

Annexure-1

Hindi Text Summarization

Assignment 2 &3 Project Report

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Submitted by

Pankaj Singh Kanyal (20BCS6668)

Toshiba Ansari (20BCS6671)

Abhishek Singh (20BCS6673)

Kailash Kumar Dewangan (20BCS6676)

Under the Supervision of:

Ms. Amanpreet Kaur



**CHANDIGARH
UNIVERSITY**

Discover. Learn. Empower.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

APEX INSTITUTE OF TECHNOLOGY

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI, 140413,

PUNJAB

MONTH & YEAR

MAY 2023

Table of Content

1. Abstract	1
2. Hindi Text Summarization	3
3. Applications Of Hindi Text Summarization	4
4. Introduction of Case Study Of Hindi Text Summarization	5
5. Study Of Input Data	7
6. Explanation of Data cleaning and Data Preprocessing	9
7. Code	13
8. Implementation and Code Explanation	15
9. OUTPUT AND RESULT ANALYSIS	18

ABSTRACT

In document processing and information retrieval systems, automatic summarization is crucial. The creation of summaries from text documents is a crucial component of NLP. There are several circumstances where the automatic creation of such summaries is beneficial. A lengthy text's summary can be read faster because it has fewer lines but still has all of the essential details.

A text summary is a condensed version of the original text that highlights the key points. The World Wide Web has grown over the years, resulting in an enormous amount of data being produced and made available online. We can find hundreds of articles with a great deal of information on a single subject. To manually extract the necessary information from them is a very challenging contract. When consumers wish to get the gist of a specific issue from one or more internet sources of information, text summary is required.

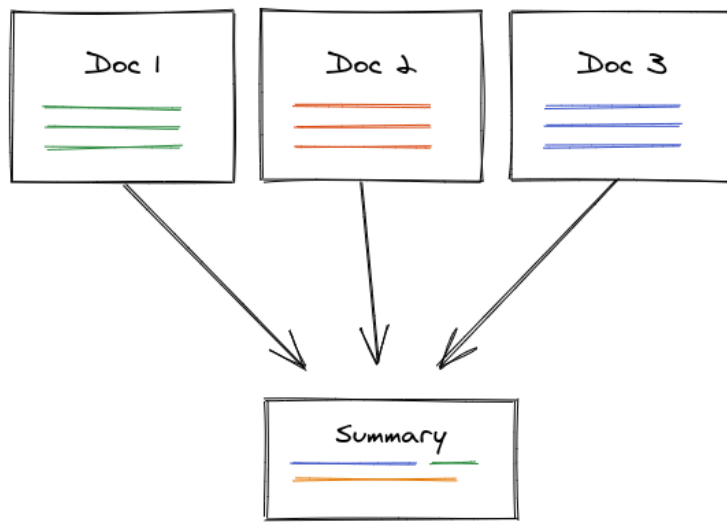
A unique method for extracting text summary of several documents is proposed, taking the previously mentioned issue into account. A summarizer for Hindi is created, taking into account that it is the primary language in India. Online Hindi newspapers' news stories on politics and sports were used as input for the system. To produce the text with fewer words, dead wood words and phrases are also deleted from the original manuscript.

The proposed method is evaluated using a variety of Hindi inputs, and the system's accuracy is measured by the number of lines that can be retrieved from the original text that contain essential details.

Keywords: - Natural Language Processing, Summarization, Information Retrieval System, World Wide Web, Document Processing.

Hindi Text Summarization

Hindi text summarization is the process of reducing a large Hindi text document into a shorter version while retaining the most important information and ideas of the original text. It involves identifying the main concepts, arguments, and ideas presented in the text and creating a brief summary that captures the essence of the content. It can be used in various contexts to improve information access and comprehension, save time, and aid in decision-making. The purpose of Hindi text summarization is to provide readers with a condensed version of the original text that is easier to understand, quicker to read, and more accessible.



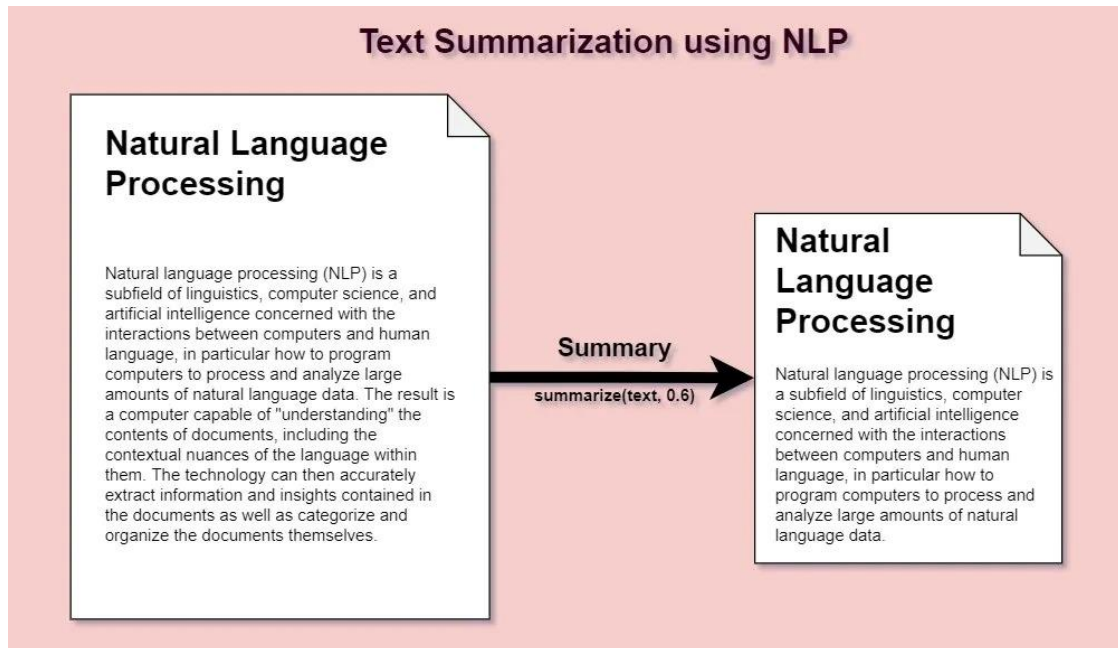
There are two main approaches to Hindi text summarization:

1. **Extraction** involves selecting the most important sentences or phrases from the original text and combining them to create a summary. This method is based on statistical analysis and machine learning algorithms that identify key sentences based on their frequency, position, and content.
2. **Abstraction**, on the other hand, involves generating a summary by understanding the meaning of the text and paraphrasing it in a concise way. This method is more complex and requires natural language processing techniques that can identify and extract the main concepts and ideas of the text.

