VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018

Project (20CIPP83)

Report on

"EXTRACTIVE SUMMARIZATION OF TEXT DOCUMENTS"

Submitted in the partial fulfillment of the requirement for the Project

Bachelor of Engineering
In
CSE(AI&ML)

Submitted by

Mr. Manu S	1NC20CI024
Mr. Pruthvi P	1NC20CI008

Under the Guidance of
Prof. A. Nithyakalyani

Assistant Professor

DEPARTMENT OF CSE(AI&ML)

NAGARJUNA COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution under VTU, Belgavi-590018)
VENKATAGIRIKOTE, DEVANAHALLI, BENGALURU– 562164

2023-2024

NAGARJUNA
COLLEGE OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF CSE (Artificial Intelligence &Machine Learning)

CERTIFICATE

This is to Certify that the project **"EXTRACTIVE SUMMARIZATION OF TEXT DOCUMENTS"** carried out by Mr. Manu S(1NC20CI024), Mr. Pruthvi P (1NC20CI008), Bonafide students of Nagarjuna College of Engineering and Technology, Bangalore an autonomous college under Visvesvaraya Technological University, Belagavi, in partial fulfillment for the project work carried out by them for the course Project during the year 2023-2024.

It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report and submitted to the department library. The project report has been approved, as it satisfies the academic requirements of Project.

Signature of the Guide	**Signature of the HOD**	**Signature of the Principal**
Prof. A. Nithyakalyani	**Dr. Lohith J.J**	**Dr. B.V. Ravishankar**
Assistant Professor	Professor &HOD	Principal, NCET
CSE(AI&ML) Dept. NCET	CSE(AI&ML) Dept. NCET	

Internal/External Viva

Name of the Examiners	**Signature with date**
1.	
2.	

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose consistent guidance and encouragement crowned our effort with success. we consider it is our privilege and duty to express our gratitude and respect to all those who guide us in the completion of this project report.

First and foremost, it's our immense pleasure to thank our beloved guide **Nithyakalyani A, Assistant Professor** Department of CSE (Artificial Intelligence & Machine Learning), Nagarjuna College of Engineering and Technology, for helping, guiding and strengthening us to complete this project work.

We would like to thank our Project coordinator **Dr. Rashmi P Karchi, Professor,** Department of CSE (Artificial Intelligence & Machine Learning), Nagarjuna College of Engineering and Technology, Bengaluru for her constant support and assistance at every stage.

We would like to express our sincere thanks to **Dr. Lohith J.J, Professor& HOD,** Department of CSE (Artificial Intelligence & Machine Learning), Nagarjuna College of Engineering and Technology for his valuable suggestions and guidance throughoutthe period of this project report.

We take this privilege to express our deep gratitude to **Dr. B. V. Ravishankar, Principal**, Nagarjuna College of Engineering and Technology for his constant support and encouragement in preparation of this report and for providing library and laboratory facilities needed to prepare this project report.

Last but not least, we would like to thank our parents, friends, teaching and non- teaching staff of NCET

Mr. Manu S(1NC20CI024)

Mr. Pruthvi P(1NC20CI008)

ABSTRACT

Automatic Text Summarization, one of the applications of Natural Language Processing, provides a concise, fluent summary of a long passage or document, retaining the salient information. With the ever-growing number of learning resources, it is becoming difficult for students to choose the best learning resource. Also, many students find it difficult to understand and retain highly factual content. The main objective of this research is to provide information at a glance by mimicking the student's style of note-making. This is achieved through two steps namely summarization and tabulation. Extractive method for automatic text summarization is adapted, as it is proven to work better for non-fictional text containing a lot of facts. The performance of various approaches on a scholastic domain-specific dataset is carried out using ROUGE as an evaluation metric. A novel approach of tabulation of the essential facts and figures that often needs to be memorized is proposed. It will help in the easy retrieval and faster consumption of important information by the student. To achieve this, Named Entity Recognition and Basic Entity-relationship techniques are used to extract noun-chunks from the passage that is suitable for tabulation. It is then evaluated using measures that include Recall and entity comparison. Automating the process of note-making saves a lot of time and effort, thus helping students to learn and revise more efficiently.

Keywords: Information Retrieval, Extractive Summarization, Single document summarization, TextRank, Word Embedding, E-Learning, Tabulation, Named Entity Recognition, Deep Learning, Visualization, ROUGE.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION AND OBJECTIVE

1.1 INTRODUCTION

In today's Information Age, there is a state of exponential growth of data. Due to the flooding of data, the ability to wrangle, store, manipulate, analyze, interpret, utilize, and work upon the data is a principal concern. In the educational domain, with thick textbooks and large digital data, it is a great difficulty for people to refer and succeed in finding the matter they intended to, due to the large volume and veracity of data. This research deals with extractive summarization and tabulation of scholastic data which will help students, research scholars, teachers/lecturers and other interested audiences to get a synopsis of the chapters and units of textbooks of various subjects. To save time of going through the entire text, deciding which content to look up further or to perform a quick reading of a text, summaries come about as a great aid. It also improves the effectiveness of retrieval and accessibility of learning materials in Electronic Learning (E-learning). Additionally, students can also assess the literary value of the resources and learn from the resource that best suits them. 1.1 Summarization The principle idea of summarization is to find a subset of text which contains the pith and core of it. Text summarization can be broadly classified as extractive text summarization and abstractive text summarization. Abstractive summarization refers to the process of generating a set of new sentences that best explains the source document and is very close to how humans write summaries using their own words, that might not even be present in the given document. On the other hand, Extractive Summarization is selecting a subset of sentences that reserve the important details of the source document. It can also help the reader assess the suitability of the author's writing style of the document, for their understanding. Extractive summaries work better in terms of the time and cost of internal operations and they don't have language- generation problems as mentioned in the paper written by Nallapati et al. (2017). There are several other categories of summarization as well. For example based on the number of input documents, it can be classified as single-document summarization and multi-document summarization. This research will focus on generating an extractive, informative, genre- specific, mono-language, single document summary based on the categories mentioned in the paper published by Rahimi et al. (2017). One of the objectives of this research is to compare different approaches and find an effective way to produce the above mentioned summary using extractive summarization.

Tabulation is the arrangement of data into columns and rows where each column signi- fies the different attributes of a fact or figure. A well-defined table is the one which is logical, well balanced in length and breadth and comparable columns are placed side by side. It also briefs the text which helps in for diagrammatic representation. It is a cheap mode for presenting the data and condensing it. Largely scattered data can be collected into a table efficiently providing maximum information. Considering the E-learning domain, the students need to memorize the dates in history, formulae in physics and maths, types and functions of different parts of the body in biology, author and poet names in English literature, etc. This kind of data can be understood and absorbed easily by students when given in the form of a table. Tables have always been a great way of data representation as seen in book publications, multiplication tables and logic tables in arithmetic, periodic tables in natural sciences and spreadsheet software in soft- ware. Even computer storage is in the table format, though the usage in that case is purely for indexing. On the other hand, pictorial representation can be understood in a better way, but that is textual dependent. Experiments in chemistry or processes in biology are better represented using a flowchart. Graphs can also be a very efficient way of presenting notes of physics. Since, the data under consideration for this paper is based on History, a table is deemed best suited.

Now, Tabulation is largely based on NER. Till now, it has been used for Word Sentence Disambiguation (WSA), summarization, Information Retrieval (IR)or Information

Extraction (IE), Question-Answer chatbots (QA) etc. We introduce another use case for NER through our proposed methodology with various other significant steps for the tabular visualization.

