A MINI PROJECT REPORT

On

MARATHI NEWS SUMMARIZER

Submitted in partial fulfillment of the requirement of
University of Mumbai for the Course

Natural Language Processing
In
Computer Engineering (VIII SEM)

Submitted By

Saail Ganesh	**117A1061**
Santosh Krishnan	**117A1065**
Nabeel Sarguroh	**117A1066**

Subject Incharge
Prof. Pranita Mahajan

Department Of Computer Engineering

SIES Graduate School of Technology

Nerul – 400706

UNIVERSITY OF MUMBAI

Academic Year 2019 – 20

Department of Computer Engineering
SIES Graduate School of Technology
Nerul – 400706

CERTIFICATE

This is to certify that the requirements for the project report entitled ‘**Marathi News Summarizer’** have been successfully completed by the following students:

Name	**Roll No.**
Saail Ganesh	117A1061
Santosh Krishnan	117A1065
Nabeel Sarguroh	117A1066

in partial fulfillment of the course Natural Language Processing in Computer Engineering (VIII SEM) of Mumbai University in the Department of Computer Engineering, SIES Graduate School of Technology, Nerul – 400706, during the Academic Year 2019 – 20.

Prof. Pranita Mahajan

Subject Incharge

Department of Computer Engineering
SIES Graduate School of Technology
Nerul – 400706

PROJECT APPROVAL

This project entitled “Marathi News Summarizer” by Saail Ganesh(117A1061), Santosh Krishnan(117A1065), and Nabeel Sarguroh(117A1066) are approved for the course Natural Language Processing in Computer Engineering (VIII sem) of Mumbai University in the Department of Computer Engineering.

Examiners:

1. _________________

2. _________________

Subject Incharge:

1. _________________

Date:

Place: Nerul

Department of Computer Engineering
SIES Graduate School of Technology
Nerul – 400706

DECLARATION

We declare that this written submission for Natural Language Processing mini project entitled "Marathi News Summarizer" represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission has not been taken when needed.

Project Group Members:

Saail Ganesh(117A1061)

____

Santosh Krishnan(117A1065)

____

Nabeel Sarguroh(117A1066)

____

Date:

Place:

Table of Contents

Abstract

Online news Articles have made it easier to read news on the go. However, summaries allow people to easily understand the contents of news article quickly. Providing brief summary or an excerpt allows readers to know what to expect from the news articles. With the headline, it would be convenient for the reader to read a brief about the article too. A summary is a type of text document that is produced by a much larger text or sometimes multiple texts, that conveys important information in the original text in a much shorter form. The goal of automatic news summarization is to provide an excerpt from the news article that covers the news article in fewer sentences and words. Languages such as Marathi don’t have resources such as automated summarization. Marathi news articles don’t come up with a summary. To generate a summary for news articles, we use a series of steps namely, Preprocessing and Processing Phase. Preprocessing phase represents Marathi text in a structured way. In Processing phase, different features determine the important sentences. Weights of features are determined to create a sentence score which allow us to sort sentences.

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Fundamentals

Marathi Language is a Devnagari language spoken predominantly by around 83 million people in Maharashtra and some parts of Goa, India. People living in rural parts of these states communicatte only via Marathi. People here get their news from Marathi news sources and articles. However, summaries for these articles need to be created under human supervision. Brief summaries allow users to read news in a brief format. News Summarization deals with reducing the source text into a short version preserving its contents and overall meaning. Extraction of useful sentences from a news article is a very time consuming task. This is where Natural Language Processing helps us to create news summaries very quickly. Summarization occurs in two phases: Preprocessing phase, Processing phase.

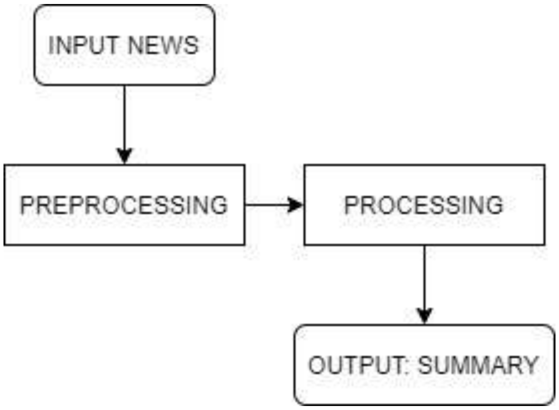


Fig 1.1: Flow of Project

The Figure 1.1 shows us the flow of our project. The Marathi news is preprocessed and then processed by out text rank algorithm to give us a summary as an output.