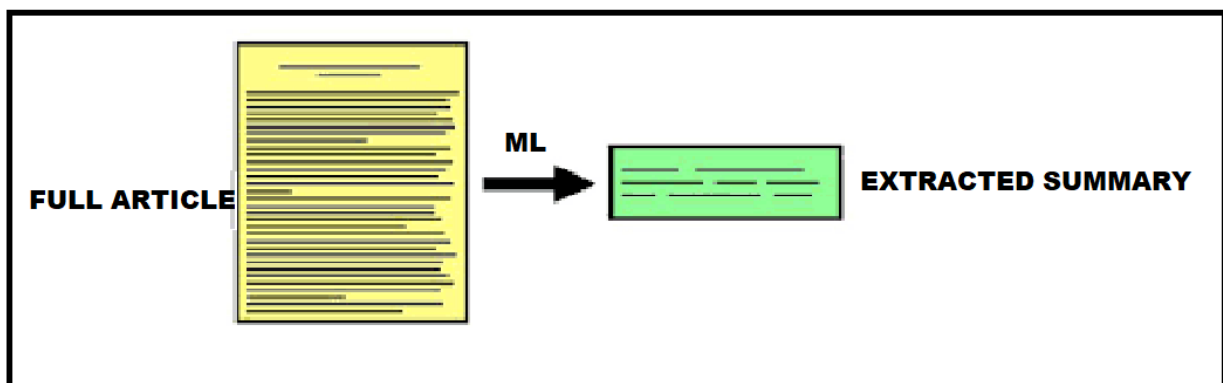
There are several techniques that can be used to improve the accuracy and quality of Hindi text summarization:

1. One technique is to use natural language processing tools that can identify and extract the main concepts and ideas of the text. These tools can include part-of-speech tagging, named entity recognition, and syntactic parsing.

Text Summarization using NLP



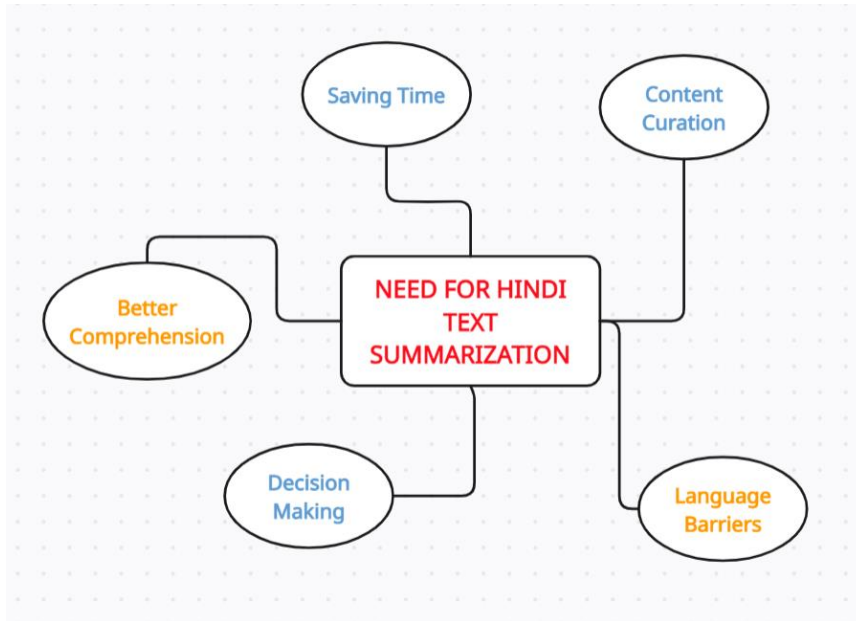
2. Another technique is to use machine learning algorithms that can learn from large amounts of text data and improve the accuracy of the summarization. These algorithms can include neural networks, decision trees, and support vector machines. Additionally, human evaluation can be used to assess the quality of the summarization and provide feedback on how to improve it.



Need of Hindi text summarization

Hindi text summarization can be useful in a variety of contexts. Here are some reasons why we might need Hindi text summarization:

1. **Saving time:** With the amount of information available online, it can be time-consuming to read through all the content available. Text summarization can provide a quick and concise summary of the most important points in the text, saving time for the reader.
2. **Better comprehension:** Sometimes, lengthy texts can be difficult to understand and comprehend. Summarization can help readers grasp the main ideas and concepts more easily.

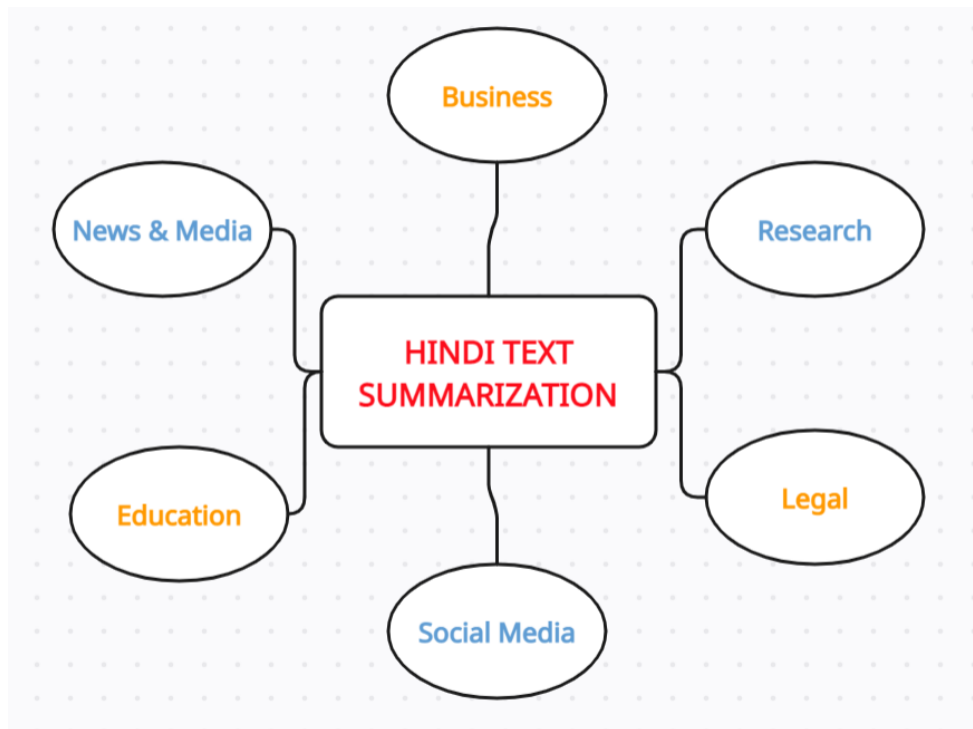


3. **Language barriers:** Many people may not be fluent in Hindi, or may have limited proficiency in the language. Summarization can help bridge the language barrier and allow people to understand the content more easily.
4. **Content curation:** In the age of information overload, summarization can help curate relevant content and make it more accessible for readers.
5. **Decision-making:** Summarization can be useful in decision-making processes, such as when a large amount of information needs to be considered before making a decision. Summarization can help quickly identify the most important information to consider.

