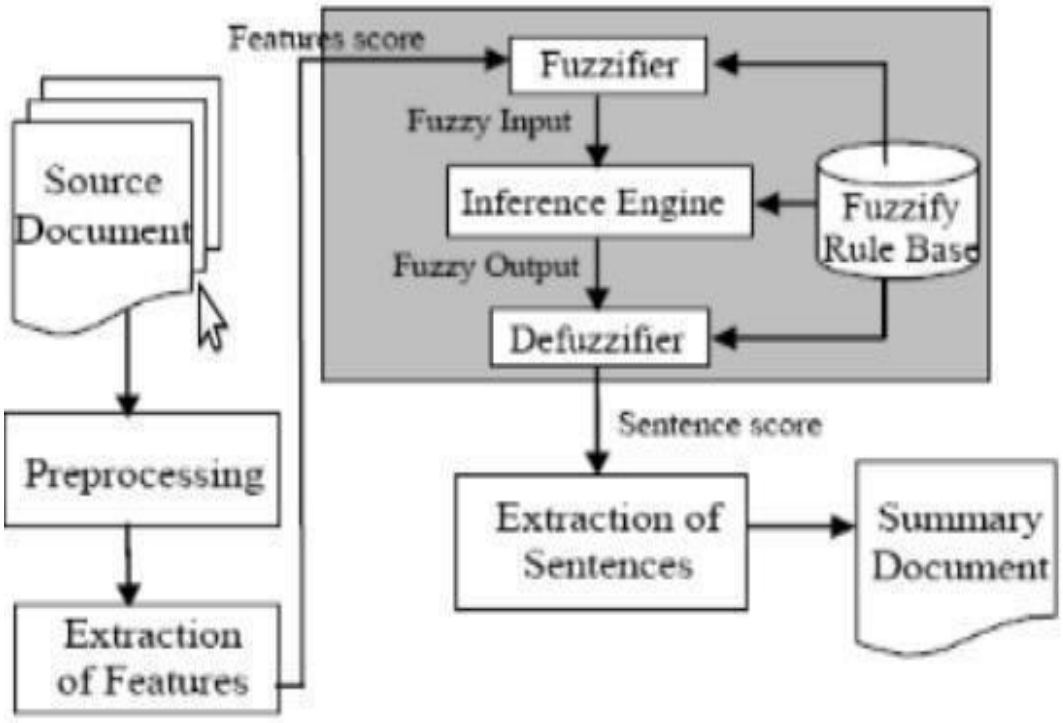


Fig.3.3.1 Optimizing Text Summarization Based on Fuzzy Logic

Based on above, A paragraph was summarized using a vectorial method and compared with the above used fuzzy method. We gave these texts and the summaries produced by both vectorial and fuzzy methods to 5 judges who had an M.A. in TEFL (teaching English as a foreign language). We asked the judge to read the main texts and to score the summaries.

Judges / Methods	The First judge	The second judge	The third judge	The four judge	The five judge
Score of vector method	%65	%68	%72	%66	%64
Score of fuzzy method	%75	%80	%79	%78	%75

Table 3.3.1 Judgment On Comparison Of fuzzy and Vector Method

This indicates that fuzzy method worked better in parts of the sentence which contained uncertainty due to the use of fuzzy quantities. Therefore, by using fuzzy approach in text summarization, we can improve the effect of available quantities for choosing sentences used in the final summaries. In Other word, we can make the summaries more intelligent. 2.4.2

3.3.2 Features

Knowledge-driven reasoning based, can take better results if integrated with data driven technique. • Works better in parts of the sentence which contained uncertainty due to the use of fuzzy quantities.

3.3.3 Limitations

Human experts are needed to define the fuzzy rules. • Overhead in designing the membership function.

3.4 Graph-Based Text Summarization Using Modified TextRank

3.4.1 Method

In the proposed method, we have considered that the sentences of the document are equivalent to Web pages in the PageRank system. The probability of going from sentence A to sentence B is equal to the similarity between the pair of sentences. The modified TextRank proposed in this paper uses the intuition behind the PageRank algorithm to rank sentences using which we can select the most important sentences from the input text document.

In PageRank, important Web pages are linked with other important Web pages. Similarly, in our approach we assume that the important sentences are linked (similar) to other important sentences of the input document. In this model, at first to identify the sentences, we have transferred the input document D into sentences {s1, s2, … , sn} using NLTK package, i.e., D ={ s1, s2, … , sn}.

We have assigned index for each of the sentence according to the input sequence of the sentences in the document. Index values are significantly used for proper ordering of the summarized sentences to make the summary meaningful. In the next step, each sentence is tokenized into a set of words.

To define similarity among the sentences, we represent each sentence of the given text document as a word vector. Next, a complete weighted graph G =(V, E, W) of the input document is built, where V is the set of vertices, each of which corresponds to a sentence represented by the word vector. E is the set of edges between every pair of vertices. W is the set of weights assign to the edges of the graph G. The weight w associated to edge (u, v) ∈ E is assigned using following logic: A] Suppose S(u)={w1 u, w2 u, … , ws u}is the sentence corresponds to the vertex u and S(v)={w1 v, w2 v, … , wt v} is the sentence corresponds to the vertex v.

Here, we have considered term frequency (tf) and inverse sentence frequency (is f) for each word in the sentence. For an example, tfw(S(u)) is the term frequency of word w in S(u) which gives the

number of occurrences of the word w in the sentence. Similarly, is fw(D) is the inverse sentence frequency of the word w in the input document D which is defined by Eq.(1).

$$isf(w, D) = \log \frac{\|D\|}{\|\{S \in D : w \in S\}\|} \quad (1)$$

Now, the is f-modified-cosine similarity issued to measure the similarity between every pair of sentences and it is used as weight w of edge (u, v) using Eq. (2).

Traditional cosine similarity only takes care of the term frequency of the corresponding words in the text document, and in case of cosine similarity, the dimension should be same for all sentence vectors, whereas isf-modified-cosine similarity takes care of the different level of importance for the corresponding words in the sentences and also considers different length of the sentence in the document.

Thus, we get a complete weighted graph which is made sparse by removing the edges having weight less than the threshold (t), set as the average weight of all the edges in the graph. This graph represents the similarity graph of the input document. For weight assignment, score is assigned to every sentence corresponding to every node of the graph G. In this proposed method, we initialize the TextRank score of each node of the graph by the average weight of the edges incident to it for giving importance to the weights associated with the edges(E), whereas in traditional PageRank the initial PageRank value of each node is set to 1 T , where T is the total number of nodes in the graph. Next, the TextRank score is updated via modified TextRank as defined in Eq (3).

$$TR(S(u)) = \frac{d}{T} + (1 - d) * \Sigma_{v \in adj(u)} \frac{TR(S(v))}{deg(S(v))} \quad (3)$$

where TR(S(u)) is the text rank of sentence S corresponds to the node u, TR(S(v)) is the text rank of the sentence S corresponds to the node v such as v ∈ adj(u),deg(S(v)) is the degree of the node v corresponding to the sentence S, T is the total number of nodes present in the graph, and d is a "damping factor." Here, we have considered the value of d as 0.15. Finally, for summarization, we have selected top n scored sentences and rearranged those n sentences according to the sentence index which we have assigned at first step. The n output sentences construct the summary of our proposed model.

3.4.2 Features

It can get better view of important sentences by constructing the similarity graph of sentences using isf-modified-cosine similarity in comparison to the centroid approach.

Algorithm make use of more of the information from the graph representation of the input document and gets better results in many cases.

3.4.3 Limitation

- It does not take care of anaphora resolution problem. We can include it to obtain more informative summary.
- It may be possible that more than one similar type of sentences with high score is selected for the summary. To eliminate this kind of duplication, we may consider any kind of clustering algorithm and merge it with our proposed graph-based algorithm.
- The weight of the graph is considered only in the initialization of the TextRank score for every sentence corresponding to the vertices of the graph.