1.2 OBJECTIVE

1. Content Digestion and Simplification

Scholastic textbooks often contain copious amounts of information, which can overwhelm learners. Extractive summarization aims to distill this wealth of content into its most crucial elements. By condensing lengthy passages and complex concepts into concise summaries, students can more easily grasp the core ideas without being inundated with unnecessary details. Textbooks often contain vast amounts of information, which can be overwhelming for students. Extractive summarization aims to distill this content into its essential components, making it more digestible and manageable. By condensing lengthy passages into concise summaries, students can grasp the main ideas without being bogged down by unnecessary details.

2. Enhancing Comprehension and Retention

Cognitive overload is a common obstacle to effective learning. Extractive summarization mitigates this by presenting information in a structured and digestible format, which aids comprehension and retention. By focusing on the essential points, learners can better understand and remember key concepts, leading to improved academic performance. Summarization requires students to evaluate the significance of information and make informed decisions about what to include and exclude. This cultivates critical thinking skills, as students must assess the relevance and importance of various ideas and arguments. By engaging in this process, students develop their ability to discern key concepts, distinguish between main ideas and supporting details, and construct coherent summaries.

3. Improving Information Accessibility

Summaries and tabulations make educational materials more accessible to a wider range of learners. For students with learning disabilities, language barriers, or limited reading proficiency, complex textbooks can pose significant challenges. Extractive summarization simplifies the content, making it more understandable and accessible to diverse learners. Additionally, tabulations provide a visual representation of information, which can aid comprehension for visual learners. Educational materials should be accessible to learners of all backgrounds and abilities. Summaries and tabulations enhance accessibility

by simplifying complex content and providing visual representations of information. This benefits diverse learners, including those with learning disabilities, language barriers, or limited reading proficiency.

4. Promoting Efficiency in Learning and Teaching

Extractive summarization saves time for both students and educators. Students can quickly review summarized content to gain a comprehensive understanding of the material, while educators can create study materials more efficiently by extracting key information from textbooks. This frees up time for deeper exploration of topics, interactive learning activities, and personalized instruction. Extractive summarization encourages students to engage critically with the material. When tasked with summarizing text, learners must analyze and synthesize information to identify the most salient points. This process fosters analytical thinking skills, as students learn to evaluate the relevance and significance of various ideas and arguments.

5. Optimizing Time Management for Students and Educators

Extractive summarization saves time for both students and educators. Students can efficiently review summarized content, while educators can create study materials more quickly by extracting key information from textbooks. This time-saving aspect allows for more effective use of instructional time and promotes a deeper exploration of topics. Summarization encourages learners to take ownership of their education. By summarizing text themselves, students actively engage with the material and develop self-directed learning skills. This autonomy fosters independence and prepares students for lifelong learning beyond the classroom.

6. Improving Accessibility and Inclusivity

Educational materials should be accessible to learners of all backgrounds and abilities. Summaries and tabulations enhance accessibility by simplifying complex content and providing visual representations of information. This benefits diverse learners, including those with learning disabilities, language barriers, or limited reading proficiency. Summaries and tabulations can facilitate collaborative learning experiences. Students can compare and discuss their summaries, fostering peer-to-peer teaching and collaborative problem-solving. This collaborative approach promotes a deeper understanding of the material through active engagement and knowledge sharing among peers.

CHAPTER-2

LITERATURE SURVEY

Table 2.1 Literature Survey

SI. NO	TITLE, AUTHOR	ABSTRCT	METHODS/TECHNIQUES	CONCLUSION
[1]	Text summarization using latent semantic analysis.. Authors: Ozsoy (2015)	The pioneer of automatic text summarization, came up with the idea that the close proximity of the greatest number of most frequent words indicate the significance of that content in the document, in his publication in the year 2015.This led to emergence of the concepts of term frequency.	The study led to emergence of the concepts of term frequency. After this, many algebraic, statistical methods were proposed based on Singular Value Decomposition (SVD) and Latent Semantic Analysis (LSA).	Based on Singular Value Decomposition (SVD) and Latent Semantic Analysis (LSA). Different text summarization approaches based on LSA are discussed in the paper published by Ozsoy et al. (2015). But one common drawback in these methods was that they failed to consider the meaning and context of the words in the document.
[2]	Optimizing text summarization based on fuzzy logic Authors: Kyoomarsi (2008), Suanmali (2009).	A model that works based on fuzzy rules and cue features based on paragraph and document level in addition to sentence level features, It involved a parserto perform pre- processing and identify non-structural features of a sentence such as location of the sentence . It involved computing	A fuzzy inference engine is used to rank the sentences for the summary, a prototype with pre- processing steps, that are even followed today, suchas stop-word removal, stemming, sentence boundary identification and tokenization.	It involved computing scores for sentences features like, thematic words, sentence similarity and numerical content which werethen processed by a fuzzy logic

		scores for sentences features like, thematic words, sentence similarity and numerical content which were then processed by a fuzzy logic system to extract the significant sentences.		system to extract the significant sentences, fuzzy-logic worked betterthan the surface- level approaches, there were problemssuch as dangling anaphora that still existed.
[3]	Automatic extractive text summarizatio n using k-means clustering. Authors: Ingole(201 2),Shetty and Kallimani(2017)	Clustering algorithm was used for summarization inthe year 2012.They have proposed a modelfor creating a query-specific summarization by building a distance matrix and using a queryprovided by the user, and the relevant sentences are assigned weights through the EM algorithm.	The K-means Cluster algorithm for summarization was presented in the paper where a cosine similarity matrix is created using Term Frequency- Inverse Document Frequency (TF-IDF)rules and cosine similarity, after the initial pre-processing steps.	In this model, sentences are grouped into clusters based on the K-means algorithm, where the input provided is the number of clusters; and one sentence is picked out from each cluster for the final summary. This paper concludes that K-means is a primitive algorithm and onlyprovides mediocre performance.
4	Extractive summarization using supervised and unsupervised learning Authors: Mao (2019)	In this paper, It is based onhubs and authorities. Authoritative pages are those that contain the search query andstring and have the most numberof incoming links. The hub pages are those that contain relevant links to authorities. Hence, the HITS algorithm distinguishes	According to this technique, the morethe links to a document, the moreimportant the document is considered to be. Based on these measures, it ranks the web pages in thesearch results. Google, the full-textsearch engine uses PageRank and othergraph-based approaches for text	In the supervised approach, sentenceranking is performed, by considering the sentences at the beginning of the document to be of greater importancethan other sentences at the end. Term Frequency-Inverse Sentence

		incoming links and outgoing links in the graph.	summarization in natural language text; they are based on the fundamentalsof PageRank.	Frequency (TF-ISF) is used to calculate sentence level scores. The cosine similarity isused to assess the relationship between sentencesin a graph-based model.
5	Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. Authors:Nallapati(2017),Zhang (2017)	A Convolutional Neural Network (CNN) model that was widely adapted for sentence embedding and text summarization . It was a simple CNN model withmax-overtime- pooling operation, that used word2vec as a word embedding technique.	Recurrent Neural Network (RNN), where the sentences in the document are visited sequentially and a binary decision is taken whether or not that sentence should be present in the final extractive summary.	Both extractive and abstractive training is proposed in this paper. In the extractive training, agreedy approach is used to create extractive labels for abstractive reference summaries. Both extractive and abstractive training is proposed in this paper. In the extractive training, a greedy approach is used to create extractive labels for abstractive reference summaries.
6	Medical text summarization system based on named entity recognition and modality identification. Authors: Aramaki E , Miuara Y (2017)	Text to table conversion has been provided by an EHR machine to convert details into a simple table. These details include smoking habits, drinking, some symptoms like headache, vomiting, chemical therapies recommended, doctor prescriptions etc.	The principal factor used for tabulation is a negative trigger of the mentioned details, which are tabulated using a modality identification method.	This methodology doesn't work well for classes with less frequency, mostly due to data in sufficiency. Later, no significant work has been performed about this concept.