1.2 Objectives

The project intends to generate a summary from news document and present it to the user.

- Creation of Corpus using WebScrapping

- Creation of Stop Word List
- Tokenization and Stemming of words in our news article
- Creating a text rank algorithm that gives us highest ranked sentences for our summary

1.3 Organization of the Report

The Report is organized as follows:

The introduction is given in Chapter 1. It describes the fundamental terms used in this project. It motivates to study and understand the different techniques used in this work. This chapter also presents the outline of the objective of the report.

The Chapter 2 describes the review of the relevant various techniques in the literature systems. It describes the pros and cons of each technique.

The Chapter 3 presents the Theory and proposed work. It describes the major approaches used in this work.

The technical applications are mentioned in Chapter 4.

The summary of the report is presented in Chapter 5.

Chapter 2

Literature Survey

2.1 Introduction

The Different Techniques and Algorithms are needed to be used for our Project. Conducting a research allows us to examine various techniques that would be necessary for our project. We have studied various papers to find out preprocessing techniques that are efficient. We also studied algorithms that are neccesary for extraction of sentences.

2.2 Literature Review

In this Literature, Researchers have talked about Automatic Text Summarization for Regional Languages in India. We use Sentence Relative Length to find summaries. This allows very short sentences to be discarded. We find Keywords from the document that would be helpful in deciding the sentence importance. Number Feature allows us to rank sentences based on Numeric data. The sentences that contains number are usually important.

To handle the linguisic part of our project, a research suggests to separate words out into Morphemes and handle processing of language. The difficulty depends greatly on the complexity of the morphology of the language considered. Applying inflectional rules allows us to categorise the sentences to their root words.

Tokenization and Stemming of the document can be handled by using lexical analyser. These separate tokens are called as lexicons. Stemming removes the suffixes of the word and gives us a root word. Marathi is a morphologically rich language. Variants of words in Marathi are usually formed by adding suffixes to root word.

After preprocessing, the research moves on to summarization of the news article. Graph based Ranking algorithms are essentially a way of deciding the importance of a vertex within a graph based on global information recursively drawn on the entire graph. The higher the importance of the node, the higher the score.

2.3 Literature Summary

SN	Paper Name	Author and Year of Publication	Key Points
1.	A Survey of Automatic Text Summarization for Different Regional Language in India	Vivat Giri, Dr. M.M. Math , Dr. U.P. Kulkarni, 2016	Important Keyphrases, Numeric Values to be considered for sentence score
2.	Text Summarization using Rich Semantic Graph for Marathi Sentences	Sheetal Shimpikar, Sharvari Govilkar, 2018	Addition of Tokenization and Stemming rules.
3.	Discovering Suffixes: A case study for Marathi Language	Mudassar Majgaonker, Tanveer Siddiqui, 2010	Finding and removal of suffixes using various rules.
4.	NLP tasks for Marathi Language	Pratiksha Gawde, Deepika Madhavi, 2013	Morphological Anlysis for Marathi Language
5.	TextRank: Bringing Order to Ranks	Rada Mihalcea, Paul Tarau, 2013	Text Rank Algorithm using nodes and graphs.

Table 2.1 Literature survey summary

Chapter 3

Implementation Details

3.1 Overview

The project works mainly on two phases. Preprocessing phase and Processing phase. Preprocessing phase consists of Tokenization, Stemming, Removal of Stop words. Processing phase consists of implementation of a Textrank algorithm for marathi language. The inspiration behind this project is to generate sentence scoring for Marathi language based on position of word in the sentence.

3.1.1Existing Methodology and Systems

In Existing Systems the classification is done by a approach in which it would be count the no. of times a word appears in a document. So, the word's importance is directly proportional to its frequency. This method is, therefore, called Term Frequency(TF).

$$Term\ Frequency(t) = \frac{number\ of\ times\ t\ appears\ in\ a\ document}{total\ number\ of\ terms\ in\ the\ document}$$

Fig. 3.1. TF-IDF

This method fails in practical use as words like stopwords, non stemmed words and special characters etc. will almost always be the result of this method, as they occur more frequently. But of course, they are not the right way to summarize our document.