3.4.4 Improvement of query-based text summarization using word sense Disambiguation

3.4.4.1 Overview

In query-based text summarization, we find the summarized text according to query. The inputs are the query and text documents, and the output is the summarized text. Summarization picks those sentences which are semantically related with the query sentence. Semantic relatedness score is calculated between a query sentence and an input text sentence. To find the score of each input text sentence, semantic relatedness score is computed for each sentence with the query. Content word always carry important information in a sentence. We consider only the content words present in the query and input text sentences. While picking sentences for summary, there is a high probability that same information might be conveyed by multiple sentences. Redundant sentence elimination is an important task in text summarization to reduce the size of summary. Before finding the semantic relatedness score between query and input text sentence, extract the important sentences from input text document. In fact, extracting important sentences helps in finding rich information for summary. Important sentences are those sentences which carry more significant information.

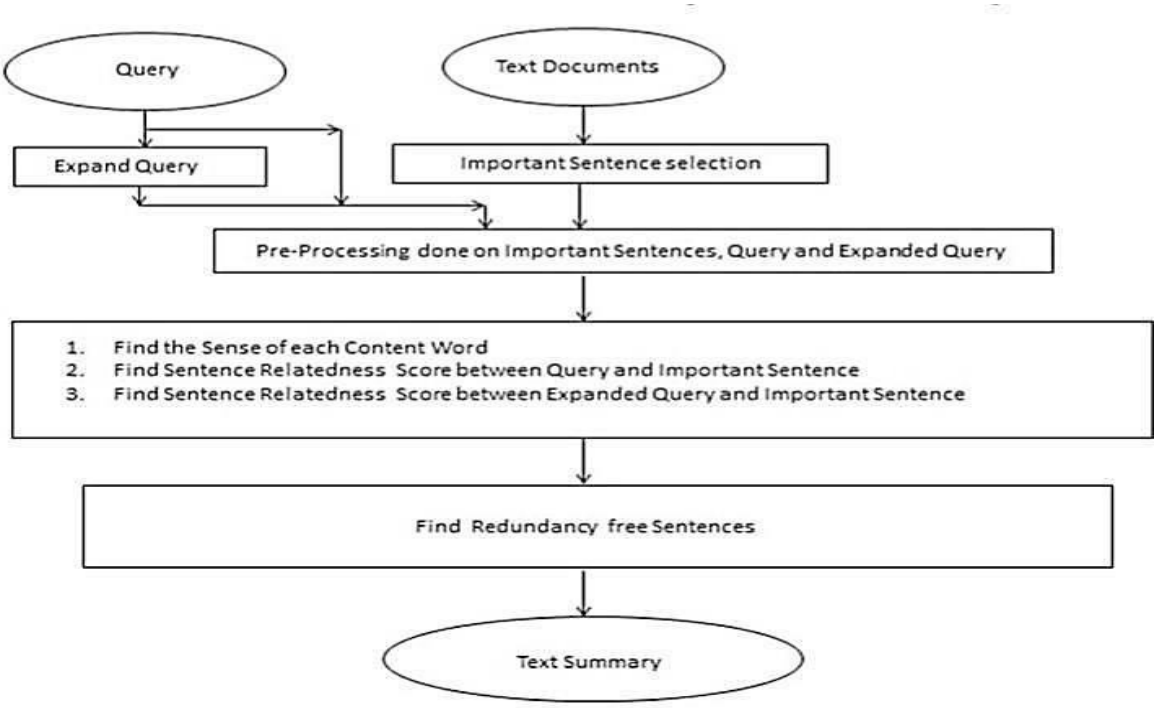


Fig.3.4.4 Flow Diagram Of query based text summarization

Now score of a word sense is calculated based on frequency of occurrence of common words among each target sense definition and the definition of all other content words. Finally, we take the highest score sense as an appropriate one. We use Eq. 1 to find the score of a sense:

$$Score\ W_{sense} = \log\left(1 + \frac{n(c_t \in S, w_d \in w)}{n(t)}\right)$$

where (ct) = Definition of all the content words present in the sentence S, (wd) = Definition of the particular sense of the word w, n (ct, wd) =Number of common words between the definition of all content words present in the sentence S and the definition of the particular sense of the word (wd), n(t) = Total number of unique words in the definition of all content words present in the sentence S and the definition of the particular sense of the word (wd). Redundancy removal is important while preparing the summary. Redundancy-free text summary decreases the summary size and helps in getting different information related to query. To remove redundancy, the method finds the common words in both the sentences. The method uses following Eq. 2 to find similarity between two sentences using Jaccard similarity coefficient:

Similarity (S1, S2) = (S1 ∩ S2) /(S1 ∪ S2) (2)

Here, S1∩S2 =CommonwordsbetweenS1 andS2 sentences. S1 ∪ S2 = Total unique words in S1 and S2 sentences. Higher similarity score signifies similarity between two sentences are more.

Extraction of sentences can be done on the basis of lowest similarity score. If the two sentences have similarity value as 90% (threshold) and above, we consider that both sentences carry the same meaning.

3.4.5 Features

It helps to find the correct sense of a word present in a sentence by finding the contextual meaning of all the words.

3.5 Automatic Text Summarization Based on Word-Clusters and Ranking Algorithm

3.5.1 Overview

Automated text summarization dates back to the end of the fifties. Two main ideas have emerged to deal with this task; the first was how a summarizer has to treat a huge quantity of data and the second, how it may be possible to produce a human quality summary. To deal with the first point, simpler approaches were explored which consist in extracting representative text-spans, using statistical techniques or techniques based on superficial domain-independent linguistic analyses. selection of a subset of the document sentences which is representative of its content. This is typically done by ranking document text-spans with respect to the similarity measure with a relevant source. Most of the recent work in SDS uses this paradigm. Summarization systems can operate in two modes: generic summarization, which consists in extracting text-spans relevant to the main topics of a whole document and query-based summarization, which aims at abstracting the information relevant to a given query.

3.5.2 Method

Defined different sentence features it considered important for a generic summary and grouped them into seven categories: Indicator phrases (such as cue-words or acronyms), Frequency and title keywords, location as well as sentence length cut-off heuristics and the number of semantic links between a sentence and its neighbours.

3.5.3 Generic Queries

They start from two baseline generic queries constituted of the most frequent terms in the collection, MFT and no-stop words in the title of a document title keyword. These queries represent two sources of evidence they use to find relevant sentences in a document. Since title keywords may be very short, they have employed query-expansion techniques such as Local Context Analysis (LCA) or thesaurus expansion methods as well as a learning-based expansion

technique. Expansion via WordNet and LCA: From the title keyword query, they formed two other queries, reflecting local links between the title keywords and other words in the corresponding document: – title keyword and LCA, constituted by keywords in the title of a document and the most frequent words in most similar sentences to the title keyword query according to the cosine measure. Expansion with Word-Clusters: They first form different word-clusters based on words co-occurring in sentences of all documents in the corpus. From these clusters they obtained two other expanded queries by first adding to title keywords, words in their respective clusters, title keyword and term-clusters. And secondly by projecting each sentence of a document and the title keyword query in the space of these word clusters, Projected title keyword. For the latter they characterize each sentence in a document and the title keyword query by a vector where each characteristic represents the number of occurrences of words from a cluster in that sentence or in the title keyword query. The characteristics in this representation are related to the degree of representation of each word cluster in a given sentence or in the title keyword query.

3.5.4 Similarity Measures

They used the tf-idf representation and compute the cosine similarity measure between sentence x and query q as

$$Sim_1(q,s) = \frac{\sum_{w \in s,q} tf(w,q)tf(w,s)idf^2(w)}{||w||\,||s||}$$

Where, tf(w, x) is the frequency of word w in x (q or s), idf(w)is the inverse document frequency of word

$$||x|| = \sqrt{\sum_{w \in x} (tf(w,x)idf(w))^2}$$

They also expected to reweight sentences containing acronyms e.g. HMM (Hidden Markov Models), NLP (Natural Language Processing), ... The resulting feature computes similarity between the title keywords and sentences using the same similarity measure than Sim1 except that acronyms are given a higher weight. The resulting similarity measure writes

$$Sim_2(q,s) = \frac{\sum_{w \in s,q} tf(w,q) tf^*(w,s) idf^2(w)}{\|w\| \|s\|}$$

Hence, they have counted as twice the term frequency of acronyms e.g. tf ∗(w, s) = 2 ∗ tf(w, s) if w is an acronym. In our experiments, acronyms are extracted using the Acronym Finding Program. They have conducted experiments on scientific articles. For these documents, sentences containing any of a list of fixed phrases like "in this paper", "in conclusion", ... are more likely to be in summaries. They counted as twice the similarity of sentences containing such cue-words: Sim3(q, s) =2Sim1(q, s) if s contains cue-terms and Sim3(q, s) = Sim1(q, s) if s does not contain cue-terms. They believe that algorithms relying on the ML ranking framework will be more effective in practice for the SDS task. In this case, instead of classifying sentences as relevant/irrelevant, a ranking algorithm classifies pairs of sentences.