				Since, the proposed methodology has a focus on NER at a great extent, it has been surveyed for improvisation.
7	"FACILE: Description of the NE system used for MUC-7. Authors: Black W J, Rinaldi(2018)	The handcrafted approach included a title followed by a capitalized word then it's a person's name, month name followed by a number less than 32 would be a year, from keyword followed by two dates with a 'to' in between would term as a range of dates, etc.	The most significant problem faced by that methodology was the expense of such a rule based and pattern based hand coding approach and reproducing the same results.	Each dataset or as such a domain will consist of its own characteristic literary genre like a newspaper report starts with a place name followed by the state name in India, reviews by people begin with a greeting, etc. This is one of the reasons for the large amount of work being done in this domain yet it is still a research topic and new algorithms still come into picture. With that, the various evaluation metrics play an equal role for betterment of the existing methodologies.
8	Named entity recognition and classification using context hidden markov model	The research work had already reached a remarkable level when Wang, laid out his work on a pre-trained bi-gram English language model with different entities	The HMM predicts names on the basis of the previous word and its category. Next is the Hybrid Model suggested as LTG , which is a hybrid approach of hand	A rule based approach was followed after training to ensure Date and time. Later in 2019 , came up with a

	Authors: Todorovic B T, Markovic(2019)	based on his mutual fund dataset. This gave similar accuracy as the decision tree model but with an added feature of reduced sentence error rate. One important solution was provided by the usage of large dictionaries was word segmentation method which was of sig nificant help for languages like Japanese where the words don't have spaces to dif ferentiate them.	coding and statistical approach of working, which is implemented stage wise.	unique way to manage less corpus for EMR with HMM. The new feature of the algorithm is that the HMM is mostly used to perform the word segmentation with elimination of certain suffix based words that are then trained which performs around 40% .
9	Multi objective approach for fea ture selection in maximum entropy based named entity recognition. Authors: Ekbal A, Saha S(2017)	This research mainly focuses on "past", "future" and "feature based" data. To eliminate time dependency during training, the MENE caches the data and uses it for future retrieval. It uses an advanced version of the iterative scaling algorithm that calculates the expected value and puts into categories based on the features.	An innovative method for enhancing the feature selection based on context and used for NER has been suggested using an algorithm and triggers like Information Gain, Mutual Information, etc. The existing MENE model till then was taken as a base to improve its efficiency by using rough set theory.	MUC 7 test corpus to train the then state of the art Maximum Entropy Named Entity (MENE) model with a 9 step training algorithm on C++.The algorithm begins with internal feature extraction followed by word segmentation and completed by feature expanding using word clusters but lexicons give better results. But, it lacks as all possible features haven't been observed Ugly Duckling theorem.
10	"Simner–an accurate and faster algorithm for named entity recognition.	The algorithm is based on a two-step process, starting with mining the rules and then using them NER with two additional	Three kinds of rules- dictionary, bigram and feature rules were considered. The recall and Precision obtained	a rule-based learning technique which utilises a 2step approach for NER namely

	Authors: Tripati S P, Rai H(2018)	parameters– support and confidence for each rule and a user defined threshold for its acceptance. The recall of the innovated model is comparable but the precision is better than the latter.	for English (66,83) are better than the one from Indonesian(62,81) news articles. a novel method of Entity Recognition (ER) using Association Rules that works at par with the MENE model.	finding out named entities and then classifying them. Regex based pattern recognition was performed to find entities with respect to WordNet a semantic dictionary with an improved version of K-means algorithm.
11	Named entity recognition from biomedical texts using a fusion attention-based bilstm-crf Authors:Wei H, Zhou A(2019)	Maximum Entropy Model (MEM) to tag initially and use that as a training method for HMM. Trained gazetteers are used in MEM model. The model gives impressive results and out performs both of its parent models. The only shortfall is the inability to cluster the fictional domain.	A unique method of two way distance evaluation called the Fuzzy Ed algorithm with many other novel techniques such as shrinking method to de crease computation, lower bound similarity for extraction, etc. The F-score lies between 0.91 and 0.97, with well-set thresholds at both levels.	The results are compared with other methods such as with the soft max layer instead of CRF and clearly show better f-measure scores. Other methodologies that came into picture include Leverag ing Linguistic Structures for NER with Bidirectional Recursive Neural Networks . A Fixed-size Ordinally Forgetting Encoding (FOFE)-based Local Detec tion Approach for NER and Mention Detection and Joint Decoding Algorithm.

Table 2.1 shows the comparison of recent papers that are surveyed related to text summarization. The paper [4], Summarization, based on the kind of input given to the model, the approachescan be categorized as supervised or unsupervised. Supervised methods for ex tractive summarization involve providing the human reference summaries in the training phase. ANN that are trained according to this method involve considerable human intervention duringthe training phase making it time consuming and cumbersome. Unsupervised methods don't require any reference summary, hence have an advantage over the supervised approaches in this aspect. This research largely focuses on the unsupervised graph-based approaches for extractive summarization, and how they can be improved by applying pre-trained word- embedding models. Supervised neural network models like CNN and RNN require huge quantities of custom-made document-summary pairs for a genre-specific/ text summarization task, making it out of scope of this paper. Figure 2.1 shows the Preprocessing steps which are used in extractive summarization.