3.1.2 Proposed Methodology and System

As we need to overcome the drawbacks faced in the Existing System , we are proposing an approach in which we will take into consideration how unique the words are. Given a news article, find out the summary of the article based on sentence scores.

The news article is first given to our tokenization algorithm. Tokenization removes any unnecessary special characters such as brackets, commas, fullstops etc. Some news articles may have English or any other Language words in it. We don’t process those words as those words are very few in quantity and also have significant meaning to them. After Tokenization of sentences, we remove any stopwords from the tokens. Stopwords are type of words which don’t add meaning to the sentence. They can be ignored safely without sacrificing the meaning of the sentence.

After removing these stopwords, we send the sentence to our stemmer. Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming is important in natural language understanding (NLU) and natural language processing (NLP). Stemming is a part of linguistic studies in morphology and artificial intelligence (AI) information retrieval and extraction.

After Stemming we get a list of root words that are important to form a sentence. Now we move onto our TextRank Algorithm. For finding the top keyphrases, we plan to use Graph-based Algorithms, where each node is considered as a word from our stemmed dictionary. Graph-based ranking algorithm is a way of deciding on importance of a vertex within a graph, by taking into account global information recursively computed from an entire graph, rather than relying on local vertex-specific information. To give each node a weight we find the maximum word score of the node. This is done by

$$\text{score}= \ 1/(\pi * \sqrt{}(\text{pos}*(1-\text{pos}))$$

Where,

$$\text{pos}=\text{word\_position}/\text{len}(\text{document})$$

The maximum score for that word is used as the weight of the node. After node creation, we add edges to our graph. The addition of a window helps us to find the nearest words to the given words. Consider a window size of ‘n’. A word will form an edge with the next ‘n’ edges and the weight of an edge would be average of the weight of the nodes. After forming a graph, we apply pagerank. It implements a random surfer model where a node with probability ‘d’ is selected and jumps to completely new probability ‘1-d’. We get the top keyphrases after applying page rank. These are then added to each sentence score where they are present. The sentences with the top score are returned as they create a summary.

3.2 Implementation Details

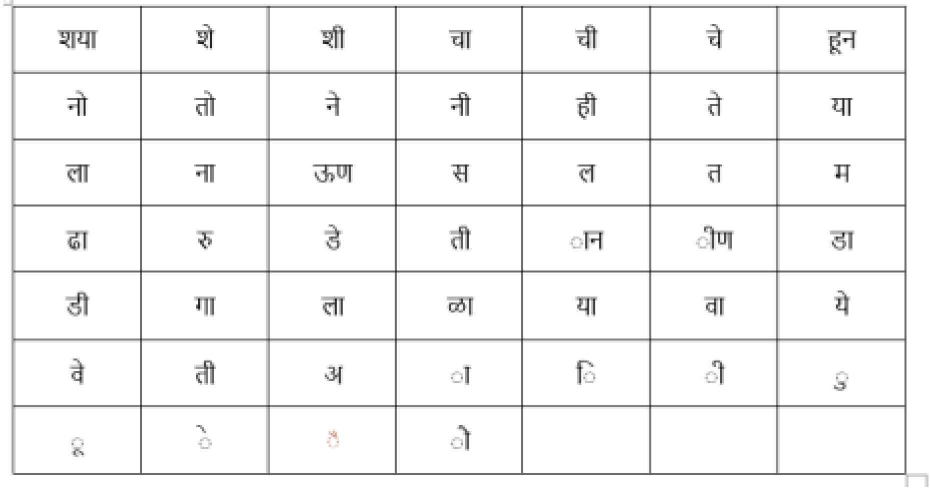
For our implementation of our system, we needed a corpus. We used webscrapping for our corpus. We scrapped Marathi News Websites and cleaned them for their tags. Usually on these websites there is an excerpt given which is the first few lines of the articles. We plan to replace that we an extractive summary that would be generated from our corpus.

3.2.1 Methodology

The article is then given for Tokenization. Each sentence is tokenized into words and we keep a track of these sentence. Now for each sentence we remove stopwords.