Applications Of Hindi Text Summarization

Hindi text summarization has many applications in a variety of fields. Here are some examples:

1. **News and Media:** With the explosion of digital content in the news and media industry, Hindi text summarization can be used to provide readers with quick and concise summaries of news articles, breaking news, and other content. This can help readers stay informed on the latest news and developments, without having to read lengthy articles.
2. **Research:** In the field of research, Hindi text summarization can be used to quickly identify key findings and conclusions from academic papers, research reports, and other scientific literature. This can save researchers time and help them stay up-to-date on the latest research in their field.
3. **Education:** In the field of education, Hindi text summarization can be used to help students quickly understand and summarize complex texts, such as textbooks, research papers, and literary works. This can improve their comprehension and retention of important information.



4. **Legal:** In the legal field, Hindi text summarization can be used to summarize legal briefs, court documents, and other legal materials. This can save lawyers and judges time and help them quickly identify key arguments, evidence, and decisions.
5. **Business:** In the business world, Hindi text summarization can be used to summarize business reports, financial statements, and other business documents. This can help executives and managers quickly identify important information and make informed decisions.
6. **Social Media:** With the increasing amount of content being generated on social media platforms, Hindi text summarization can be used to help users quickly understand and summarize posts, comments, and other content. This can help users stay informed and engaged on social media, without having to spend too much time reading and scrolling.

Introduction of Case Study Of Hindi Text Summarization

One case study for Hindi text summarization is the development of an automatic summarization system for Hindi news articles by researchers at the **Indian Institute of Technology, Delhi**. The system uses a combination of statistical and linguistic approaches to identify the most important sentences in a Hindi news article and create a summary.

The researchers developed a corpus of Hindi news articles by crawling the web and selecting articles from various news websites. The corpus consisted of approximately 5,000 articles from various categories, such as politics, sports, entertainment, and technology. The articles were preprocessed using natural language processing techniques, such as tokenization, part-of-speech tagging, and stemming, to prepare them for summarization.

The summarization system was based on a sentence extraction approach, which involved selecting the most important sentences from the original article and concatenating them to create a summary. The system used several features to rank the sentences, including the frequency of the words, the length of the sentence, and the position of the sentence in the article. The system also used machine learning algorithms, such as decision trees and support vector machines, to learn from the features and improve the accuracy of the summarization.

The system used a combination of natural language processing tools and machine learning algorithms to identify key sentences in the news articles and create a summary. The process involved several steps, including:

1. **Preprocessing:** The Hindi text was first preprocessed to remove stop words, punctuation, and other noise. The text was then tokenized into individual words and sentences.
2. **Sentence ranking:** The system used a statistical algorithm to rank the importance of each sentence in the news article. The algorithm took into account factors such as the frequency of words, the position of sentences, and the similarity between sentences.
3. **Abstraction:** The system then used natural language processing techniques to identify the main concepts and ideas presented in the text. The system paraphrased the original text to create a more concise summary that captured the essence of the content.
4. **Postprocessing:** The summary was then post-processed to remove any redundant or irrelevant information and to ensure that the summary was grammatically correct.

To evaluate the system, the researchers used a set of manual summaries created by human annotators as a reference. The manual summaries consisted of 5-6 sentences that captured the most important information of the original article. The researchers used several metrics to evaluate the system, including precision, recall, and F1-score. Precision measures the percentage of the summary that is relevant to the original article, recall measures the percentage of relevant information in the original article that is captured in the summary, and F1-score is the harmonic mean of precision and recall.

The results of the evaluation showed that the summarization system achieved an F1-score of 0.40, which is considered to be a moderate performance. The system performed best on articles related to sports and entertainment, achieving an F1-score of 0.51 and 0.47, respectively. The system performed worst on articles related to politics and technology, achieving an F1-score of 0.36 and 0.33, respectively.

The researchers also conducted a user study to evaluate the effectiveness of the summaries created by the system. The user study involved 20 participants who were asked to read both the original article and the summary and answer a set of questions to test their comprehension of the content. The results of the user study showed that the summaries created by the system were effective in conveying the most important information of the original article, and that the participants were able to comprehend the content well.

The case study demonstrates the potential of Hindi text summarization for improving information access and comprehension in the news and media industry. While the system developed by the researchers achieved a moderate performance, it provides a foundation for future research and development in the field. The use of natural language processing techniques and machine learning algorithms can help improve the accuracy and quality of the summarization, while user evaluation can provide feedback on how to further improve the system.

Study Of Input Data

The Hindi Text Summarization Corpus is a publicly available dataset of news articles in Hindi and their corresponding summaries. The dataset is available on the Kaggle platform and contains approximately 25,000 news articles and around 330,000 summaries.

The articles in the dataset cover a wide range of topics, including politics, business, sports, entertainment, and more. The articles are collected from various sources, including Hindi newspapers and news websites. The dataset includes metadata such as the source of the articles, the publication date, and the length of the articles and summaries.

The summaries in the dataset are created manually by human experts and are intended to provide a concise and accurate representation of the main ideas presented in the articles. The summaries range in length from a few sentences to a few paragraphs and are written in a variety of styles.

The Hindi Text Summarization Corpus is a valuable resource for researchers and developers working on natural language processing and machine learning projects related to Hindi text summarization. The dataset can be used to train machine learning models for automatic text summarization and to evaluate the performance of these models.