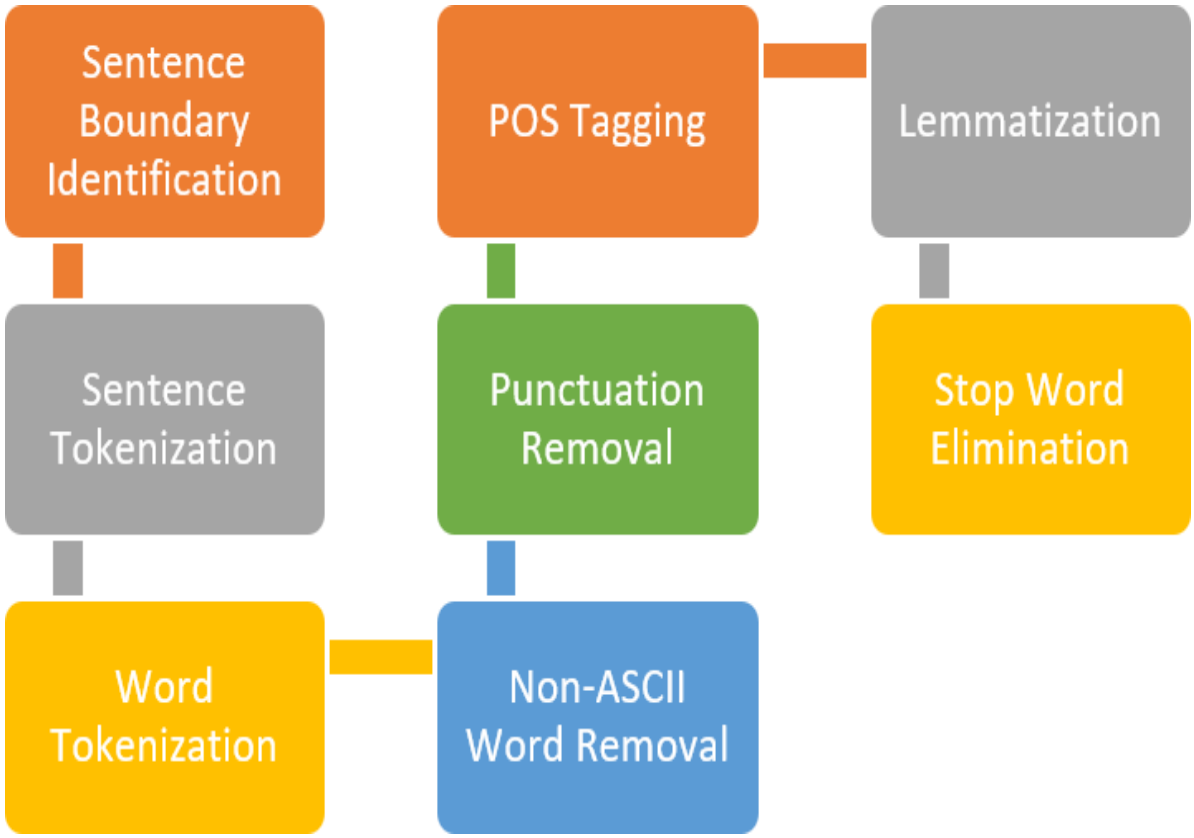

Figure2.1 Block Diagram of Preprocessing Steps

The paper [6], Medical text summarization is a crucial task in the healthcare domain, aimed at condensing large volumes of medical data into shorter, more manageable forms while retaining essential information. A system based on Named Entity Recognition (NER) and modality identification can enhance the efficiency and accuracy of this process.

The paper [11], Figure 2.2 shows the summarization using data acquisition, in Each dataset or such as a domain will consist of its own characteristic literary genre like a newspaper report starts with a place name followed by the state name in India, reviews by people begin with a greeting, etc. This is one of the reasons for the large amount of work being done in this domain yet it is still a research topic and new algorithms still come into picture. With that, the various evaluation metrics play an equal role for betterment of the existing methodologies.

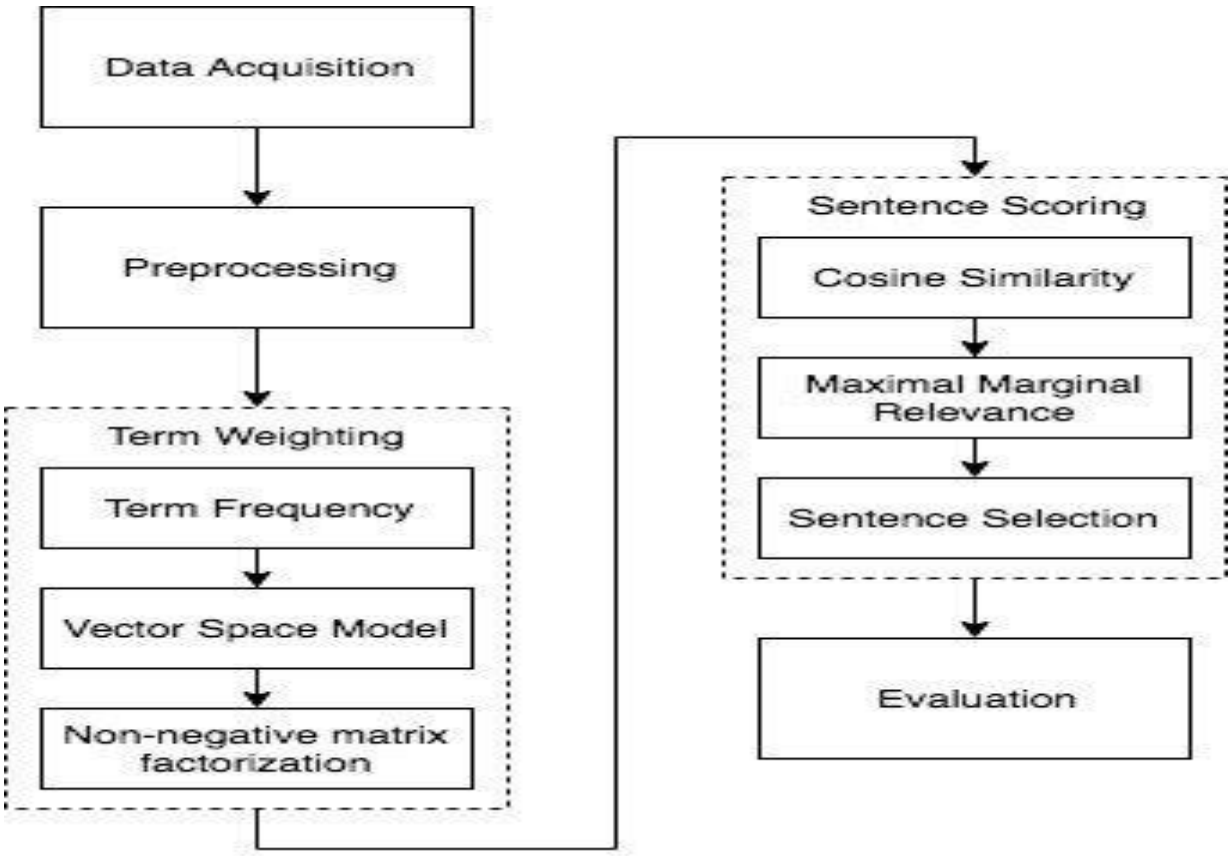

Fig 2.2 Block Diagram of Summarization using Data Acquisition

CHAPTER-3

METHODOLOGY

Since our research is aimed at presenting highly factual scholastic data in an easy-to absorb format, school history textbooks were considered most suitable. History books are generally loaded with events, years, places and people. The books published by National Council of Educational Research and Training (NCERT) , India are a rich source of scholastic content, as they are used in schools in India. These books (written in English) have been made available online at ncert.nic.in. The same have been used in NLP related research in the past (Sachan et al., 2018; Agrawal et al., 2014). There have been a few publications about document summarization in the field of E-learning such as (Wang et al., 2009; Baralis and Cagliero, 2015). But, as per our research, note making using NLP involving extractive summaries and tables to help the readers in revising important facts in scholastic domain were not found at the time of writing this paper. Hence, we propose a novel technique to achieve this. Towards this, four datasets were prepared from Class IX and Class X History text books written in English language, varying in length and genre to test our model on scholastic text and analyse how it works; used for research purposes only. All the four test datasets are text documents with passages ranging from 35 to 68 sentences. For brevity, they are categorized as Short (S), Medium (M1, M2) and Long (L) based on the size of the text document. We perform Extractive Summarization on each of these four documents. The topics of the datasets were chosen such that they belong to different genres like politics, sports and lifestyle. Of these, the dataset based on Indian Independence has been utilised for tabulation. Data Cleaning for Summarization Data cleaning generally involves removing the unnecessary data and punctuation and storing the data in the desired format. Here, only the textual part was required so phrases like “as shown in figure” and other references to images in the book were removed from the text. Additional informative content provided like New Words, Activity etc were also excluded. The exclusion of certain pages while extracting various levels of the category of the training dataset made from Wikipedia dump has to be carried out based on the use case with the help of Wikipedia api 1. Here, pages like movies, family details, memorials ,etc have been eliminated. Cutting off sections that might lead to misunderstanding of con text for the model is to be done, like references, publications, external links, lyrics, awards, films, etc. The conversion to the JSONL format, which is defined as the conversion of the key-value pair JSON format based on lines with two keys; namely text and source is essential for training the data for NER.