Stopwords are removed from a stop word list. We create a stop word list from our corpus. We tokenize our corpus into words. Now we find the most frequent words from the tokenize wordlist. These words are appended to our stopword list. Stopwords are type of words which don’t add meaning to the sentence. They can be ignored safely without sacrificing the meaning of the sentence. After removing stopwords, we pass these words to a stemmer. Stemming is done by removing suffixes from the words and then checking whether the words are in a wordlist. This wordlist contains words and relates them to their stems and is provided by IIT Bombay. For removal of suffixes, we check them with the suffixes normally used in Marathi language.

For Example, Words such as आनंदादितीशयाने is first removed with suffix ने and then is recursively stripped of suffixes, शया, ती, दि and finally is removed of a Maatra ा to get a root word as आनंद.



शया	शे	शी	चा	ची	चे	हून
नो	तो	ने	नी	ही	ते	या
ला	ना	ऊण	स	ल	त	म
ढा	रु	डे	ती	ान	ीण	डा
डी	गा	ला	ळा	या	वा	ये
वे	ती	अ	ा	ि	ी	ु
ू	े	ै	ो			

Table 3.1. Stemming Suffixes

Table 3.1 shows different suffixes which we remove for finding the stemmed word.

After stemming the word, the remains of every sentence are stored in a dictionary and are used for further processing. Meanwhile, we store sentences as a separate entity. We also find the total number of stemmed words in our dictionary. To find the weight of each node, we find the word score using the formula. For every word, we find the maximum of these node weight and assign it to the node hash.

Graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. The basic idea im plemented by a graph-based ranking model is that of "voting" or "recommendation". When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Moreover, the importance of the vertex casting the vote determines how important the vote itself is, and this information is also taken into ac count by the ranking model. Hence, the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices cast ing these votes.

Formally, let G (V, E) be a directed graph with the set of vertices V and set of edges E, where E is a subset of V x V. For a given vertex Vi, let In(V) be the set of vertices that point to it (predecessors), and let Out (V) be the set of vertices that vertex V; points to (successors).

The task of a keyword extraction application is to au tomatically identify in a text a set of terms that bes describe the document. Such keywords may consti tute useful entries for building an automatic index fo a document collection, can be used to classify a text or may serve as a concise summary for a given doc ument. Moreover, a system for automatic identifica tion of important terms in a text can be used for the problem of terminology extraction, and construction of domain-specific dictionaries.

The simplest possible approach is perhaps to use a frequency criterion to select the "important" key words in a document. However, this method was generally found to lead to poor results, and conse quently other methods were explored. The state-of the-art in this area is currently represented by super vised learning methods, where a system is trained to recognize keywords in a text, based on lexical and syntactic features.

For adding edges to the graph, we use windows. Each window is of certain size and are the total number of edges added at a time. For example, Suppose the stemmed dictionary contains words:

विद्यार्थ्यां, ऑनलाईन, शिक्षणातून, अखेर

Now for word विद्यार्थ्यां, we add an edge from विद्यार्थ्यां to ऑनलाईन, शिक्षणातून, अखेर. These edges would have weight as an average of weights of both nodes.

After applying Textrank, we get keyphrases for the document. For every article with 'n' stemmed words, we take the minimum of 0.1*n or 7*log(n). This allows normalization of total stemmed words. If these keyphrases are present in the stemmed dictionary, we add the weight to sentence score. The sentence with the highest scores are returned as an output.

3.2.1 Details of packages, data set

The dataset which we used in our system was webscrapped from abpmajha.com. The wordlist for stemming was provided by the Center for Indian Language Technology (CFILT) which was setup by Department of Information Technology (DIT), Ministry of Communication and Information Technology, Government of India in 2000 at the Department of Computer Science and Engineering, IIT Bombay.

1. collections: Collections in Python are containers that are used to store collections of data, for example, list, dict, set, tuple etc. These are built-in collections.

2. io: Python io module allows us to manage the file-related input and output operations. The advantage of using the IO module is that the classes and functions available allows us to extend the functionality to enable writing to the Unicode data.

3. re: This module provides regular expression matching operations similar to those found in Perl.

4. nltk: NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries. We would be using sent_tokenize.

5. pandas: Pandas is a software library written for the Python programming language for data manipulation and analysis.

6. numpy: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices

7. Matplotlib.pyplot: Matplotlib is a plotting library for creating static, animated, and interactive visualizations in Python.

8. operator: The operator module exports a set of efficient functions corresponding to the intrinsic operators of Python. For example, operator.add(x, y) is equivalent to the expression x+y. Many function names are those used for special methods, without the double underscores.