More specifically, it considers the pair of sentences (s, s) coming from a same document, such that just one of the two sentences is relevant. The goal is then to learn a scoring function H from the following assumption: a pair is correctly classified if and only if the score of the relevant sentence is greater than the score of the irrelevant one. The error on the pairs of sentences, called the Ranking loss of H [7], is equal to:

$$Rloss(\mathcal{D}, H) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \frac{1}{|\mathcal{S}_1^d||\mathcal{S}_{-1}^d|} \sum_{\mathbf{s} \in \mathcal{S}_1^d} \sum_{\mathbf{s}' \in \mathcal{S}_{-1}^d} [[H(\mathbf{s}') \geq H(\mathbf{s})]]$$

where D is the training document collection, Sd 1 the set of relevant sentences for document d, and Sd −1 the set of the irrelevant ones of the same document and [[$\pi \geq 0$]] is equal to 1 if $\pi \geq 0$ holds and 0 in the contrary case. It is straightforward that minimizing Ranking lossis equivalent to minimize the number of irrelevant sentences scored higher than the relevant ones of the same document. Ranking loss results then in a direct optimization of the ranks of the relevant sentences. This fact motivates the use of a ranking algorithm instead of a classification algorithm

3.5.5 Features

The features introduced are based on word clusters, which group words co-occurring in the same sentences. These clusters can be used to provide words for query expansion, or to enrich the representation of the sentences. In all cases, they show promising performance in terms of

precision/recall, but future work is needed to fully understand the difference between the two techniques as well as studying the effect of the number of clusters and their size on the performance of the features. This paper is the first one to propose the use of a ML ranking algorithm for SDS. They have shown empirically that ranking algorithms outperform classification algorithms.

3.5.6 Limitation

Ranking algorithms have a weaker working hypothesis than classification algorithms, and seem more appropriate to the SDS, although the difference of performance between the two depends on the dataset they are studying. However, important gains can be expected on specific datasets, while it is probable that classification algorithms can do worse. This understanding of the behavior of ranking algorithms can lead to its use on other tasks of passage level extraction, where the optimization criterion as well as the working hypothesis of ranking algorithms may be more suited than classification algorithms.

3.6 Abstractive Text Summarization based on Improved Semantic Graph Approach

3.6.1 Overview

This paper focuses on Multi-document Summarization. Various graph-based methods have also been explored for extra active summarization of multi-documents. These methods employ PageRank algorithm and its variations for computing the relative importance of sentences. Abstractive summarization methods are generally grouped into two categories: Linguistic (Syntactic) and semantic based approaches. Syntactic approaches for abstractive summaries include tree-based method, lead and body phrase method and information item- based method. However, semantic based approaches mainly use template-based methods and ontology-based methods. Multi document summary normally presents a brief topic description for collection of documents on same topic and assist the users to quickly scan many documents. A definite problem for MDS is that there is certain overlapping information in different documents about same topic. Hence, efficient summarization methods that combine similar information content across different documents are required. In this connection, numerous methods have been devised but suffer from some limitations. In particular, the above-mentioned graph-based models presented for multi-document extractive summarization (MDES) represent sentence as vector of words without comprehending its meaning. These models make use of content similarity to find sentence similarities, which may not be able to detect semantically equivalent redundant sentences, and

hence will lead to a poor final summary. On other hand, the graph-based abstractive summarization approach depends on humans and is restricted to single domain i.e. not applicable to other domains. Based on our literature knowledge, we know that semantic graph-based method has not been considered for MDAS. Therefore, this study introduces a semantic graph-based method for MDAS, which attempts to overcome the disadvantages of existing graph-based approaches. The contributions of this research are highlighted as given:

- Introduce a semantic graph approach for MDAS.
- Improve graph-based ranking algorithm by taking into consideration PASs semantic similarity, and two kinds of PAS (predicate argument structures) relationships.
- Integration of semantic similarity in the graph-based approach to determine semantic relationship between PASs, which will assist in detecting redundancy. PASs is assumed to be redundant if their similarity threshold is greater than 0.5. In other words, no link is established between PASs in the semantic graph whose similarity exceed 0.5.
- Propose a voting method for arrangement of sentences in the multi-document summary.

3.6.2 Method

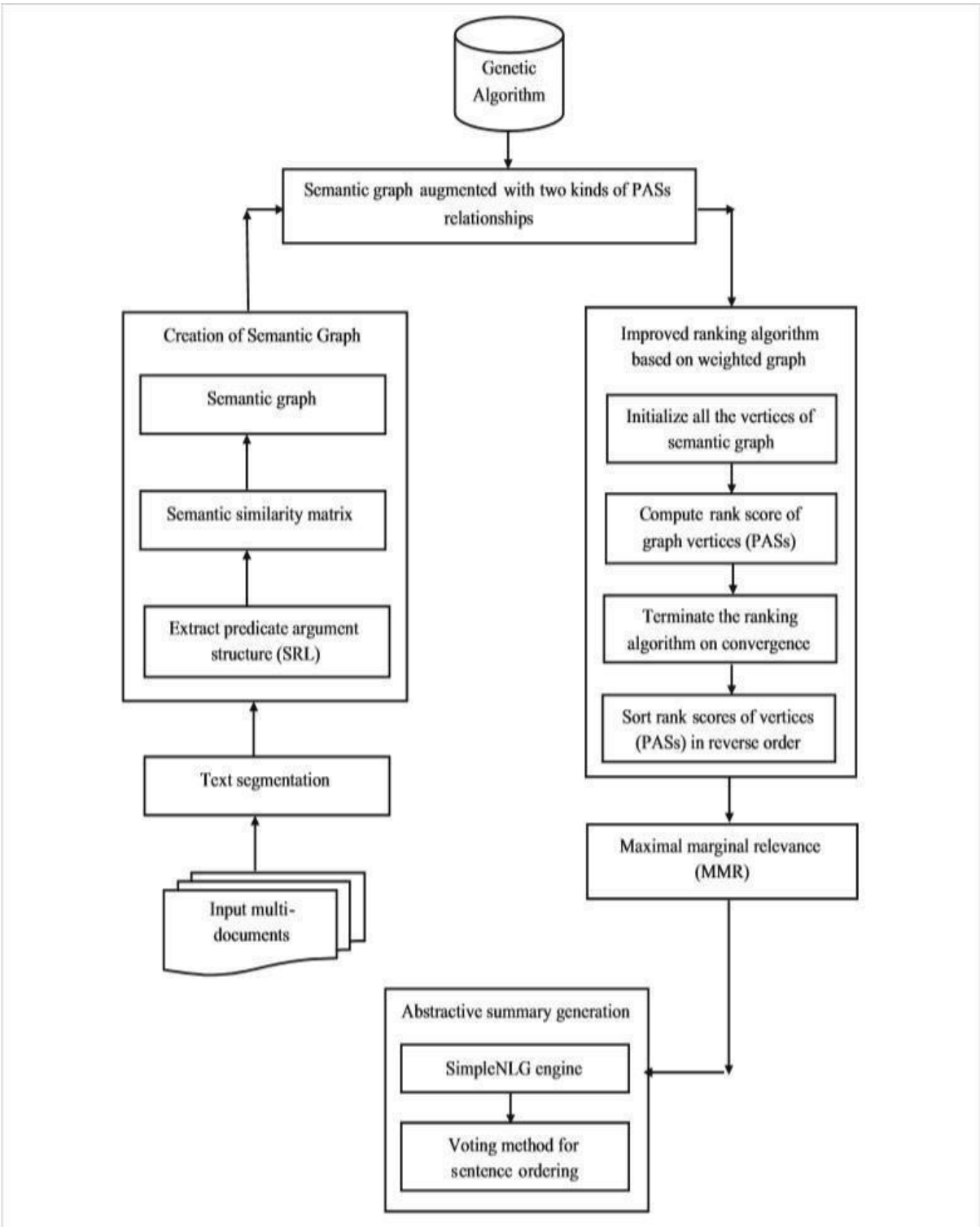


Fig.3.6.2. Abstractive Text Summarization Based on Improved Semantic Graph Approach