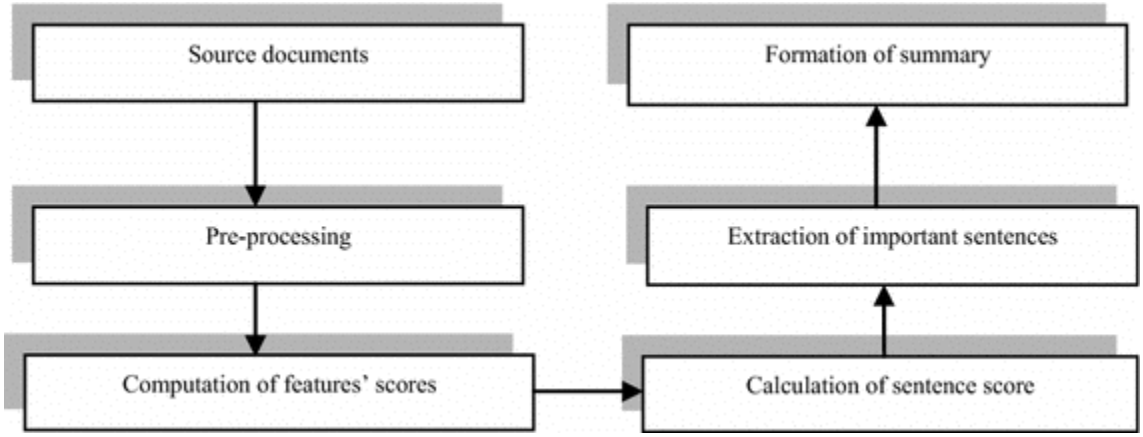

Fig3.1: Block diagram of Extractive summarization

Figure 3.1 describes about the methodology and work flow of the summarization. Starting with summarization, based on the kind of input given to the model, the approaches can be categorized as supervised or unsupervised. Supervised methods for extractive summarization involve providing the human reference summaries in the training phase. ANN that are trained according to this method involve considerable human intervention during the training phase making it time consuming and cumbersome. Unsupervised methods don't require any reference summary, hence have an advantage over the supervised approaches in this aspect. This research largely focuses on the unsupervised graph-based approaches for extractive summarization, and how they can be improved by applying pre-trained word-embedding models.

CHAPTER-4

SYSTEM DESIGN

4.1 FLOW CHART

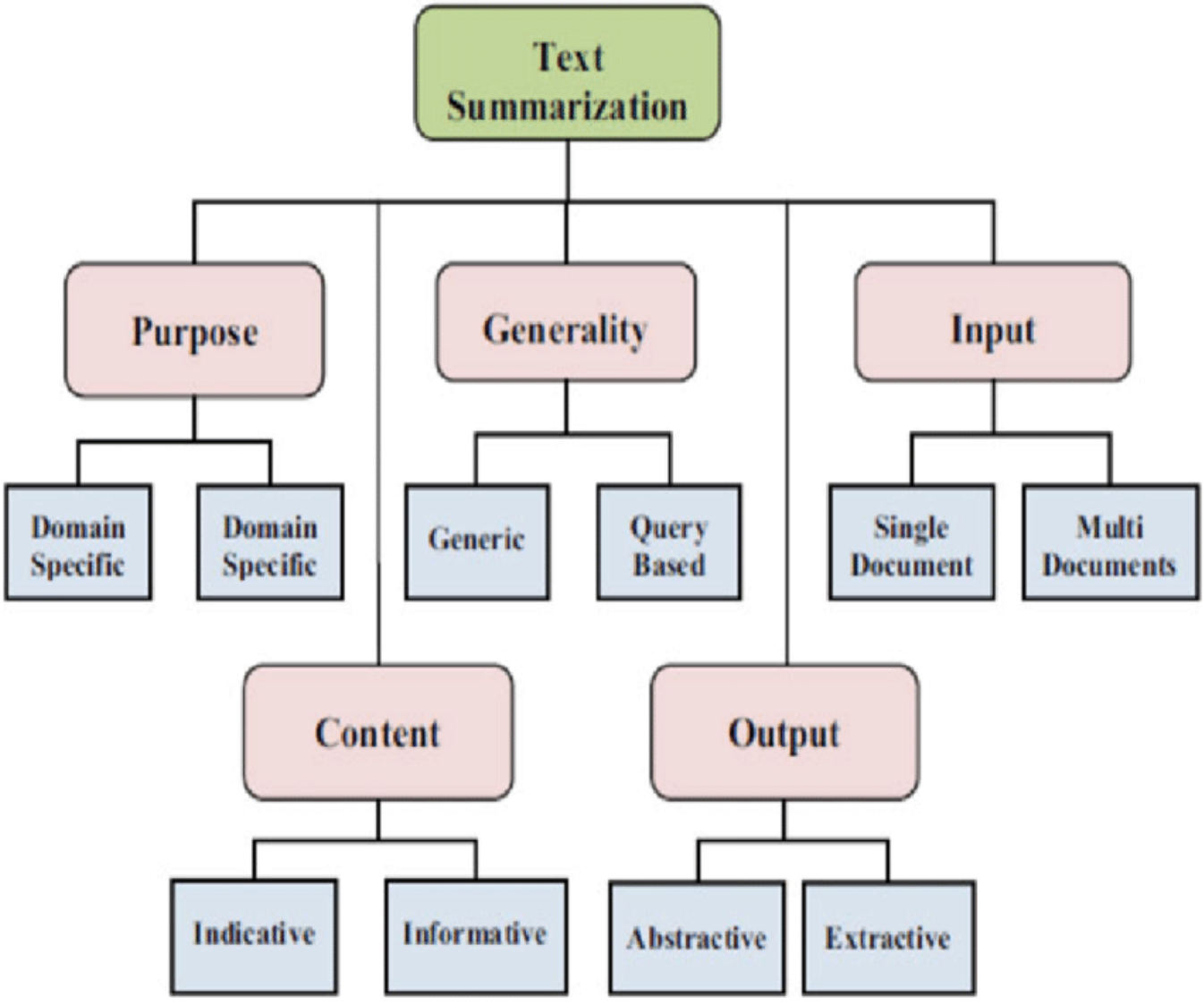

Fig4.1. Flow chart of Extractive Text Summarization

The aim of this research is to create a model which can take a document as the input and provide a summary containing important points from the text and a table containing facts and figures. In order to achieve this, the steps are shown in Figure 4.1. The input document is pre-processed and passed into the extractive summarization and tabulation modules. In summarization, ten different approaches are tested out on four input documents with differing lengths and genre. Since this is a relatively new domain for this kind of research, some common statistical and graph-based approaches are tested on the scholastic datasets where the performance of these algorithms are recorded based on the ROUGE evaluation metric. Some methods involving the use of pretrained models are also proposed in this research. Their performance is compared with existing algorithms; these well-established algorithms are implemented through open-source Python pack- ages. Now for tabulation, which is a novel technique, a methodology based on NER concepts is employed. The proposed model is adapted from the SpaCy 1 pipeline and further improvised with some additional steps. To train the model, the Wikipedia dump is used with the aid of wikipediaapi 2. For training, the pre-processed data is further processed using the JavaScript Object Notation Lines (JSONL) parser. The unlabelled data obtained is then used to train the model through an Annotation Tool called Prodigy incorporating two functions - correct and teach. The named entities thus obtained are cleaned, filtered, structured in a row-column format for the table representation.
Thus, the system proposed in this project produces a concise summary with points to remember from the text and a table that provides facts at a glance. For the evaluation of the model metrics like Recall, Precision and F1-score is used. For summarization, this is achieved using ROUGE, while, for tabulation, this is achieved through Recall and additional three comparison metrics specified .