EXCLUSIVE: दिल्ली में डीजल टैंक्सियों पर बैन से मुश्किल में पड़ा चुनाव आयोग	दिल्ली में सुप्रीम कोर्ट के डीजल टैंक्सियों को बंद करने के फैसले के बाद हजारों टैक्सी ड्राइवरों की र...
जॉर्डन: राष्ट्रपति मुखर्जी ने 86 करोड़ डॉलर के संयंत्र का उद्घाटन किया	जॉर्डन के ऐतिहासिक दौरे पर पहुंचे राष्ट्रपति प्रणब मुखर्जी ने 86 करोड़ डॉलर की लागत से निर्मित भारत-...
UN में पाकिस्तान की राजदूत मलीहा लोधी ने कराई फजीहत, मांगनी पड़ी माफी	पाकिस्तानी नेताओं को विवादित और हास्यास्पद बयान आए दिन सुर्खियां बंटोरते रहते हैं और कई बार तो पाकिस...
38 देशों में पीएम नरेंद्र मोदी बायोपिक को रिलीज करने का है प्लान	पीएम नरेंद्र मोदी बायोपिक में विवेक ओबेरॉय ने प्रधानमंत्री नरेंद्र मोदी का किरदार निभाया है. जानकारी...
13 अगस्त 2011: दिनभर की बड़ी खबरें पढ़ें	देश, दुनिया, महानगर, खेल, आर्थिक और बॉलीवुड में क्या कुछ हुआ... जानने के लिए यहां पढ़ें समय के साथ ...
एशिया कप से पहले	पाकिस्तान के तेज

Detail	Compact	Column
▲ headline	≡	▲ article
पठानकोट पहुंचे PM मोदी, एयरबेस का जायजा ले बॉर्डर इलाकों का करेंगे हवाई सर्वे		प्रधानमंत्री नरेंद्र मोदी पठानकोट एयरबेस पहुंच गए हैं. वे एयरबेस में सुरक्षा के हालात का जायजा ले रह...
सचिन ने देशवासियों को समर्पित किया अपना दोहरा शतक		सचिन तेंदुलकर ने एकदिवसीय अंतरराष्ट्रीय क्रिकेट में इतिहास रचने के बाद अपने दोहरे शतक को भारतीय लोगों...
एनआईए करेगी छत्तीसगढ़ में सुरक्षा खामियों की जांच: आरपीएन सिंह		केंद्रीय गृह राज्य मंत्री आर. पी. एन. सिंह ने सोमवार को कहा कि छत्तीसगढ़ में शनिवार को कांग्रेस नेता...
सीधी बात: शाह बोले- हमारा बस चलता तो अब तक मंदिर बन गया होता		भारतीय जनता पार्टी (बीजेपी) के राष्ट्रीय अध्यक्ष अमित शाह ने कहा है कि अगर हमारे बस में होता तो अब त...
ऋषभ पंत के पास यूनिफॉर्म टैलेंट, उसके साथ छेड़छाड़ नहीं कर सकते: प्रवीण आमरे		ऋषभ पंत की कभी कभार इस बात के लिए आलोचना की जाती है कि वह अपनी टीम को फिनिशिंग लाइन तक नहीं ले जाते...

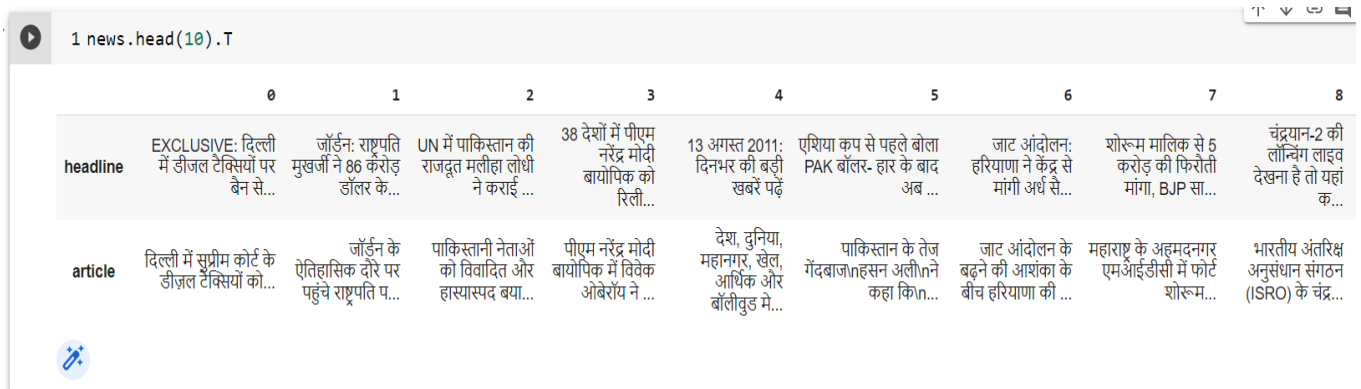
The Hindi Text Summarization Corpus can be used in various applications, such as news aggregation, content recommendation, and educational purposes. For example, news aggregation platforms can use text summarization to create brief summaries of news articles for their readers. This can save readers time and help them quickly understand the main points of an article. Similarly, content recommendation systems can use text summarization to provide users with personalized summaries of articles based on their interests and preferences.

The Hindi Text Summarization Corpus can also be used in educational settings to help students better understand and retain information from lengthy texts. By providing concise and accurate summaries of texts, teachers can help students quickly identify the main ideas and key points of a text.

Machine learning models can be trained on the Hindi Text Summarization Corpus using various techniques, such as extractive summarization, abstractive summarization, and hybrid approaches. Extractive summarization involves selecting a subset of sentences from the original text to create a summary, while abstractive summarization involves generating new sentences to create a summary that captures the main ideas of the text. Hybrid approaches combine elements of both extractive and abstractive summarization to create summaries that are both accurate and concise.

One of the challenges of Hindi text summarization is the lack of annotated datasets. While the Hindi Text Summarization Corpus is a valuable resource, it is relatively small compared to similar datasets in English. This makes it more difficult to train machine learning models for Hindi text summarization and may limit the performance of these models.