9. math: Python has a built-in module that you can use for mathematical tasks.

10. NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. By definition, a Graph is a collection of nodes (vertices) along with identified pairs of nodes (called edges, links, etc). In NetworkX, nodes can be any hashable object e.g., a text string, an image, an XML object, another Graph, a customized node object, etc.

Chapter 4

Project Inputs and Outputs

4.1 Inputs Details

Each input is a news document which can vary from about 15 lines to 50 lines, depending on the context. Each article has a headline.

Size of Corpus: 120 articles.

Number of stem/ related words in wordlist: 56000

Number of stopwords: 200

Input Article for Processing:

शाळांना 1 मे पासून सुट्ट्या जाहीर, 14 जूनपासून नवे शैक्षणिक वर्ष सुरू होणार.

मुंबई : शिक्षकांना, विद्यार्थ्यांना ऑनलाईन शिक्षणातून अखेर विश्रांती देण्यात आली आहे. राज्यातील प्राथमिक, माध्यमिक, उच्च माध्यमिक शाळांना 1 मे पासून सुट्ट्या जाहीर करण्यात आल्या आहे. 1 मे ते 13 जून दरम्यान शाळांना सुट्टी राहणार आहे. तर 14 जूनपासून नवे शैक्षणिक वर्ष सुरू होणार आहे. राज्यातील शिक्षक संघटनाकडून वारंवार केल्या जात असलेल्या मागणीनंतर शाळांना सुट्टी जाहीर करण्यात आल्या आहेत. विदर्भातील तापमान विचारात घेता 28 जूनपासून शाळा सुरू होणार आहेत.

कोरोनाच्या पार्श्वभूमीवर मागील वर्षभरापासून शाळा जरी अनेक ठिकाणी बंदच राहिल्या तरी ऑनलाइन वर्ग सुरू होते. ऑनलाइन शिक्षण देऊन अभ्यासक्रम यावर्षी शिक्षकांनी पूर्ण केला. मात्र, कोरोनाची राज्यातील सद्यस्थिती नुकताच इयत्ता पहिली ते आठवी विद्यार्थ्यांना परीक्षा रद्द करून व त्यांना पास करून पुढील वर्गात प्रमोट केले तर इयत्ता 9 ते 11 वर्गाच्या परीक्षा सुद्धा रद्द केल्या आहे. तरीसुद्धा, अनेक शाळांमध्ये ऑनलाइन वर्ग शिक्षकांकडून घेतले जात होते.

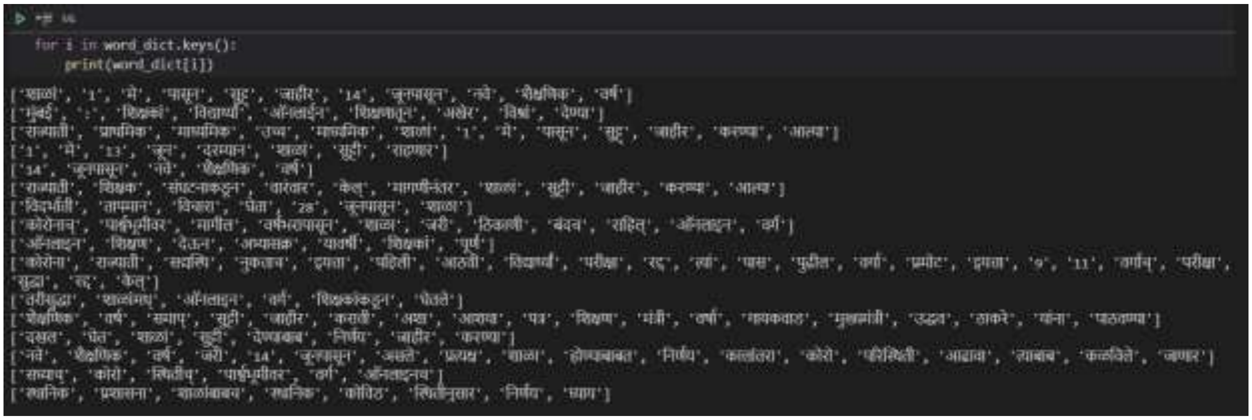
त्यामुळे शैक्षणिक वर्ष समाप्ती करून सुट्टी जाहीर करावी अशा आशयाचे पत्र शिक्षण मंत्री वर्षा गायकवाड, मुख्यमंत्री उद्धव ठाकरे यांना पाठवण्यात येत होते. याची दखल घेत, आज शाळांना सुट्टी देण्याबाबत निर्णय जाहीर करण्यात आला. नवे शैक्षणिक वर्ष जरी 14 जूनपासून सुरू होणार असले तरी प्रत्यक्ष शाळा सुरू होण्याबाबतचा निर्णय कालांतराने कोरोना परिस्थितीचा आढावा घेऊन त्याबाबत कळविले जाणार आहे. त्यामुळे