3.6.3 Creation of Semantic Graph

3.6.3.1 Semantic Role Labelling

The goal of this step is to parse each sentence in the document collection and extracts PAS from them. We begin by segmenting the document collection into sentences. Each sentence is given a key based on its document number and location number. 2.8.2.1.2 Semantic Similarity Matrix This step intends to calculate pair-wise semantic similarity scores PASs and then build a matrix from it. The verb, noun, time and location arguments of each PAS are compared with other PASs to compute pair wise similarities 2.8.2.1.3 Semantic Graph A semantic graph (an undirected weighted graph) is created from the similarity matrix in a manner that if similarity weightism pi, p j between PASs (vertices) pi and p j is greater than zero, then link is set up between PASs; otherwise a link is not created. So, the link or edge of graph represent similarity weight between PASs. This study defines a semantic similarity threshold i.e. $0 < \beta \leq 0.5$ based on empirical analysis [12]. Consequently, a link is created between PASs (vertices), if semantic similarity is within the range; otherwise no link is set up if similarity is outside the range. We let sim pi, p j = 0 to avoid self-transitions. PASs with similarity greater than 0.5 are assumed to be semantically equivalent, and are not incorporated in the semantic graph so that redundant PASs is ignored in the summary generation phase. The similarity weightism pi, p j between PASs pi and p j (i = j) is determined based on Jiang semantic similarity measure. 2.8.2.2 Semantic Graph Augmentation with PAS Relationships Previous graph-based approaches treat sentences uniformly in the same and diverse documents, i.e. the sentences are rated without considering relationships of sentence to document and document set. However, this work assume PAS—the semantic structure of sentence; and besides PAS-to-PAS semantic similarity, we also examine the relative importance of two kinds of PAS relationships in the importance analysis of PASs. The edge of semantic graph is augmented with PAS-to-document and PAS-to-document set relationships, whereas the graph edge represents semantic similarity weight between PASs. Based on text summarization literature, we employ text features that are more relevant to represent predicate argument structure to document relationship, which includes: title feature, location and PAS to document semantic similarity. Similarly, we also use text features that are more appropriate to signify the correlation/relationship of predicate argument structure to document set, and are given as follows: term weight and frequent term and PAS to document set semantic similarity. Summary quality is vulnerable to text featuresi.e. different feature have not same importance in the process of producing summary. GA is selected since it is an optimization technique which is robust, and employed in diverse disciplines of applications and research, especially to deal with the problems of optimization. The fitness

function computes the fitness values of chromosomes, which indicates how the good the chromosome is. When all the training multidocuments are summarized, then in this study, the best average recall will serve as a fitness function, which is computed from multi- documents summaries generated by ROUGE-1, and is given in the following Equation where n is the length of the n-gram, gramn and countmatch(gramn) is the number of ngrams that simultaneously occur in a system summary and a set of human summaries.

$$F(X) = \frac{\sum_{S \in \{Reference\ Summaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{Reference\ Summaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

3.6.3.2 Improved Weighted Graph Based Ranking Algorithm

Conventionally, Google's PageRank and HITS algorithm are ranking algorithms that have been effectively employed in link analysis and social networks. PageRank algorithm attempted on undirected graph gave the best performance in DUC 2002 extractive summarization task for multi-documents. The PageRank is a ranking algorithm that offers a way for determining the significance of a vertex in a graph by assuming the global information from the whole graph. Existing graph-based methods employ procedure similar to PageRank.

3.6.3.3 MMR to Control Redundancy

The ranking algorithm may assign same rank score to PASs representing the same concept and hence the final summary may contain redundancy. Furthermore, the other concepts of the documents which represents group of least similar PASs may not be included in the final summary, and thus significant information may lose. In this study, we employ a modified version of MMR to reduce redundancy by reranking the PASs for inclusion in summary. A predicate argument structure is included in the summary generation list if it is not too similar to any existing PAS in the summary generation list. At first, the PAS with the maximum salience score is chosen and appended into the summary generation list. Then, the subsequent PAS having the higher salience score according to Equation is selected and added into summary generation list. This process chooses PASs by taking into account both importance and redundancy and keeps on repeating until the compression rate of summary met.

$$IWGRA(v_i) = (1 - d_p) + d_p \cdot \sum_{v_j \in In(v_i)} \frac{IWGRA(v_j).w_{ji}}{\sum_{v_z \in Out(v_j)} w_{zj}}$$

3.6.3.4 Summary Generation

In this phase, we make use of Simple NLG along with simple heuristic rules defined in it, to produce summary sentences from the top scored representative PASs chosen in previous phase. Simple NLG engine utilizes simple English grammar rules to convert syntactical structures into sentences. The engine is robust i.e. it is not crashed, when partial syntactical structures are provided as input. After the generation of summary sentences, then we propose a voting scheme to rearrange the summary sentences. The scheme operates in four steps:

- **Step 1:** The voting scheme is based on the idea of voting or recommendation i.e. one sentence vote another sentence, if it addresses the same concept as described in another sentence, which is determined based on different similarity relations between them. In this work, a well know similarity measure, Jaccard similarity measure is employed to calculate the similarity between a summary sentence and source sentences in document set. A scoring threshold (β) is assigned a value of 0.5 for sentence similarity, according to the literature. Jaccard measure for sentences S1 and S2 is stated as the following equation.

  J (s1, s2) = |s1 ∩ s2| |s1 ∪ s2| (20) 123 Int J Parallel Prog
- **Step 2:** Once the similar source sentences (contained in the document set) to the corresponding summary sentence are identified, then the positions of the source sentences are averaged to give score to the corresponding summary sentence. We exploit position feature to determine the normalize position [0,1] of a source sentence in the document set using the following equation. Position of Source Sentence = Document length − Sentence position in doc + 1 Document length
- **Step 3:** Repeat step 1 and step 2 until scores are assigned to all summary sentences.
- **Step 4:** Arrange the sentences in the summary according to ascending order of scores.

3.6.3.5 Feature

There are three variants of our semantic graph based abstractive summarization approach that will be evaluated: the complete summarization approach (Sem Graph Both-Rel), which takes into account both PAS-document relationship and PA document set relationship, besides PAS to PAS semantic similarity.

3.6.3.5 Limitations

Existing graph-based approaches consider sentence as vector of words and cannot detect semantically equivalent redundant sentences, as they mostly rely on content similarity measure.

3.7 Abstractive text summarization using LSTM-CNN based deep learning

3.7.1 Overview

They have proposed an LSTM-CNN based ATS framework, named ATSDL, and make the following main contributions in this work:

They applied LSTM model that was originally developed for machine translation summarization, and combine CNN and LSTM together to improve the performance of TS. ATSDL considers both semantics and Multimedia Tools Apply syntactic structure, which is different from existing ETS models and ATS models. ATSDL firstly applies phrase extraction method to extract key phrases from sentences, and then uses the LSTM model to learn the collocation of phrases. After training, the new model will generate a sequence of phrases. This sequence is the text summary that is composed of natural sentences.

In order to solve the key problem of rare words, they used phrase location information, so they can generate more natural sentences.

The experiment results show that ATSDL outperforms state-of-the-art abstractive and extractive summarization systems on both two different datasets.

3.7.2 Method

Model can be divided into three steps: text pre-processing, phrase extraction and text summary generation.