4.2 CODE

```
import re
import string
import nltk
import heapq
import collections
from nltk.stem import WordNetLemmatizer

from nltk import sent_tokenize
from nltk import word_tokenize
from nltk.corpus import wordnet
```

```
#Downloading all the models and corpora required
nltk.download('punkt') nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')

from google.colab import files uploaded =
files.upload()

dataset = "CricketLong.txt" def
load_dataset(dataset):

file = open(dataset,'r',encoding='cp1252')
text = file.read()  #contains the dataset as a string
file.close()
return text
text = load_dataset(dataset)
  print("Raw Text:")
print(text)
original_sentences = sent_tokenize(text)

def only_ascii(word):
for ch in word:
if ord(ch)>127:
return False
return True

def initial_preprocessing(text):
punctuation = string.punctuation
cleaned_sentences = []
```

```
sentences = sent_tokenize(text)
for sentence in sentences:
word_list = ""
words = word_tokenize(sentence)
for word in words:
if(only_ascii(word)):
if word not in punctuation:
word_list = word_list + word.lower() + " "
cleaned_sentences.append(word_list)
return cleaned_sentences

  cleaned_sentences = initial_preprocessing(text)
print(cleaned_sentences)

  #POS Tagging
def get_mapping(first_char):
tag2first = dict()
tag2first = {
"J": wordnet.ADJ,
"N":   wordnet.NOUN,
"V":   wordnet.VERB,
"R":   wordnet.ADV
}
   return  tag2first.get(first_char, wordnet.NOUN)

#getting the wordnet based POS tag of the word to feed in the lemmatizer
def get_tagged_sentence(sentence):
sent_tag = []
pos_tag = nltk.pos_tag(sentence.split())
 #Sentence is split up into
```

```
words and passed to the tagger
for i in range(0,len(sentence.split())):
first = pos_tag[i][1][0].upper()
#first letter of the pos tag obtained
sent_tag.append(get_mapping(first))
return sent_tag


#Combining lemmatization with POS tagging for meaningful lemmatization with context
def sentence_lemmatization(cleaned_sentences):
 processed_sentences = []
lr = WordNetLemmatizer()
for sentence in cleaned_sentences:
lemmatized_sentence = ""
sent_tag = get_tagged_sentence(sentence)
i=0
for word in sentence.split():
root_word = lr.lemmatize(word,sent_tag[i])
lemmatized_sentence = lemmatized_sentence + root_word.lower() + " "
i=i+1
processed_sentences.append(lemmatized_sentence)
return processed_sentences
lemmatized_text = sentence_lemmatization(cleaned_sentences)
print("--------------------------After Lemmatization
---------------------------------")
print(lemmatized_text)
#creating a weighted histogram from the sentences
 word_count = {}
 wordList = []
 total_stopwords = nltk.corpus.stopwords.words('english')

 #obtaining word frequency
 for sent in lemmatized_tex
 for word in sent.split():
 if word not in total_stopwords:  #Stop-Word removal
 wordList.append(word)
```

```
 if word not in word_count.keys():
 word_count[word]=1
 else:
word_count[word]+=1


 sorted_wc = sorted(word_count.items(), key=lambda kv: kv[1], reverse= True)
 #printing word frequency table
 for item in sorted_wc:
 key, val = item
 if int(val)>0:
 print(key,"-",val)

#calculating word weights
 word_weights={}
 for key in word_count.keys():
word_weights[key]=word_count[key]/max(word_count.values()) #to find
relative frequency of the word
 print(word_weights)

#The sentence score will go up if the no. of frequent words in the sentence is the
more
sent_score = {}
sent_index = {}

count=0

for sentence in lemmatized_text:
 sent_index[sentence] = count
 for word in sentence.split():
```

```
 if word in word_weights.keys():
 if sentence not in sent_score.keys():
 sent_score[sentence]=word_weights[word]
 else:
 sent_score[sentence]+=word_weights[word]
count = count+1
#can be improved by removing wordy sentences

print(sent_score)

  best_sentences = heapq.nlargest(17,sent_score,key=sent_score.get)
  summary_indices=[]
  summary=''

  for sentence in best_sentences:
  summary_indices.append(sent_index[sentence])
  summary_indices.sort()

  for index in range(0,len(original_sentences)):
  if index in summary_indices:
  summary = summary + original_sentences[index] +'\n'
  print(summary)

  #writing the summary into a text file
  from google.colab import files

  filename = "SRL_WF.txt"

  with open(filename,"w") as f:
  f.write(summary)
 files.download(filename)
```

```
import os
import re
import string
import nltk
import numpy as np
 import networkx as nx
import matplotlib.pyplot as plt from nltk.corpus
import stopwords
from nltk.tokenize import sent_tokenize
from nltk.tokenize import word_tokenize
from nltk.stem.wordnet import WordNetLemmatizer
from sklearn.metrics.pairwise import cosine_similarity


nltk.download('punkt') # one time execution for sentence tokenization
nltk.download('stopwords')
nltk.download('wordnet')

!pip install PyDrive

import os
from pydrive.auth import GoogleAuth from
pydrive.drive import GoogleDrive from google.colab
import auth
from oauth2client.client import GoogleCredentials
auth.authenticate_user() gauth =
GoogleAuth()
```

```
gauth.credentials = GoogleCredentials.get_application_default() drive =
GoogleDrive(gauth)
download = drive.CreateFile({'id': 'google-drive-file-id'})
download.GetContentFile('glove.6B.50d.txt')
dim = 50

#dimension of the VxD matrix present in the pre-trained word_embedding file

glove_file = "glove.6B.50d.txt"
#loading the word embeddings file
def load_glove_model(glove_file):

word_vectors = {}

glove_data = open(glove_file,encoding='utf-8')

for line in glove_data:

values = line.split()

word = values[0]

#since the first column of the matrix represents words

vectors = np.asarray(values[1:],dtype='float32')

#float32 can represent upto 7 decimal places

word_vectors[word] = vectors

#storing the 100-dimension vector representation of each word in a dictionary

glove_data.close()

return word_vectors
word_vectors = load_glove_model(glove_file)

#Structure of the word_vectors dictionary-  { word1 : [50d word vector

representing word1], word2: [50d word vector representing word1]..}

print("Number of words:" + str(len(word_vectors)))  #V

x = next(iter(word_vectors))

print("Dimensionality:" + str(len(word_vectors[x])))
```

```
#D as in VxD matrix
  print("The first record in the dictionary:\n")
  print("KEY: " + str(x) )
print("VALUE: " + str(word_vectors[x]))

#Uploading the input document that is to be summarized
from google.colab import files
uploaded = files.upload()
#Reading the input document and storing the given passage as a string
This is the raw text that needs to be processed.

dataset = "CricketLong.txt"
def load_dataset(dataset):
file = open(dataset,’r’,encoding=’cp1252’)
text = file.read() #contains the entire text as a string
file.close()
return text
text = load_dataset(dataset)
print("Raw Text:")
print(text)
original_sentences = sent_tokenize(text)

 """
excluded_stopwords = set([’but’, ’if’, ’because’, ’as’, ’until’, ’ while’, ’against’, ’before’,
’after’, ’again’, ’all’, ’any’, ’both
’, ’each’, ’few’, ’more’, ’most’, ’other’, ’some’, ’no’, ’nor’, ’
not’, ’only’, ’own’, ’same’, ’so’, ’very’, ’until’])
"""
def only_ascii(word):
for ch in word:
if ord(ch)>127:
return False
return True
def preprocessing(text):
total_stopwords = set(stopwords.words(’english’))
#stopwords_list = total_stopwords - excluded_stopwords
```

```
punctuation = string.punctuation
processed_sentences = []



sentences = sent_tokenize(text)
for sentence in sentences:
word_list = ""
words = word_tokenize(sentence)
for word in words:
if(only_ascii(word)):
if word.lower() not in total_stopwords and word not in punctuation:
word_list = word_list + word.lower() + " "
processed_sentences.append(word_list)
return processed_sentences
processed_sentences = preprocessing(text)
print(len(processed_sentences))

#Obtaining sentence vectors using word vectors derived from the GloVe word
embedding

def vectorize_sentences(processed_sentences):
sentence_vectors = []
for sentence in processed_sentences:
no_of_words = len(sentence)
if no_of_words != 0:
#making sure the sentence is not empty
word_sum = sum([word_vectors.get(word, np.zeros(dim,)) for word in sentence.split()])
#if the word exists in the keys (in glove file), vector is obtained, else zeros matrix is
created
vec = word_sum/(no_of_words +0.001)
else:
vec = np.zeros((dim,))
sentence_vectors.append(vec)
return sentence_vectors
sentence_vectors = vectorize_sentences(processed_sentences)
print("Sample: First Sentence Vector :\n")
print(sentence_vectors[0])
```

```
#Forming Cosine Similarity Matrix
def get_similarity_matrix(processed_sentences):
sent_count = len(processed_sentences)
sim_mat = np.zeros([sent_count, sent_count])

#creating a symmetric zeros matrix
for i in range(sent_count):
for j in range(sent_count):
if i != j:
#cosine_similarity() takes two matrices,so reshaping list items to 1D vectors. The
resulting matrix is of 1x1. Storing that value in i,j position of sim_mat
sim_mat[i][j] = cosine_similarity(sentence_vectors[i].
reshape(1,dim),sentence_vectors[j].reshape(1,dim))[0,0]
sim_mat = np.round(sim_mat,4)
#rounding off upto three decimal places
return sim_mat
sim_mat = get_similarity_matrix(processed_sentences)
print(sim_mat)
def network_scores(sim_mat):
nx_graph = nx.from_numpy_array(sim_mat)
scores = nx.pagerank(nx_graph)
#Document Graph Visualization
pos = nx.spring_layout(nx_graph)
nx.draw(nx_graph, with_labels=True, font_weight='bold')
nx.draw_networkx_edge_labels(nx_graph,pos,font_color='brown')
plt.figure(figsize=(12, 12))
plt.savefig("TextRankGloVeGraph.png")
files.download("TextRankGloVeGraph.png")
plt.show()
return scores
scores = network_scores(sim_mat)
print(scores)
print(type(scores))
ranked_sentences = sorted(((scores[i],i) for i,s in enumerate( original_sentences)),
reverse=True)
print(ranked_sentences)
arranged_sentences = sorted(ranked_sentences[0:int(len( original_sentences)*0.25)],
key=lambda x:x[1])
print("FINAL SUMMARY:\n")
```

```
summary = "\n".join([original_sentences[x[1]] for x in
arranged_sentences]) print(summary)

#writing the summary into a text file from google.colab

import files

filename = "TR_Glove50d_Summary.txt"

with open(filename,"w") as f: f.write(summary)

files.download(filename)
```