सध्याच्या कोरोना स्थितीच्या पार्श्वभूमीवर वर्ग ऑनलाइनच होणार आहेत. स्थानिक प्रशासनाने शाळांबाबच स्थानिक कोविड स्थितीनुसार निर्णय घ्यायचा आहे.

4.2 Output Details and Screenshots

Preprocessing:

After Tokenization, Stopword removal and stemming the output is stored in stemmed dictionary

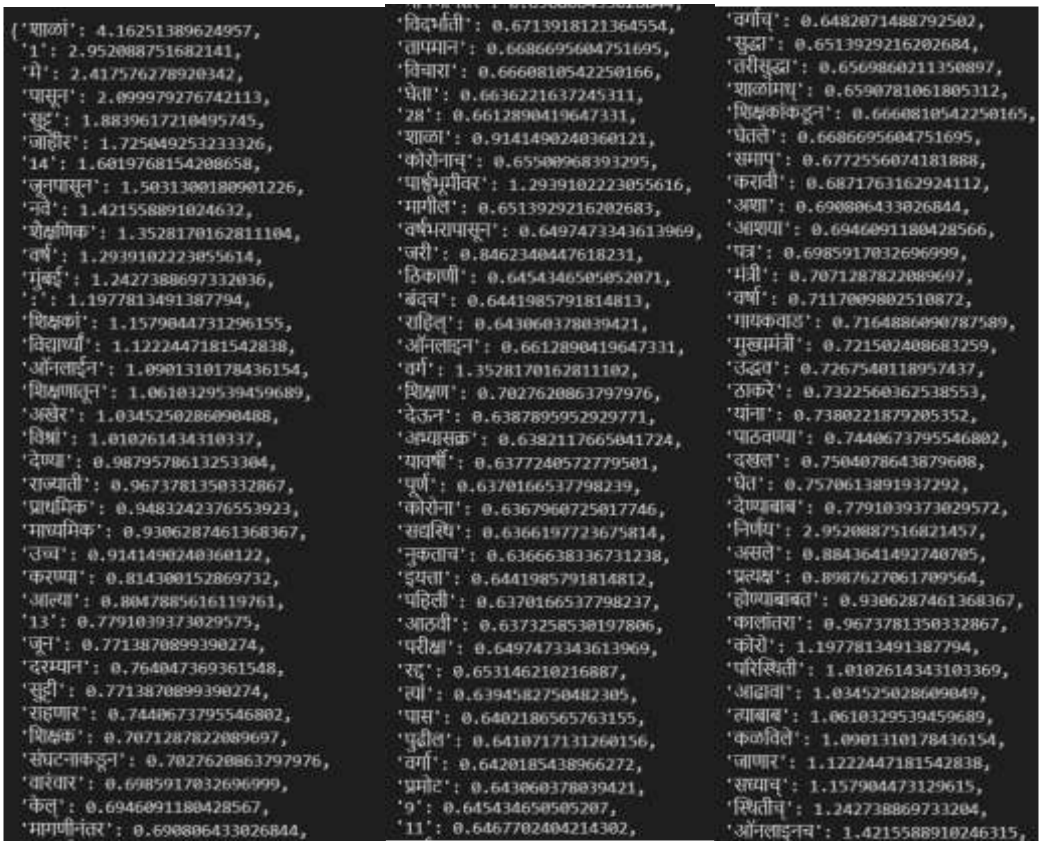


```
for i in word_dict.keys():
    print(word_dict[i])
```

```
['शाळां', '1', 'मे', 'पासून', 'सुट्ट', 'जाहीर', '14', 'जूनपासून', 'नवे', 'शैक्षणिक', 'वर्ष']
['मुंबई', ':', 'शिक्षकां', 'विद्यार्थी', 'ऑनलाईन', 'शिक्षणातून', 'अखेर', 'विश्रां', 'देण्या']
['राज्याती', 'प्राथमिक', 'माध्यमिक', 'उच्च', 'माध्यमिक', 'शाळां', '1', 'मे', 'पासून', 'सुट्ट', 'जाहीर', 'करण्या', 'आल्या']
['1', 'मे', '13', 'जून', 'दरम्यान', 'शाळां', 'सुट्टी', 'राहणार']
['14', 'जूनपासून', 'नवे', 'शैक्षणिक', 'वर्ष']
['राज्याती', 'शिक्षक', 'संघटनाकडून', 'वारंवार', 'केल्', 'मागणीनंतर', 'शाळां', 'सुट्टी', 'जाहीर', 'करण्या', 'आल्या']
['विदर्भाती', 'तापमान', 'विचारा', 'घेता', '28', 'जूनपासून', 'शाळा']