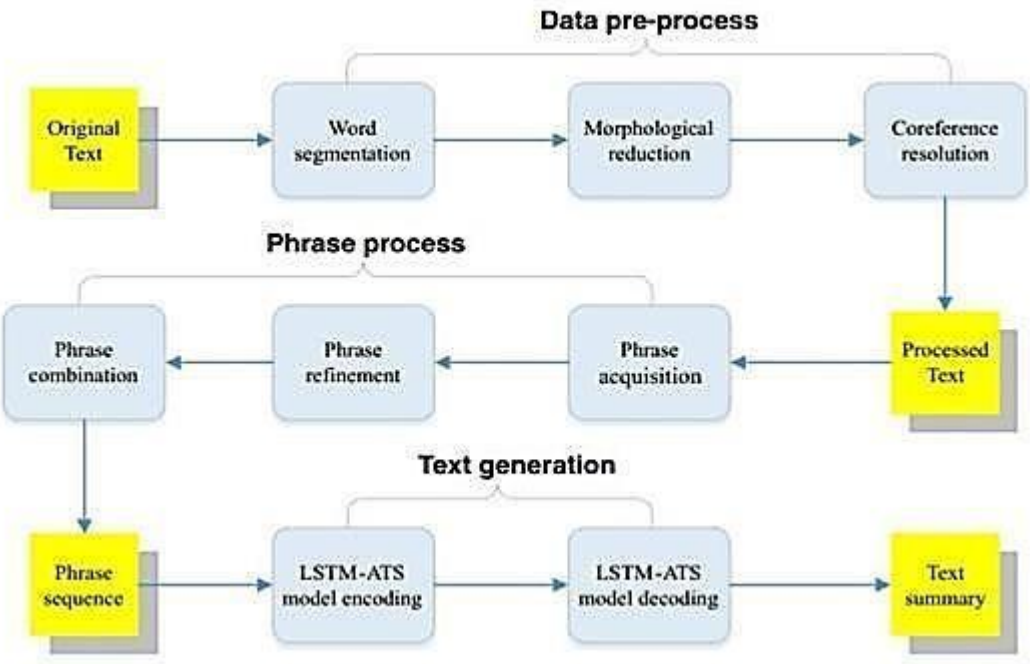


Fig 3.7.2 Abstractive text summarization using LSTM-CNN

3.7.3 Phrases extraction

Phrase extraction includes three sub-steps: Phrase acquisition, phrase refinement, and phrase combination. The task of this step is to extract as many phrases as possible. Therefore, a novel phrase extraction method called Multiple Order Semantic Parsing (MOSP) is proposed to construct a binary semantic multilayer structure to analyze multiple order semantics. The process of MOSP for scattering and restructuring as shown.

1. Firstly, sentences are split into fragments with semantic association information which contains lexical and syntactic features in sentences. The Stanford NLP
2. Then, with the aid of the dependency parser, the semantic information is reorganizing into a binary tree structure.
3. Then, we identify an initial root node as a starting point, which is the relational phrase in the main clause.
4. From the initial root, its associated relational argument pairs are assembled into a tree structure as a child node. If the current root has no child, construction of the semantic tree

is completed. Otherwise, the child node will be as a new root node for recursive construction.

5. Next, the dependency parsed finds if there is a compound clause. The relational phrase of other clauses is considered as brothers of the relational phrase of the main clause. For each brother, a conjunction node will be generated as a parent node to connect the its root and its brothers with its conjunction semantics.
6. Then, the relational phrase of the brother node will be treated as a root node to construct the semantic tree for the compound clause.
7. The constructed semantic tree in front may contain a single child subtree (only left child or only right child). Therefore, after the construction of the binary semantic tree, this tree will be traversed to upgrade to a strict binary tree, which is our MOS Tree. In the process of traversing, the optimization work mainly composes of two parts. We determine whether the subtree has complete semantic information. Different solutions to these two cases are designed to adjust the structure of the subtree. If it has binary semantics, the subtree will be processed to a strict binary tree.

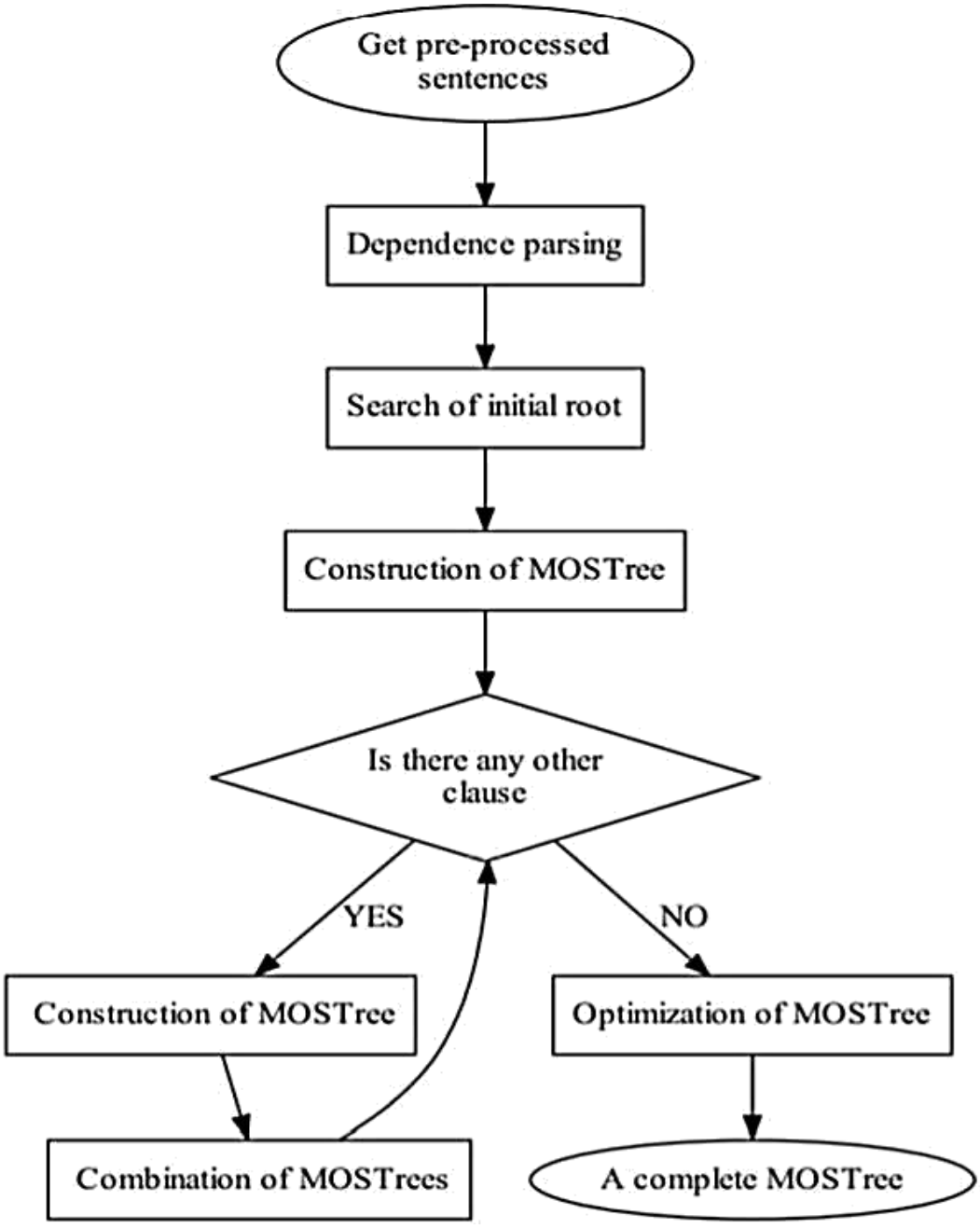


Figure 3.7.3 Abstractive text summarization using LSTM-CNN

3.7.3.1 Text summary generation

After phrase extraction, we need to generate a summary by our LSTM-CNN model. We implement the model in the deep learning framework Torch and encode phrases as one-hot. On average, our model takes about 8 min to train a text (approximately 3 KB). Training this model can be tricky, so we can use some heuristics and tips for the optimization. After the completion of our model training, we need to generate a summary by our model. We set a threshold δ, If the absolute value of conditional probability is more than threshold δ, our model will enter generate mode.

3.7.3.2 Evaluation method

In our work, we use the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) toolkit, which has been widely adopted by DUC for automatic summarization evaluation. ROUGE measures summary quality by counting overlapping units such as the n-gram, word sequences, and word pairs between the system summary and the reference summary. We choose automatic evaluation methods ROUGE-N in our experiment, which is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows:

Where n stands for the length of the n-gram, Ref is the set of reference summaries. Count match (gram n) is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries, and Count (gram n) is the number of n-grams in the reference summaries.

3.7.4 Features

1. Core NLP is a very useful tool because Core NLP makes it very easy to apply linguistic analysis tools to process text content. The accuracy of word segmentation using Core NLP is very high, but text pre-processing is time consuming. Accordingly, we hope that we can use or develop a more convenient text preprocessing tool to help us complete this task.
2. Determining semantic similarity between phrases is difficult. We use WordNet to complete this task. However, WordNet cannot give the similarity between phrases directly, so we may be able to provide potential improvements to solve this problem in future.
3. Training deep learning models are a very time-consuming task and determining the threshold δ in the decoder of the model requires manual tuning. In the future, we can analyze the relationship between the threshold δ and some influencing factors like text length and then give a more precise threshold definition.