CHAPTER-5

SYSTEM REQUIREMENTS AND SPECIFICATIONS

5.1 SOFTWARE REQUIREMENTS

Packages and Libraries:

• Natural Language Toolkit (NLTK)

• SpaCy

• Scikit-learn

• Pandas, NumPy

• NetworkX , Matplotlib

• Sumy

• Lexrank

• Bert-embedding

• Wikipediaapi

• Json

Tools Used:

• Data Turks

• Prodigy

Environment: Environments used for coding, testing and data visualization are:

• Google Colab

• JupyterNotebook

5.2 HARDWARE REQUIREMENTS

1. Processor Type: Intel(R) Core i7-8750H
2. RAM: 16GB DDR4 SDRAM
3. Windows 10 Pro 64-bit OS, with Nvidia GeForce GTX 1050Ti GPU

CHAPTER-6

RESULTS AND DISCUSSIONS

6.1 RESULTS

The Gold Standard Summaries and the gold standard table were created for the automatic evaluation of the two major tasks undertaken in this research: Summarization and Tabulation. Four documents were prepared to be provided as inputs to the summarization model To create the ideal summary for each of these documents, human-generated summaries were obtained. A group of 10 qualified individuals were asked to pick out the essential points from the passages, that they would want the summary to hold. So, 10 human-generated extractive summaries were used as a reference for accurate judging of the quality of each of the system-generated summaries. These peer summaries (reference summaries) act as gold standard summaries, and are compared with the system-generated summaries, referred to as system summaries. The system summary generated by each algorithm discussed in our paper, is compared against each of the ten reference summaries, for accurate results. We have used ten peer summaries for every document as there can't be one fixed gold standard summary for a document. Though this increases the number of comparisons, it leads to the accurate assessment of the algorithms. In terms of tabulation, a gold-standard table has been created manually for the specific text under testing which involves dimensions based on the type of dataset shown in the Table 6.1. The table has been cross-verified by taking opinions.

Table 6.1. Dimensions of Gold Standard Table

Dataset/Count	Number of words in testing set	Number of rows in table	Number of columns in table
History related dataset	5882	12	3

Now, the table created was sparse, so it had been worked on by choosing and understanding which information holds meaning when taken in conjugation with the other data in the same row. For example, DATE and PERSON isn't useful without its corresponding EVENT. Lastly, we have tried with different datasets involving content with artforms and writings without training the model. The results obtained are drastically low for the literary work based dataset. The recall measure falls down radically to 24%. The precision score is also not very commendable. These have been observed for the medium language model based tabular formation. By all the experiments performed and their comparison, the proposed algorithm gives notable results and decent scores for the novel dataset of E-learning that has been dealt here.

	DATE	PERSON	EVENT
0	1920s	Mahatma Gandhi	colonialism
1	1920-21	None	influenza epidemic
2	January 1915	None	satyagraha
3	1917	Mahatma Gandhi	satyagraha
4	1917	None	satyagraha
5	1918	Mahatma Gandhi	satyagraha movement
6	1919	Gandhiji	satyagraha
7	13 April	Dyer	Jallianwalla Bagh incident
8	summer of 1920	Gandhiji	civil disobedience campaign
9	December 1920	None	Non-Cooperation programme
10	January 1921	Swaraj	Non-Cooperation-Khilafat Movement
11	1921 and 1922	Baba Ramchandra	Non-Cooperation Movement
12	1921	Baba Ramchandra	Non-Cooperation Movement
13	1859	swaraj	Inland Emigration Act
14	1919	Mahatma Gandhi	Non-Cooperation Movement
15	1930	John Simon	Simon Commission
16	October 1929	Irwin	Round Table Conference
17	11 March	Mahatma Gandhi	civil disobedience campaign
18	6 April	swaraj	Non-Cooperation Movement
19	5 March 1931	Irwin	Gandhi-Irwin Pact
20	December 1931	Gandhiji	Round Table Conference
21	1932	None	Depression
22	1920	None	Chapter 5
23	1930	swaraj	Round Table Conference
24	September 1932	Gandhiji	Poona Pact
25	1927	Muhammad Ali Jinnah	All Parties Conference
26	twentieth century	Mahasabha	Chapter 1

Fig.6.1. Tabular Representation obtained using Trained + Rule Based model

6.2 DISCUSSION

In the discussion section , We use the ROUGE evaluation metric, to gauge the performance of the different algorithms discussed in this paper. Out of the various metrics proposed in the paper (Lin, 2004), ROUGE-1, ROUGE-2 and ROUGE-L are used for our research. ROUGE-1 measures the overlap of unigrams between the system summary and the reference summary. ROUGE-2 measures the overlap of bigrams between the system summary and reference summary. ROUGE-L is used to calculate the LCS that is common between the two summaries; it helps in measuring coherency and fluency of the summary. The ROUGE metric has the following sub components that is used to calculate the values mentioned above, namely, Precision, Recall and F-Measure or F1-Score. Ten different methods were tested on each of the four documents varying size and genre to draw conclusions on the method that works best the task under consideration. The ROUGE scores obtained as per the details mentioned . This provides inference on how a given algorithm works on scholastic content based on four different genres and sizes.