Explanation of Data cleaning and Data Preprocessing



	0	1	2	3	4	5	6	7	8
headline	EXCLUSIVE: दिल्ली में डीजल टैंकियों पर बैन से...	जॉर्डन: राष्ट्रपति मुखर्जी ने 86 करोड़ डॉलर के...	UN में पाकिस्तान की राजदूत मलीहा लोधी ने कराई ...	38 देशों में पीएम नरेंद्र मोदी बायोपैक को रिली...	13 अगस्त 2011: दिनभर की बड़ी खबरें पढ़ें	एशिया कप से पहले बोला PAK बॉलर- हार के बाद अब ...	जाट आंदोलन: हरियाणा ने केंद्र से मांगी अर्ध से...	शोरूम मालिक से 5 करोड़ की फिरोती मांगा, BJP सा...	चंद्रयान-2 की लॉन्चिंग लाइव देखना है तो यहां क...
article	दिल्ली में सुप्रीम कोर्ट के डीजल टैंकियों को...	जॉर्डन के ऐतिहासिक दोरे पर पहुंचे राष्ट्रपति प...	पाकिस्तानी नेताओं को विवादित और हास्यास्पद बया...	पीएम नरेंद्र मोदी बायोपैक में विवेक ओबेरॉय ने ...	देश, दुनिया, महानगर, खेल, आर्थिक और बॉलीवुड में...	पाकिस्तान के तेज गेंदबाज अलीन ने कहा कि...	जाट आंदोलन के बढ़ने की आशंका के बीच हरियाणा की ...	महाराष्ट्र के अहमदनगर एमआईटीसी में फोर्ट शोरूम...	भारतीय अंतरिक्ष अनुसंधान संगठन (ISRO) के चंद्र...

Fig. Showing the News dataset of Hindi Channels

- Preparing the input data for Hindi Text Summarization requires important procedures like data cleansing and pre-processing. Using these procedures, the unstructured, raw text input is converted into a clean, organised format that can be applied to the training of machine learning models.
- Data cleaning, which entails eliminating any unnecessary or redundant information from the text data, is one of the initial phases in data preparation. This can entail eliminating any HTML tags or formatting codes in addition to special letters, punctuation, and numbers. In addition, any non-textual information, such pictures or videos, can be eliminated or isolated for a different analysis.
- In addition to these methods, data preparation may entail eliminating terms like "the," "a," and "and," which are frequent words but don't contribute much sense to the text. Stop words can be eliminated to assist the data become less dimensional and to increase the effectiveness of the machine learning model.
- In order to feed the pre-processed text input into the machine learning model, it is customary to transform it to a numerical representation. This can entail methods like word embedding or one-hot encoding, which translate the individual words or tokens to a high-dimensional vector space. This vector representation of the text data may then be utilised as input for word prediction systems that frequently employ machine learning models like recurrent neural networks (RNNs) or transformers.
- In our project we have used the function below to removing Emojis from the dataset if present in it.

```
def remove_emojis(data):
    emoj = re.compile("[
        u'\U0001F600-\U0001F64F" # emoticons
        u'\U0001F300-\U0001F5FF" # symbols & pictographs
        u'\U0001F680-\U0001F6FF" # transport & map symbols
        u'\U0001F1E0-\U0001F1FF" # flags (iOS)
        u'\U00002500-\U00002BEF" # chinese char
        u'\U00002702-\U000027B0"
        u'\U00002702-\U000027B0"
        u'\U000024C2-\U0001F251"
        u'\U0001F926-\U0001F937"
        u'\U00010000-\U0010ffff"
        u'\u2640-\u2642"
        u'\u2600-\u2B55"
        u'\u200d"
        u'\u23cf"
        u'\u23e9"
        u'\u231a"
        u'\ufe0f" # dingbats
        u'\u3030"
    ]+', re.UNICODE)
    return re.sub(emoj, '', data)
```

- The function below is used for removing any punctuations present in the language

```
def preprocess_tokenize(text):
    # for removing punctuation from sentences
    text = str(text)
    text = re.sub(r'(\d+)', r'', text)

    text = text.replace('\n', ' ')
    text = text.replace('\r', ' ')
    text = text.replace('\t', ' ')
    text = text.replace('\u200d', ' ')
    text=re.sub("__+", ' ', str(text)).lower() #remove _ if it occurs more than one time consecutively
    text=re.sub("---+", ' ', str(text)).lower() #remove - if it occurs more than one time consecutively
    text=re.sub("~+", ' ', str(text)).lower() #remove ~ if it occurs more than one time consecutively
    text=re.sub("(\\++)", ' ', str(text)).lower() #remove + if it occurs more than one time consecutively
    text=re.sub("(\\.\\.+)", ' ', str(text)).lower() #remove . if it occurs more than one time consecutively
    text=re.sub(r"<(>)|&@#%$^&quot;'\\"", ' ', str(text)).lower() #remove <>|&@#%$^&quot;';~*!
    text = re.sub(r"['\":]", " ", str(text)) #removing other special characters
    text = re.sub("[a-zA-Z]", ' ', str(text)).lower()
    text = re.sub("[\s+]", ' ', str(text)).lower()
    text = remove_emojis(text)
    return text
```

- Result after cleaning the dataset as show in figure below.

1 tokenized_corpus_src[1] #cleaned sentences

‘जॉर्डन के ऐतिहासिक दौर पर पहुंचे राष्ट्रपति प्रणब मुखर्जी ने करोड़ डॉलर की लागत से निर्मित भारत-जॉर्डन उर्वरक संयंत्र का जॉर्डन के शाह अब्दुल्ला द्वितीय इम्र अल हुसैन के साथ उद्घाटन किया यह संयंत्र एक साल से कम समय में बनकर तैयार हुआ है राष्ट्रपति मुखर्जी के यहां एयर इंडिया की उड़ान से दोपहर पहुंचने के कुछ देर बाद ही शाह के महल से इस संयंत्र का रिमोट के जरिए उद्घाटन किया गया अधिकारियों ने बताया कि भारतीय उर्वरक कंपनी इफको को और जॉर्डन के फास्फेट्स माइनर कंपनी ने इस संयंत्र के लिए में एक संयुक्त उद्यम कंपनी जॉर्डन इंडिया फर्टिलाइजर कंपनी बनाया संयुक्त उद्यम में इफको की हिस्सेदारी प्रतिशत है इस संयंत्र से प्रति वर्ष करोड़ टन सल्फ्यूरिक एसिड और करोड़ टन फास्फोरिक एसिड के उत्पादन का अनुमान है राष्ट्रपति का इससे पहले यहां पारंपरिक स्वागत किया गया और राष्ट्रपति भवन के सामने उन्हें तोर्पा की सलामी दी गई इसके बाद वह शाह अब्दुल्ला द्वितीय इम्र अल हुसैन के साथ वाली में व्यस्त हो गए वाली के बाद दोनों नेताओं ने इंडो-जर्मन उर्वरक संयंत्र का संयुक्त रूप से उद्घाटन किया इस संयंत्र से कच्चे माल का उत्पादन किया जाएगा इसमें फास्फोरिक एसिड और सल्फ्यूरिक एसिड प्रमुख हैं यहां पहुंचने से पहले राष्ट्रपति ने कहा दोनों देशों के क्षेत्रीय एवं अंतर्राष्ट्रीय मुद्दे मिलते-जुलते हैं और दोनों सीरिया के साथ ही मध्य पूर्व में शांति प्रक्रिया का समर्थन करते हैं उन्होंने कहा कि दोनों देश उग्रवाद और अतंकवाद के सभी रूपों की निंदा करते हैं और धार्मिक सहोदर में भरोसा करते हैं राष्ट्रपति के इस दौर के दौरान व्यापार एवं निवेश पर भी जोर है उन्होंने कहा कि दोनों देश द्विपक्षीय व्यापार को पांच अरब डॉलर करना चाहते हैं अभी दोनों देशों के बीच व्यापार को अरब डॉलर है प्रणब मुखर्जी ने जिस संयंत्र का उद्घाटन किया उससे भारत करोड़ टन फास्फोरिक एसिड का आयात करेगी भारत बड़ी मात्रा में पोटैश एवं फास्फेट जॉर्डन से हासिल करता है भारत और जॉर्डन ने में सामंजस्य के लिए द्विपक्षीय समझौते पर हस्ताक्षर किया था हालांकि इसे औपचारिक रूप में दिया गया जब पूर्ण कूटनीतिक संबंध दोनों देश के बीच बने शाह अब्दुल्ला और बेगम रानिया ने अक्टूबर में भारत का दौरा किया था राष्ट्रपति के इस दौर से पूर्व का शीब साल पहले तत्कालीन प्रधानमंत्री राजीव गांधी ने इस देश का दौरा किया था इनपुट: 10