4. Using the Rouge evaluation method to determine the quality of a generated summary is very limited. The Rouge evaluation method only calculates the repeatability of Ngram and cannot analyze the quality of a generated summary from

3.7.5 Limitations

The present ETS models are concerned with syntactic structure, while present ATS models are concerned with semantics. Our model draws on the strengths of both of summarization models. The new ATSDL model firstly uses phrase extraction method called MOSP to extract key phrases from the original text, and then learns the collocation of phrases. After training, the model will generate a phrase sequence that meets the requirement of syntactic structure. In addition, we use phrase location information to solve the rare words problem that almost all ATS models would encounter.

CHAPTER-4

PROPOSED METHOD

4.1 Proposed Framework

This project focuses on the Fuzzy logic extraction approach for text summarization and the graph-based approaches TextRank and LexRank for extracting the keyword using structure of the article and the semantic approach of text summarization using Latent Semantic Analysis. Our hybrid model consists of four components: TextRank, LexRank, LSA, Fuzzy Logic. In this model, each text summarization method is used to extract the keywords, each method has ranked the keywords based on their importance (recurrence, centrality, noun etc), These keywords with their scores are then given as input to the final keyword extractor. In this phase, the keywords occurring in the final result of all the four methods are taken into final keyword list, Then the remaining keywords of all the methods are arranged in the descending order of the scores. From this list the top m keywords are selected in the final output. Keyword extraction can be done to select as many keywords we want, but, usually 1% of the total keywords (except stop words) are enough to represent the central idea of the document.

Initially the document is passed through a pre-processing phase, to get only the required data. Like in the previous line, the keywords like is, the, a, to, only are not going to be the keywords representing the idea of the document. So, we first remove the unwanted stuff from the document and then also add some words which semantically mean the same, to get better semantic understanding. For example, the keywords like improving, enhancing are typically mean the same, so we can replace one with the other, so the actual occurrence of the word could be identified.

In TextRank the keywords or sentences of the document are represented as the vertices of the graph and the edges exist only between the vertices which has some sort of relationship like in case of sentences the common words, the position of two sentences, for keywords nouns, adjectives etc can be used for defining the relationship between the vertices. Number of common tokens like noun, verbs which are keywords to the text define the similarity of sentences or also called as overlapping nature.

LexRank calculates the importance of a sentence or keyword by finding their eigenvector centrality in the graph representation of the sentences. A connectivity matrix based on intrasentence cosine similarity is developed to find the relation of a sentence with other sentences in the document

In LSA Singular value decomposition (SVD) is used to capture and model inter-relationships among terms, so that it can semantically cluster terms that are recurring in the document and are

represented as singular vectors. It improves the chances of getting more relevant results, as semantic content of the words is used rather than the direct mapping.

Fuzzy Logic is used as it handles the imprecision and vagueness in the data, but it does not consider the semantics of the text. Feature score of each word are calculated using statistical methods. First each feature is inputted with Trapezoidal membership function, then IF-THEN rules are applied to get the output score of each sentence. Summary generated by each method individually is passed on to summary selector which arranges the keywords in descending order of their scores. Top n unique keywords are selected from the list giving the keywords identifying the central idea of the document.

CHAPTER 5

ARCHITECTURAL VIEW

5.1 Architectural View

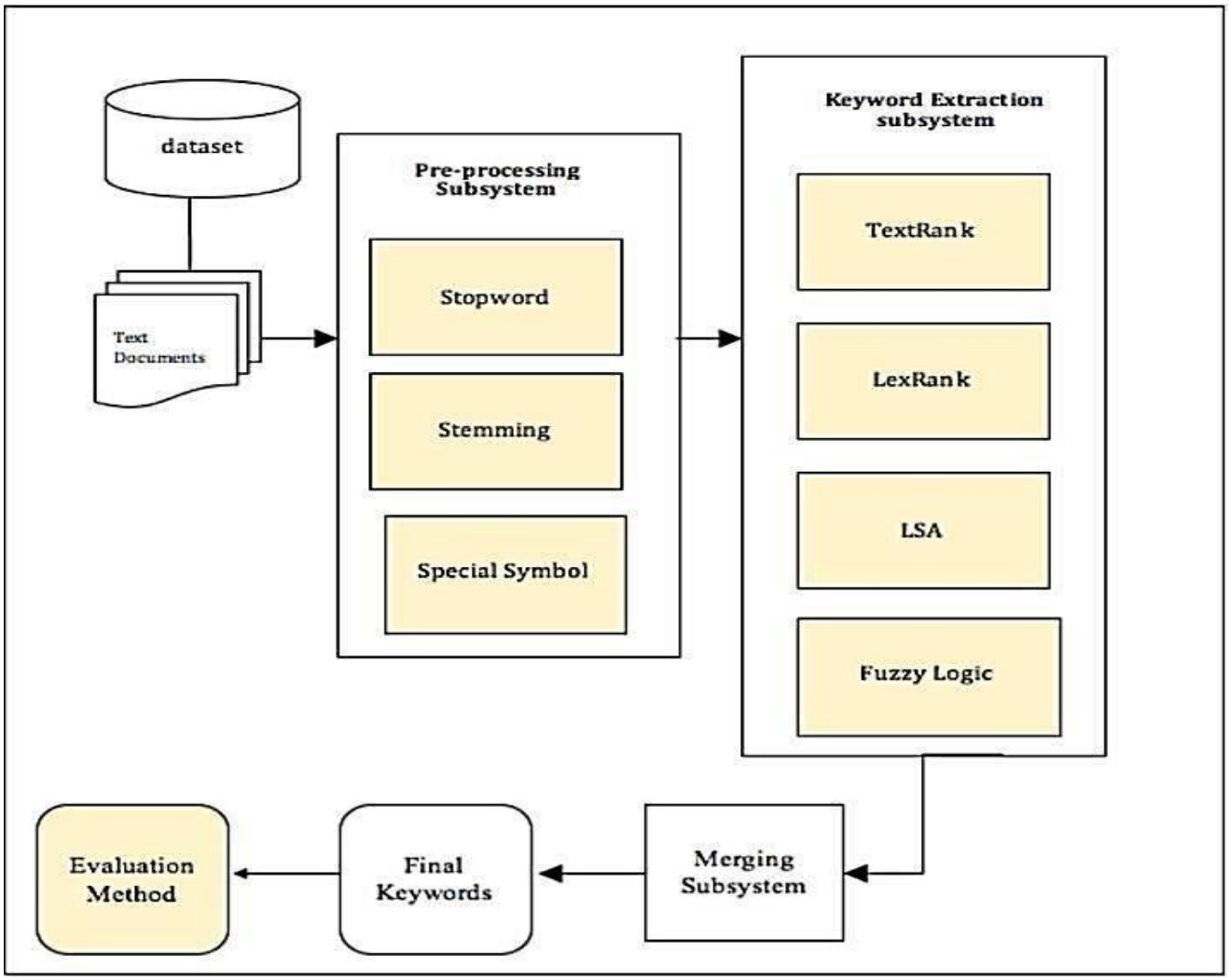


Fig 5.1 Architectural View

The proposed approach firstly retrieves text documents from Opinosis dataset and then preprocess them by removing the unwanted words and converting rest of the words to its stem(root). For Keyword extraction the four methods process the document in their own way. Later, the keywords extracted from all the methods are merged and arranged in descending order of their scores. Finally, the top 1% keywords are extracted from the list. Figure shows the overview of the system proposed in this research.

5.2 Proposed Algorithm

Each of the four techniques were first implemented separately, then their output is given as the input to the algorithm to extract more effective keywords, for determining the central idea of the document. Following figure shows the detailed flow chart of the proposed method containing step-wise procedure of all the four methods.