['कोरोनाच्', 'पार्श्वभूमीवर', 'मागील', 'वर्षभरापासून', 'शाळा', 'जरी', 'ठिकाणी', 'बंदच', 'राहित्', 'ऑनलाइन', 'वर्ग']
['ऑनलाइन', 'शिक्षण', 'देऊन', 'अभ्यासक्र', 'यावर्षी', 'शिक्षकां', 'पूर्ण']
['कोरोना', 'राज्याती', 'सद्यस्थि', 'नुकताच', 'इयत्ता', 'पहिली', 'आठवी', 'विद्यार्थ्यां', 'परीक्षा', 'रद्द', 'त्यां', 'पास', 'पुढील', 'वर्गा', 'प्रमोट', 'इयत्ता', '9', '11', 'वर्गांच्', 'परीक्षा',
'सुद्धा', 'रद्द', 'केल्']
['तरीसुद्धा', 'शाळांमध्', 'ऑनलाइन', 'वर्ग', 'शिक्षकांकडून', 'घेतले']
['शैक्षणिक', 'वर्ष', 'समाप्', 'सुट्टी', 'जाहीर', 'करावी', 'अशा', 'आशया', 'पत्र', 'शिक्षण', 'मंत्री', 'वर्षा', 'गायकवाड', 'मुख्यमंत्री', 'उद्धव', 'ठाकरे', 'यांना', 'पाठवण्या']
['दखल', 'घेत', 'शाळा', 'सुट्टी', 'देण्याबाब', 'निर्णय', 'जाहीर', 'करण्या']
['नवे', 'शैक्षणिक', 'वर्ष', 'जरी', '14', 'जूनपासून', 'असले', 'प्रत्यक्ष', 'शाळा', 'होण्याबाबत', 'निर्णय', 'कालांतरा', 'कोरो', 'परिस्थिती', 'आढावा', 'त्याबाब', 'कळविले', 'जाणार']
['सध्याच्', 'कोरो', 'स्थितीच्', 'पार्श्वभूमीवर', 'वर्ग', 'ऑनलाइनच']
['स्थानिक', 'प्रशासना', 'शाळांबाब', 'स्थानिक', 'कोविड', 'स्थितीनुसार', 'निर्णय', 'घ्याय']
```