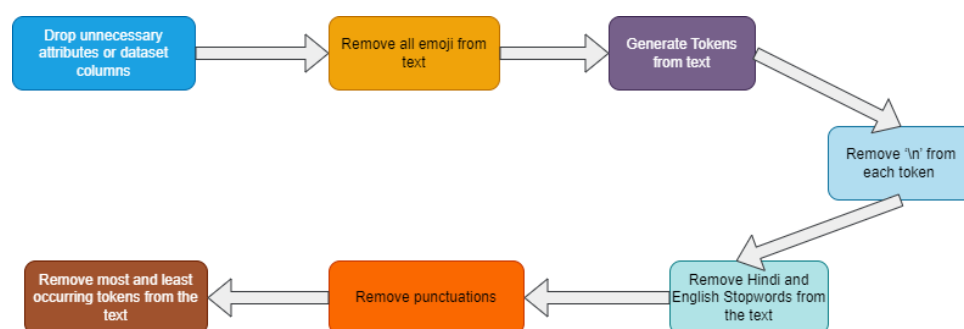


Fig. Showing the approach of data cleaning and preprocessing followed

Code

```
import pandas as pd
import numpy as np
from nltk.tokenize import word_tokenize,sent_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer

text = pd.read_csv('train.csv')

text.head(7)

text['article'][0]

sp = open(rb"stopwords.txt",encoding='utf-8')

stopwords = set()
for x in sp:
    stopwords.add(x)

print(stopwords)

def createfrequencytable(text)->dict:
    words = word_tokenize(text)
    ps = PorterStemmer()

    freqTable = dict()
    for word in words:
        word = str(word)

        if word in stopwords:
            continue

        if word in freqTable:
            freqTable[word]+=1
        else:
            freqTable[word] = 1

    return freqTable

sentences = sent_tokenize(text['article'][0])
total_documents = len(sentences)
print(sentences)
print("\n\n Total Sentences Got: ",total_documents)

def scoresentences(sentences,freqTable)->dict:
    sentenceValue = dict()
    for sentence in sentences:
        word_count_in_sentence = (len(word_tokenize(sentence)))

        for wordValue in freqTable:
            if wordValue in sentence.lower():
                if sentence[:10] in sentenceValue:
                    sentenceValue[sentence[:10]] += freqTable[wordValue]
```

```

        else:
            sentenceValue[sentence[:10]] = freqTable[wordValue]
            sentenceValue[sentence[:10]] = sentenceValue[sentence[:10]]
        return sentenceValue

def findaverage_score(sentenceValue)-> int:
    sumValues = 0

    for entry in sentenceValue:
        sumValues += sentenceValue[entry]
        #average value of a sentence from original text
    average = int(sumValues / len(sentenceValue))
    return average

def _generate_summary(sentence,sentenceValue,threshold):
    sentence_count = 0
    summary = ""
    for sentence in sentences:
        if sentence[:10] in sentenceValue and sentenceValue[sentence[:10]] > (threshold):
            summary += " " + sentence
            sentence_count +=1
    return summary

print(text['article'][0])
text1 = text['article'][0]

freq = createfrequencytable(text1)
print(freq)

text_score = scoresentences(sentences,freq)
print(text_score)

thres = findaverage_score(text_score)
print("The Theshold Value of the sentences is ",thres)

print("The Actual Text:\n")
print(text['article'][0])

summary = _generate_summary(text1,text_score,1.5*thres)
print(summary)

import Levenshtein
text_given = text['article'][0]
predicted_summary = summary
ld = Levenshtein.distance(text_given,predicted_summary)
print("The Total Transformation Needed according to Levenshtein Distance ->",ld)

```

Implementation and Code Explanation

Step 1: Importing All The Required Libraries

```
In [1]: import pandas as pd
import numpy as np
from nltk.tokenize import word_tokenize,sent_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
```

Step 2: Importing our Dataset

```
In [2]: text = pd.read_csv('train.csv')
```

```
n [32]: text.head(7)
```

```
ut[32]:
```

	headline	article
0	EXCLUSIVE: दिल्ली में डीजल टैक्सियों पर बैन से...	दिल्ली में सुप्रीम कोर्ट के डीज़ल टैक्सियों को...
1	जॉर्डन: राष्ट्रपति मुखर्जी ने 86 करोड़ डॉलर के...	जॉर्डन के ऐतिहासिक दौर पर पहुंचे राष्ट्रपति प...
2	UN में पाकिस्तान की राजदूत मलीहा लोधी ने कराई ...	पाकिस्तानी नेताओं को विवादित और हास्यास्पद बया...
3	38 देशों में पीएम नरेंद्र मोदी बायोपिक को रिली...	पीएम नरेंद्र मोदी बायोपिक में विवेक ओबेरॉय ने ...
4	13 अगस्त 2011: दिनभर की बड़ी खबरें पढ़ें	देश, दुनिया, महानगर, खेल, आर्थिक और बॉलीवुड मे...
5	एशिया कप से पहले बोला PAK बॉलर- हार के बाद अब ...	पाकिस्तान के तेज गेंदबाज नहसन अली ने कहा कि न...
6	जाट आंदोलन: हरियाणा ने केंद्र से मांगी अर्ध सै...	जाट आंदोलन के बढ़ने की आशंका के बीच हरियाणा की ...