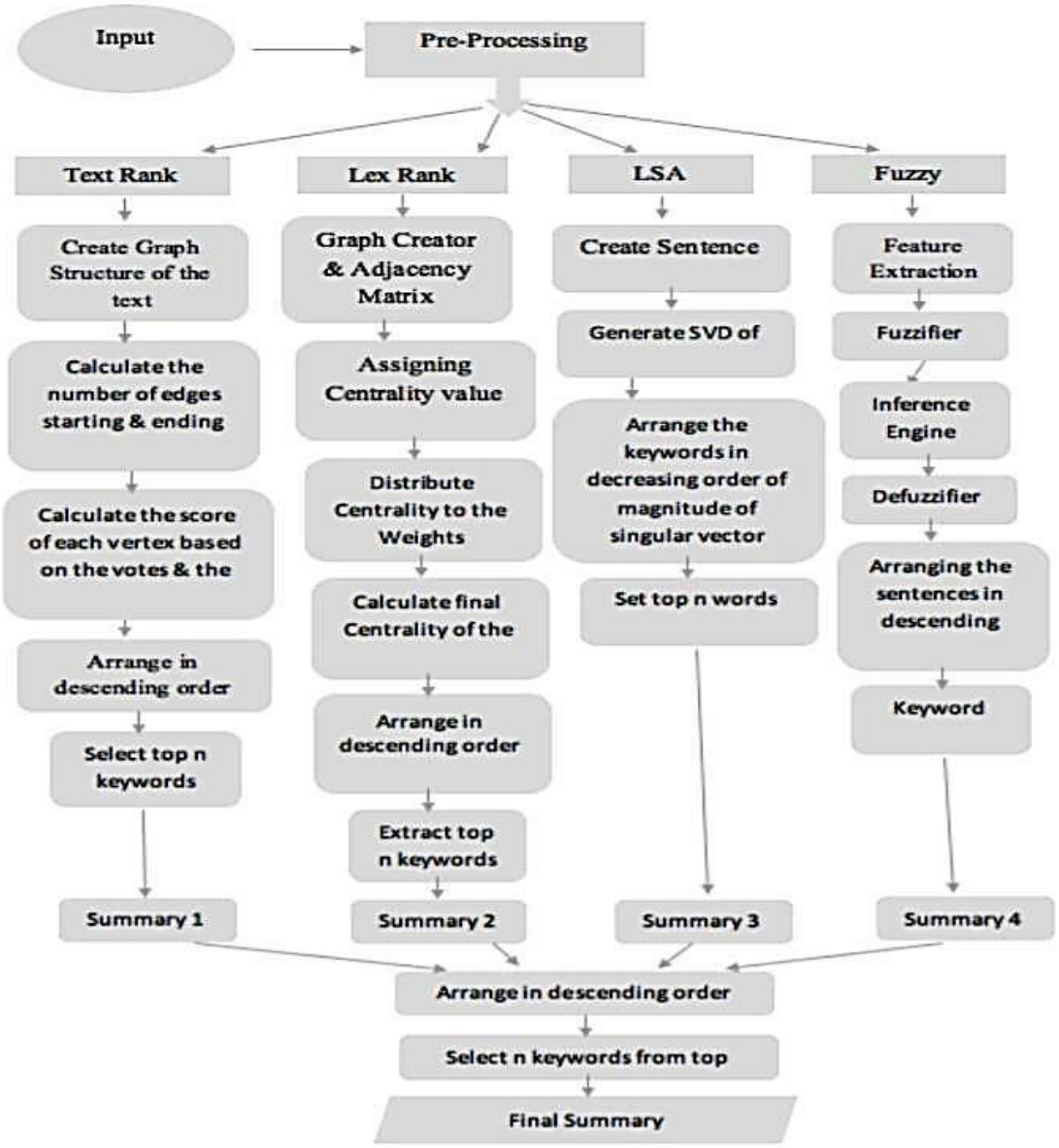


Fig 5.2 Proposed Algorithm

CHAPTER 6

PROGRAMMING ASPECT

6.1 Programming Aspect

Firstly, features selection process is done, in our approach we have used eight features for determining the prominence score of a sentence These are Title word, Sentence length, sentence position, numerical data, thematic words, sentence to sentence similarity, term weight and proper noun. The feature score is then given as input to fuzzifier, which using trapezoidal membership function represents these scores in fuzzy values. These are then passed on to the inference engine, where based on the fuzzy rule set, the status of the sentence is extracted. The final step is defuzzifier which using the triangular membership function provide the prominence score to each sentence. These sentences with their score are then given as input to the merger subsection of the proposed method.

The rest of the three methods TextRank, LexRank and LSA were implemented in a Linux environment using python and bash. The bash scripts were written to act as drivers for automation of the project. The algorithms were implemented with python. The output file of each of the file is then passed onto the merger subsystem.

The final summary done by a program which took four files generated by each method as input files containing the sentence and their scores determining their importance in the document. Firstly, the content of all four files are merged in one file, then the sentences which were occurring in more than one input file are kept only once to remove the redundancy. The sentences which were coming in more than one algorithm as the important sentence are first taken into the output file and are removed from the merged document. The rest of the sentences in the merged document are then arranged in descending order of their scores. If a document is of 1000 sentences and we want summary of only 20 lines that is only 2% of the actual document, we will select the top 20 sentences from the arranged list, only if none of the sentence was occurring in more than one method. If some sentences were coming as important in more than one method, say 3 sentences were found in summary of more than one method, then from final list only 17 sentences will be selected.

CHAPTER 7

RESULT AND DISCUSSION

7.1 Result and Discussion

This project focuses on the Fuzzy logic extraction approach for text summarization and the graph-based approaches TextRank and LexRank for extracting the keyword using structure of the article and the semantic approach of text summarization using Latent Semantic Analysis. Our hybrid model consists of four components: TextRank, LexRank, LSA, Fuzzy Logic. In this model, each text summarization method is used to extract the keywords, each method has ranked the keywords based on their importance (recurrence, centrality, noun etc), These keywords with their scores are then given as input to the final keyword extractor. In this phase, the keywords occurring in the final result of all the four methods are taken into final keyword list, Then the remaining keywords of all the methods are arranged in the descending order of the scores. From this list the top m keywords are selected in the final output. Keyword extraction can be done to select as many keywords we want, but, usually 1% of the total keywords (except stop words) are enough to represent the central idea of the document. Initially the document is passed through a pre-processing phase, to get only the required data. Like in the previous line, the keywords like is, the, a, to, only are not going to be the keywords representing the idea of the document. So, we first remove the unwanted stuff from the document and then also add some words which semantically mean the same, to get better semantic understanding. For example, the keywords like improving, enhancing are typically mean the same, so we can replace one with the other, so the actual occurrence of the word could be identified.

CHAPTER 8

SUMMARY AND CONCLUSION

8.1 Research Summary

The evolution of Web from the 'Web of Documents' to the 'Web of People' (Social Web) to the 'Web of Data' (Semantic Web) has multiplied the volume, velocity and variety of information available online. Finding useful, relevant, trending, interesting information from this 'garbage in-garbage out' puddle has been a major focus of current research. Text summarization has emerged as one of the vital techniques to handle this problem. The incredible ability of fuzzy inference systems to make logical assessment in an ambiguous and uncertain environment has made is a trending choice for practical applications such as text summarization, that involve imprecision and uncertainty. In our work, we have reviewed various studies on research papers based on text summarization from 2017 to 2019, published in conference proceedings and journals of high repute. The purpose was to evaluate the progress made so far, identify the trends and gaps in studies to ascertain the future scope of research within the domain. The following key-points were observed:

- Fuzzy logic is an attractive choice to achieve optimal summaries due to its resemblance to human reasoning. The goal of text summarization is to achieve superlative results comparable to human generated summaries and mapping the fuzzy logic inference mechanism gives the desired brainpower.
- It is promising to see the paradigm shift from conventional summarization methods to contemporary, novel intelligence-based methods. Nearly 50% studies have been done with hybrid models of fuzzy logic with optimization methods, followed by equal number of studies done on hybrids of fuzzy-logic with statistical and semantic techniques respectively. Combinational hybrids using two or more conventional techniques with fuzzy-logic though have been reported but the sphere is an open research problem to achieve enhanced summarization results.
- Hybrid models taking semantic into consideration are generating better results than the other methods

The research in this work introduces a fuzzy logic-based hybrid model taking 2 graphs based and one semantic based technique together to generate an upgraded summary, which performs better than all the fuzzy logic-based models.