Table.6.2. Evaluation Result for different model based on three metrics

Table forming model / Evaluation Metric	**Column Retention**	**Row Retention**	**Order Retention**
Small Language model	30.53%	16.66%	Medium
Medium Language model	47.22%	33.33%	Medium
Large Language model	24.99%	8.3%	Low
Rule-based model	63.88%	66.66%	Slightly better than Medium
Trained Language model	55.52%	25%	Medium
Rule-based model+ Trained Language model	74.96%	58.33%	Medium

6.3 CONCLUSION

In conclusion, The goal of the research to make the note-making process effortless and manageable through clear and concise summaries and structured table formations. Towards this, various extractive summarization approaches were implemented on scholastic dataset, and their performance was analyzed using ROUGE evaluation metric. Based on experiments conducted, it can be concluded that the Word Frequency approaches (with and without Lemmatization) that uses the preprocessing routine recommended in this paper, is competent in extractive summarization of scholastic text. The highest scores recorded after the experiments, were 79.13% and 78.78% obtained through the word frequency approaches. The Modified TextRank with GloVe approach suggested in this paper works better than some of the popular python package-based implementations. The highest score obtained by using GloVe embedding was 75.98%. It is note-worthy that, using a Cased version of a pre-trained model proves to be more beneficial for this dataset, in comparison to the Uncased version, as observed in the case of the BERT pre-trained model. We can also conclude that a decent looking table can be created with the ground-breaking methodology provided. Slow training process was due to the cold-start problem which can be resolved by acquiring a lot of data in the particular domain with the various contexts of the entity in text with the same meaning. The named entities specified according to the domain are very dedicated to its corpus. After training with large and specific corpus, a recall value of 76.7%, precision value of 75.838% and F1-score of 74.056% has been acquired. The medium language model cascaded with rule-based learning provides best retention scores tested using table-slot comparison.

REFERENCES

1. Agrawal, R., Christoforaki, M., Gollapudi, S., Kannan, A., Kenthapadi, K., and Swaminathan, A. (2014). "Mining videos from the web for electronic textbooks." International Conference on Formal Concept Analysis, Springer. 219–234.

2. Aramaki, E., Miura, Y., Tonoike, M., Ohkuma, T., Mashuichi, H., and Ohe, K. (2009). "TEXT2TABLE: Medical text summarization system based on named entity recognition and modality identification." Proceedings of the BioNLP 2009 Workshop, Boulder, Colorado. Association for Computational Linguistics, 185–192.

3. Baralis, E. and Cagliero, L. (2015). "Learning from summaries: Supporting e-learning activities by means of document summarization." IEEE Transactions on Emerging Topics in Computing, 4(3), 416–428.

4. Barrios, F., López, F., Argerich, L., and Wachenchauzer, R. (2016). "Variations of the similarity function of textrank for automated summarization." arXiv preprint arXiv:1602.03606.

5. Black, W. J., Rinaldi, F., and Mowatt, D. (1998). "FACILE: Description of the NE system used for MUC-7" Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998.

6. Bölücü, N., Akgöl, D., and Tuç, S. (2019). "Bidirectional lstm-cnns with extended features for named entity recognition." 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), IEEE. 1–4.

7. Borthwick, A. E. (1999). "A maximum entropy approach to named entity recognition," PhD thesis, New York University. AAI9945252.

8. Brin, S. and Page, L. (1998). "The anatomy of a large-scale hypertextual web search engine." Proceedings of the Seventh International Conference on World Wide Web 7, WWW7, NLD. Elsevier Science Publishers B. V., 107–117.

9. Budi, I. and Bressan, S. (2003). "Association rules mining for name entity recognition." Proceedings of the Fourth International Conference on Web Information Systems Engineering, WISE '03, USA. IEEE Computer Society, 325.

10. Byrne, K. (2007). "Nested named entity recognition in historical archive text." International Conference on Semantic Computing (ICSC 2007), IEEE. 589–596.

11. Chan, S.-K. and Lam, W. (2007). "Efficient methods for biomedical named entity recognition." 2007 IEEE 7th International Symposium on BioInformatics and BioEngineering, IEEE. 729–735.

12. Chen, Y., Wu, Y., Qin, Y., Hu, Y., Wang, Z., Huang, R., Cheng, X., and Chen, P. (2019). "Recognizing nested named entity based on the neural network boundary assembling model." IEEE Intelligent Systems, 35(1), 74–81.

13. Chiong, R. and Wei, W. (2006). "Named entity recognition using hybrid machine learning approach." 2006 5th IEEE International Conference on Cognitive Informatics, Vol. 1, IEEE. 578–583.

14. Coates-Stephens, S. (1992). "The analysis and acquisition of proper names for robust text understanding. The analysis and acquisition of proper names for robust text understanding.

15. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). "Natural language processing (almost) from scratch." Journal of machine learning research, 12(Aug), 2493–2537.

16. Cruz, B. M. D., Montalla, C., Manansala, A., Rodriguez, R., Octaviano, M., and Fabito, B. S. (2018). "Named-entity recognition for disaster related filipino news articles." TENCON 2018-2018 IEEE Region 10 Conference, IEEE. 1633–1636.

17. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). "Bert: Pre- training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

18. Ekbal, A., Saha, S., and Hasanuzzaman, M. (2010). "Multiobjective approach for feature selection in maximum entropy based named entity recognition." Proceedings of the 2010 22nd IEEE International Conference on Tools with Artificial Intelligence - Volume 01, ICTAI '10, USA. IEEE Computer Society, 323–326.

19. Erkan, G. and Radev, D. R. (2004). "Lexrank: Graph-based lexical centrality as salience in text summarization." Journal of Artificial Intelligence Research, 22(1), 457–479.

20. Feng, C., Pan, Z., Zheng, J., and Xu, Y. (2018). "Memory-based extractive summarization." 2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE), IEEE. 549–552.

21. Gallo, I., Binaghi, E., Carullo, M., and Lamberti, N. (2008). "Named entity recognition by neural sliding window." Proceedings of the 2008 The Eighth IAPR International Workshop on Document Analysis Systems, DAS '08, USA. IEEE Computer Society, 567–573.

22. Han, X. and Ruonan, R. (2011). "The method of medical named entity recognition based on semantic model and improved svm-knn algorithm." 2011 Seventh International Conference on Semantics, Knowledge and Grids, IEEE. 21–27.

23. Ingole, M. N., Bewoor, M., and Patil, S. (2012). "Text summarization using expectation maximization clustering algorithm." International Journal of Engineering Research and

Applications, 2(4), 168–171.

24. Jiang, W., Guan, Y., and Wang, X.-l. (2006). "Improving feature extraction in named entity recognition based on maximum entropy model." 2006 International Conference on Machine Learning and Cybernetics, IEEE. 2630–2635.

25. Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). "Bag of tricks for efficient text classification." arXiv preprint arXiv:1607.01759.