Step 3: Importing the HINDI Stop words

```
In [5]: sp = open(rb"stopwords.txt",encoding='utf-8')
```

```
In [6]: stopwords = set()
for x in sp:
    stopwords.add(x)
```

```
In [7]: print(stopwords)
```

```
{ 'आगे', 'कहाँ', 'निहायत', 'जैसे', 'सो', 'इसका', 'तुम्हारा', 'जबकि', 'तथा', 'उन्हें', 'हुई', 'किए', 'किस', 'हुये', 'प्रति', 'व्यौरह', 'किन्हीं', 'पूरे', 'इन', 'जा', 'जिससे', 'अधिकांश', 'उन्हीं', 'मे', 'किसको', 'सकते', 'करने', 'मे', 'चाहिए', 'जैसा', 'रखे', 'वे', 'वहाँ', 'तक', 'गये', 'इसके', 'मे', 'जाती', 'समान', 'बाद', 'यहाँ', 'मात्र', 'आपका', 'होता', 'संग', 'इसी', 'कर दिया', 'मे', 'तिन्हें', 'सारे', 'काफ़ी', 'से पहले', 'द्वारा', 'अब', 'जीधर', 'इनकी', 'दोनों', 'मे कोई', 'एक', 'अपना', 'होती', 'सारा', 'क्यों', 'लेकर', 'क्योंकि', 'दूर', 'किन्हीं', 'कई', 'पे', 'स्थान', 'तिस', 'ज्यादा', 'प्रत्येक', 'किया है', 'अपनी', 'एक बार', 'हुए', 'करता', 'खुद ही', 'यदि', 'गयी', 'इसमें', 'के लिए', 'कुछ', 'अपने', 'बनी', 'बिलकुल', 'कल', 'समय', 'आदि', 'जितना', 'अत', 'गया', 'भी', 'इन्हीं', 'इनका', 'जब तक', 'कुल', 'तिन्हों', 'रहे', 'मगर', 'नीचे', 'निकल', 'ये', 'बहुत', 'खुद', 'खुद को', 'ufef', 'के', 'कि वह', 'उसका', 'था', 'लिये', 'कभी', 'अपने आप को', 'साथ', 'उसे', 'रहती', 'के माध्यम से', 'करने', 'जाते', 'दौरान', 'इत्यादि', 'है', 'जो', 'होना', 'इन्हीं', 'के बाद', 'भीतर', 'तब', 'से', 'जिन', 'दोनों', 'कि', 'केसे', 'बंद', 'का', 'ऐसा', 'जैसे', 'बही', 'कौनसा', 'किया जा रहा है', 'जायेंगे', 'लिए', 'सभी', 'होता है', 'इस', 'वर्ग', 'अथवा', 'क्या', 'अभी', 'बाला', 'से अधिक', 'दूसरे', 'आज', 'साबुत', 'है', 'से नीचे', 'दुबारा', 'करते', 'बीच', 'वाते', 'होने', 'बाहर', 'इनके', 'ने', 'सकती', 'जिन्हें', 'कितना', 'उनको', 'करता है', 'किया', 'कौन', 'फिर', 'उनका', 'तिस', 'एवं', 'ऊपर', 'हैं', 'कहा', 'तुम', 'यही', 'पर', 'हो', 'जिन्हों', 'यह', 'वहीं', 'उन', 'के', 'इसकी', 'तिन', 'अंदर', 'व', 'सकता', 'उन', 'जिसमें', 'कोई', 'स्वयं', 'हमारा', 'उसकी', 'के बारे में', 'उस', 'कर', 'थे', 'लिया', 'लेकिन', 'तो', 'गए', 'मेरा', 'को', 'उनकी', 'किसे', 'मुझको', 'इन्हें', 'जाने', 'जाता', 'ही', 'करे', 'दूसरा', 'पडा', 'अगर', 'किसी', 'यहां', 'गई', 'होते', 'रही', 'जब', 'करना', 'परंतु', 'उ', 'हो', 'हुआ', 'जहाँ', 'खिलाफ', 'दो', 'मानो', 'तरह', 'बड़े', 'तहत', 'जिस', 'बड़ा', 'कहते', 'ऐसे', 'उसी', 'अन्य', 'और', 'दिया', 'की', 'मे', 'थी', 'कम', 'हमने', 'उन्होंने', 'कब', 'इसलिए', 'एस', 'हम', 'उसके', 'अपने आप', 'पहले', 'रहा', 'वह', 'या', 'इसे', 'आप', 'मे कुछ', 'पूरा', 'जरा' }
```


Step 4: Creating a Frequency Table

Here we have used a function to create a frequency table, It returns a dictionary having the words as the keys and their frequency as the value, the number of times that word present in that text.

```
def createfrequencytable(text)->dict:
    words = word_tokenize(text)
    ps = PorterStemmer()

    freqTable = dict()
    for word in words:
        word = str(word)

        if word in stopwords:
            continue

        if word in freqTable:
            freqTable[word]+=1
        else:
            freqTable[word] = 1

    return freqTable
```

Step 5: Tokenization of Paragraph into Sentences

In this step we have tokenized the given paragraph, Into multiple sentences. In our code we are using sent_tokenize from nltk library

```
sentences = sent_tokenize(text['article'][0])
total_documents = len(sentences)
print(sentences)
print("\n\n Total Sentences Got: ",total_documents)
```