CHAPTER 9

FUTURE SCOPE

This study is empirical in nature and can be improved by giving different weights to the four techniques, the weightage given to the methods can affect the output of the proposed method drastically. Features taken in the feature extraction step can also be given different weightage to make the summary creation process like humans, as we human don't take each attribute as same, like the title keywords are given more importance than the noun. So, the different weights assigned to each feature can improvise the results.

We can further extend this work by:

- Assigning different weights to each algorithm used.
- Assigning different weights to the features extracted in the process of summary generated using Fuzzy logic Incorporate some more Nature Language based (NLP) techniques to generate abstractive summary.

CHAPTER 10

RESEARCH

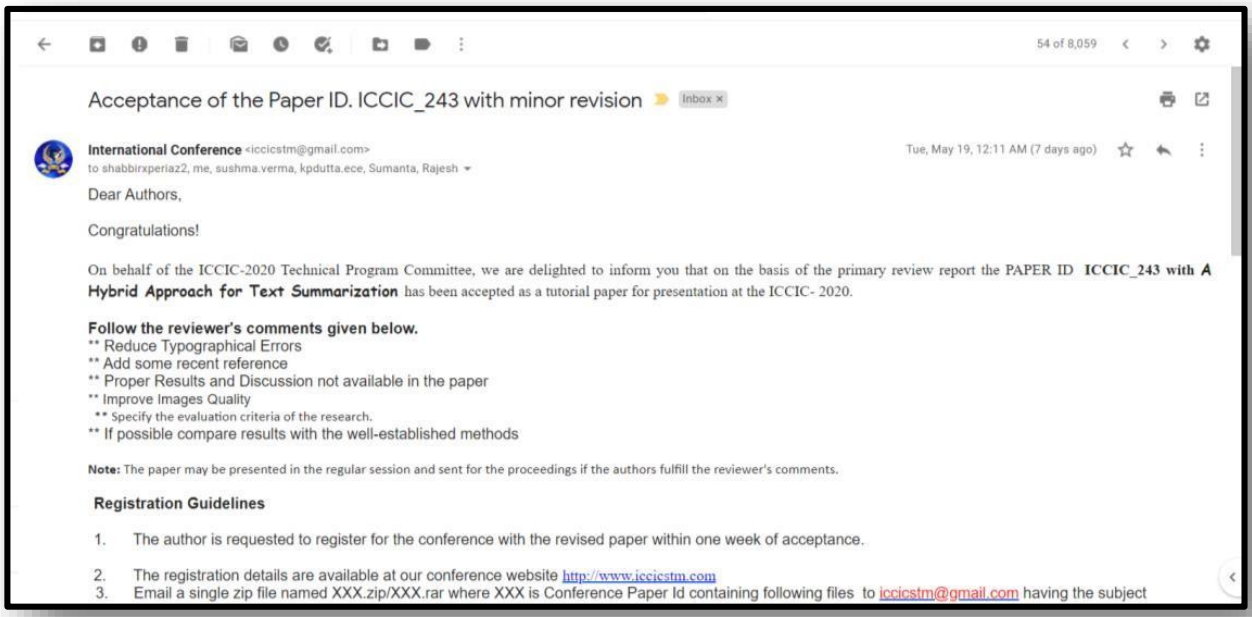
10.1 Research Paper Details

Publication Conference: International Conference on Contemporary Issues in Computing - ICCIC 2020

Conference Date: 25-26 July 2020 Conference Venue: Kolkata, West Bengal Paper Id:- ICCIC_241

Author Name: Shabbir Sidhpurwala , Saiyam Jain

Corresponding-Author Name: Sushma Verma



54 of 8,059

Acceptance of the Paper ID. ICCIC_243 with minor revision Inbox ×

International Conference <iccicstm@gmail.com> Tue, May 19, 12:11 AM (7 days ago)
to shabbirxperiaz2, me, sushma.verma, kpdutta.ece, Sumanta, Rajesh

Dear Authors,

Congratulations!

On behalf of the ICCIC-2020 Technical Program Committee, we are delighted to inform you that on the basis of the primary review report the PAPER ID **ICCIC_243 with A Hybrid Approach for Text Summarization** has been accepted as a tutorial paper for presentation at the ICCIC- 2020.

Follow the reviewer's comments given below.
** Reduce Typographical Errors
** Add some recent reference
** Proper Results and Discussion not available in the paper
** Improve Images Quality
** Specify the evaluation criteria of the research.
** If possible compare results with the well-established methods

Note: The paper may be presented in the regular session and sent for the proceedings if the authors fulfill the reviewer's comments.

Registration Guidelines

1. The author is requested to register for the conference with the revised paper within one week of acceptance.
2. The registration details are available at our conference website http://www.iccicstm.com
3. Email a single zip file named XXX.zip/XXX.rar where XXX is Conference Paper Id containing following files to iccicstm@gmail.com having the subject

Fig 10.1 Research Paper Acceptance

CHAPTER 11

BIBLIOGRAPHY AND REFERENCES

BIBLIOGRAPHY AND REFERENCES

[1] Abbasi-ghalehtaki, Razieh, Hassan Khotanlou, and Mansour Esmaeilpour. "Fuzzy evolutionary cellular learning automata model for text summarization." Swarm and Evolutionary Computation 30 (2016): 11-26.

[2] Aggarwal, Charu C., and ChengXiangZhai, eds. Mining text data. Springer Science & Business Media, 2012. Aggarwal, Charu C., and ChengXiangZhai, eds. Mining text data. Springer Science & Business Media, 2012.

[3] Cambria, Erik, and Bebo White. "Jumping NLP curves: A review of natural language processing research." IEEE Computational intelligence magazine 9, no. 2 (2014): 48-57.

[4] Witte, René, Ralf Krestel, and Sabine Bergler. "ERSS 2005: Coreference-based summarization reloaded." In DUC 2005 Document Understanding Workshop, Canada. 2005.

[5] Kyoomarsi, Farshad, Hamid Khosravi, Esfandiar Eslami, and Mohsen Davoudi. "Extraction-based text summarization using Fuzzy analysis." Iranian Journal of Fuzzy Systems 7, no. 3 (2010): 15-32.

[6] Binwahlan, Mohammed Salem, Naomie Salim, and LaddaSuanmali. "Fuzzy swarm diversity hybrid model for text summarization." Information processing & management 46, no. 5 (2010): 571-588.

[7] Kyoomarsi, Farshad, Javad Akbari Torkestani, Pooya Khosravyan Dehkordy, and Mohammad Hosseyn Peyravi. "Using chemical cellular automata in simulation of chemical materials." In 5th ACIS International Conference on Software Engineering Research, Management & Applications (SERA 2007), pp. 769-773. IEEE, 2007.

[8] Megala, S. Santhana, A. Kavitha, and A. Marimuthu. "Enriching text summarization using Fuzzy logic." International Journal of Computer Science and Information Technologies 5, no. 1 (2014): 863-867.

[9] Kumar, Yogan Jaya, Naomie Salim, Albaraa Abuobieda, and Ameer Tawfik Albaham. "Multi document summarization based on news components using Fuzzy cross-document relations." Applied Soft Computing 21 (2014): 265-279.

[10] Khan, Atif, Naomie Salim, and Yogan Jaya Kumar. "A framework for multi-document abstractive summarization based on semantic role labelling." Applied Soft Computing 30 (2015): 737-747.

[11] Lee, Chang-Shing, Yea-Juan Chen, and Zhi-Wei Jian. "Ontology-based Fuzzy event extraction agent for Chinese e-news summarization." Expert Systems with Applications 25, no. 3 (2003): 431-447.

[12] Jianpeng Cheng, Mirella Lapata “Neural Summarization by Extracting Sentences and Words”

ACL2016 conference paper with appendix(2016).

[13] Sumit Chopra, Michael Auli, Alexander M. Rush. “Abstractive Sentence Summarization with Attentive Recurrent Neural Networks.” San Diego, California, June 12-17, 2016. c 2016 Association for Computational Linguistics.

[14] Shashi Narayan, Shay B. Cohen, Mirella Lapata.“Ranking Sentences for Extractive Summarization with Reinforcement Learning.” NAACL 2018

[15] Xingxing Zhang, Mirella Lapata, Furu Wei, Ming Zhou.” Neural Latent Extractive Document Summarization” to appear in EMNLP 2018