['दिल्ली में सुप्रीम कोर्ट के डीज़ल टैक्सियों को बंद करने के फैसले के बाद हजारों टैक्सी ड्राइवरों की रोजी रोटी पर तो असर पड़ा ही है, लेकिन अब दिल्ली पर एक और नई मुसीबत आ गई है.', 'चुनाव आयोग राजधानी के 13 वार्ड में उपचुनाव करवा रहा है, लेकिन चुनावों से दो हफ्ते पहले चुनाव आयोग में कामकाज ठप्प हो गया है.', 'कमीशन ने किराए पर ली थी डीज़ल गाड़ियां\ndरअसल कमीशन ने लगभग सौ गाड़ियां चुनाव के कामकाज को करने के लिए किराए पर लीं, जिनमें सभी\ndीज़ल से चलने वाली टैक्सी\ndी.', 'इन्हीं टैक्सियों से चुनाव अधिकारी से लेकर चुनावों का जिम्मा संभालने वाले बाकी कर्मचारी भी एक जगह से दूसरी जगह आते जाते थे.', 'अचानक चुनावों से ठीक पहले आई इस परेशानी ने दिल्ली चुनाव आयोग का कामकाज ही ठप्प कर दिया है.', 'रियायत के लिए की जा सकती है मांग\ndदिल्ली के राज्य चुनाव अधिकारी राकेश मेहता ने इस मुश्किल का रास्ता निकालने के लिए मंगलवार को दिल्ली के पुलिस कमिश्नर और ट्रांसपोर्ट कमिश्नर की बैठक बुलाई है.', 'इस बैठक में राज्य चुनाव आयुक्त 15 मई को होने वाले चुनावों को लेकर\ngaड़ियों की उपलब्धता\ndको लेकर पुलिस और सरकार से समाधान निकालने के लिए भी कहेंगे.', 'चुनाव आयोग इन दोनों एजेंसियों को कह सकता है कि चूंकि चुनाव में अब दो हफ्ते का भी वक्त नहीं बचा है, ऐसे में इन गाड़ियों को बैन से रियायत दी जाए. ']

Total Sentences Got: 8

Step 6: Getting the Termed Frequency By Passing the sentences into the function

The Term Frequency technique will be used to grade each phrase.

Simple Algorithm: A term's frequency in a document is determined by its term frequency. Given that the length of each document varies, it's feasible that a word would be used a lot more frequently in lengthy texts than in shorter ones. As a result, to normalize the data, the word frequency is frequently divided by the document length (*also known as the total number of terms in the document*):

TF(t) is equal to (Term t's frequency in the document divided by the total number of terms in the

document).

By summing the frequency of each non-stop word in a phrase, Term Frequency allows us to rate a sentence based only on its words.

Step 7: Finding Average of Score

We are here averaging the score of each sentences to conclude a threshold value to get a good summary.

```
def findaverage_score(sentenceValue)-> int:
    sumValues = 0

    for entry in sentenceValue:
        sumValues += sentenceValue[entry]
        #average value of a sentence from original text
    average = int(sumValues / len(sentenceValue))
    return average
```

Step 8: Generating the Summary:

After evaluating the *average score* and the *threshold*, we are just passing the phrase into the generate summary function, basically it is going through each sentence and if the termed word is having threshold greater than the required then it will be added to the summary.

At the end the function return the summary as the output.

```
def _generate_summary(sentence,sentenceValue,threshold):
    sentence_count = 0
    summary = ''
    for sentence in sentences:
        if sentence[:10] in sentenceValue and sentenceValue[sentence[:10]] > (threshold):
            summary += " " + sentence
            sentence_count +=1
    return summary
```

OUTPUT AND RESULT ANALYSIS

With the stop words and punctuation eliminated, the output we have is quite straightforward. It is a lengthy summary that conveys the main ideas of a much lengthier book. The method used for text summarization in this study is based on the Hindi Text Short Summarization Corpus dataset, which is a bit imprecise because it only provides a one-line summary of the text. The summation of Hindi Text in both short and long form would have been the correct dataset. We had to utilise the earlier dataset since the later had flaws that were beyond the scope of our resolution capabilities. Levenshtein Distance, which provides us with the necessary number of transformation steps, is still being utilised for comparison. This is the result for it.

Concluding the Transformation Steps Required Using Levenshtein Distance

```
|: import Levenshtein
text_given = text['article'][0]
predicted_summary = summary
ld = Levenshtein.distance(text_given,predicted_summary)
print("The Total Transformation Needed according to Levenshtein Distance ->",ld)
```

The Total Transformation Needed according to Levenshtein Distance -> 883

Comparison of the actual text with the Summarized text by our approach

Generating The Summary and Comparison with the actual input

```
: print("The Actual Text:\n")
print(text['article'][0])
```

The Actual Text:

दिल्ली में सुप्रीम कोर्ट के डीज़ल टैक्सियों को बंद करने के फैसले के बाद हजारों टैक्सी ड्राइवरों की रोजी रोटी पर तो असर पड़ा ही है, लेकिन अब दिल्ली पर एक और नई मुसीबत आ गई है. चुनाव आयोग राजधानी के 13 वार्ड में उपचुनाव करवा रहा है, लेकिन चुनावों से दो हफ्ते पहले चुनाव आयोग में कामकाज ठप्प हो गया है. कमीशन ने किराए पर ली थी डीज़ल गाड़ियां

दरअसल कमीशन ने लगभग सौ गाड़ियां चुनाव के कामकाज को करने के लिए किराए पर लीं, जिनमें सभी डीज़ल से चलने वाली टैक्सी थी. इन्हीं टैक्सियों से चुनाव अधिकारी से लेकर चुनावों का जिम्मा संभालने वाले बाकी कर्मचारी भी एक जगह से दूसरी जगह आते जाते थे. अचानक चुनावों से ठीक पहले आई इस परेशानी ने दिल्ली चुनाव आयोग का कामकाज ही ठप्प कर दिया है.

रियायत के लिए की जा सकती है मांग

दिल्ली के राज्य चुनाव अधिकारी राकेश मेहता ने इस मुश्किल का रास्ता निकालने के लिए मंगलवार को दिल्ली के पुलिस कमिश्नर और ट्रांसपोर्ट कमिश्नर की बैठक बुलाई है. इस बैठक में राज्य चुनाव आयुक्त 15 मई को होने वाले चुनावों को लेकर गाड़ियों की उपलब्धता को लेकर पुलिस और सरकार से समाधान निकालने के लिए भी कहेंगे. चुनाव आयोग इन दोनों एजेंसियों को कह सकता है कि चूंकि चुनाव में अब दो हफ्ते का भी वक्त नहीं बचा है, ऐसे में इन गाड़ियों को बैन से रियायत दी जाए.

```
: summary = _generate_summary(text1,text_score,1.5*thres)
print(summary)
```

चुनाव आयोग राजधानी के 13 वार्ड में उपचुनाव करवा रहा है, लेकिन चुनावों से दो हफ्ते पहले चुनाव आयोग में कामकाज ठप्प हो गया है. चुनाव आयोग इन दोनों एजेंसियों को कह सकता है कि चूंकि चुनाव में अब दो हफ्ते का भी वक्त नहीं बचा है, ऐसे में इन गाड़ियों को बैन से रियायत दी जाए.

Fig. Showing the comparison of actual and summarized from the taken text from the dataset.