Fig. 4.1. Tokenization of Input Text

After Finding Stemmed dictionary, We pass the stemmed dictionary to find nodes and their weights.



```
{'शाळां': 4.16251389624957,
 '1': 2.952088751682141,
 'मे': 2.417576278920342,
 'पासून': 2.099979276742113,
 'सुट्ट': 1.8839617210495745,
 'जाहीर': 1.725049253233326,
 '14': 1.6019768154208658,
 'जूनपासून': 1.5031300180901226,
 'नवे': 1.421558891024632,
 'शैक्षणिक': 1.3528170162811104,
 'वर्ष': 1.2939102223055614,
 'मुंबई': 1.2427388697332036,
 ':': 1.1977813491387794,
 'शिक्षकां': 1.1579044731296155,
 'विद्यार्थी': 1.1222447181542838,
 'ऑनलाईन': 1.0901310178436154,
 'शिक्षणातून': 1.0610329539459689,
 'अखेर': 1.0345250286090488,
 'विश्रां': 1.010261434310337,
 'देण्या': 0.9879578613253304,
 'राज्याती': 0.9673781350332867,
 'प्राथमिक': 0.9483242376553923,
 'माध्यमिक': 0.9306287461368367,
 'उच्च': 0.9141490240360122,
 'करण्या': 0.814300152869732,
 'आल्या': 0.8047885616119761,
 '13': 0.7791039373029575,
 'जून': 0.7713870899390274,
 'दरम्यान': 0.764047369361548,
 'सुट्टी': 0.7713870899390274,
 'राहणार': 0.7440673795546802,
 'शिक्षक': 0.7071287822089697,
 'संघटनाकडून': 0.7027620863797976,
 'वारंवार': 0.6985917032696999,
 'केल्': 0.6946091180428567,
 'मागणीनंतर': 0.690806433026844,
 'विदर्भाती': 0.6713918121364554,
 'तापमान': 0.6686695604751695,
 'विचारा': 0.6660810542250166,
 'घेता': 0.6636221637245311,
 '28': 0.6612890419647331,
 'शाळा': 0.9141490240360121,
 'कोरोनाच्': 0.655009683093295,
 'पार्श्वभूमीवर': 1.2939102223055616,
 'मागील': 0.6513929216202683,
 'वर्षभरापासून': 0.6497473343613969,
 'जरी': 0.8462340447618231,
 'ठिकाणी': 0.6454346505052071,
 'बंदच': 0.6441985791814813,
 'राहित्': 0.643060378039421,
 'ऑनलाइन': 0.6612890419647331,
 'वर्ग': 1.3528170162811102,
 'शिक्षण': 0.7027620863797976,
 'देऊन': 0.6387895952929771,
 'अभ्यासक्र': 0.6382117665041724,
 'यावर्षी': 0.6377240572779501,
 'पूर्ण': 0.6370166537798239,
 'कोरोना': 0.6367960725017746,
 'सद्यस्थि': 0.6366197723675814,
 'नुकताच': 0.6366638336731238,
 'इयत्ता': 0.6441985791814812,
 'पहिली': 0.6370166537798237,
 'आठवी': 0.6373258530197806,
 'परीक्षा': 0.6497473343613969,
 'रद्द': 0.653146210216887,
 'त्यां': 0.6394582750482305,
 'पास': 0.6402186565763155,
 'पुढील': 0.6410717131260156,
 'वर्गा': 0.6420185438966272,
 'प्रमोट': 0.643060378039421,
 '9': 0.645434650505207,
 '11': 0.6467702404214302,
 'वर्गांच्': 0.6482071488792502,
 'सुद्धा': 0.6513929216202684,
 'तरीसुद्धा': 0.6569860211350897,
 'शाळांमध्': 0.6590781061805312,
 'शिक्षकांकडून': 0.6660810542250165,
 'घेतले': 0.6686695604751695,
 'समाप्': 0.6772556074181888,
 'करावी': 0.6871763162924112,
 'अशा': 0.690806433026844,
 'आशया': 0.6946091180428566,
 'पत्र': 0.6985917032696999,
 'मंत्री': 0.7071287822089697,
 'वर्षा': 0.7117009802510872,
 'गायकवाड': 0.7164886090787589,
 'मुख्यमंत्री': 0.721502408683259,
 'उद्धव': 0.7267540118957437,
 'ठाकरे': 0.7322560362538553,
 'यांना': 0.7380221879205352,
 'पाठवण्या': 0.7440673795546802,
 'दखल': 0.7504078643879608,
 'घेत': 0.7570613891937292,
 'देण्याबाब': 0.7791039373029572,
 'निर्णय': 2.9520887516821457,
 'असले': 0.8843641492740705,
 'प्रत्यक्ष': 0.8987627061709564,
 'होण्याबाबत': 0.9306287461368367,
 'कालांतरा': 0.9673781350332867,
 'कोरो': 1.1977813491387794,
 'परिस्थिती': 1.0102614343103369,
 'आढावा': 1.034525028609049,
 'त्याबाब': 1.0610329539459689,
 'कळविले': 1.0901310178436154,
 'जाणार': 1.1222447181542838,
 'सध्याच्': 1.157904473129615,
 'स्थितीच्': 1.242738869733204,
 'ऑनलाइनच': 1.4215588910246315,
```

Fig. 4.2. Stemmed Dictionary

We create a graph for the nodes using networkx package.

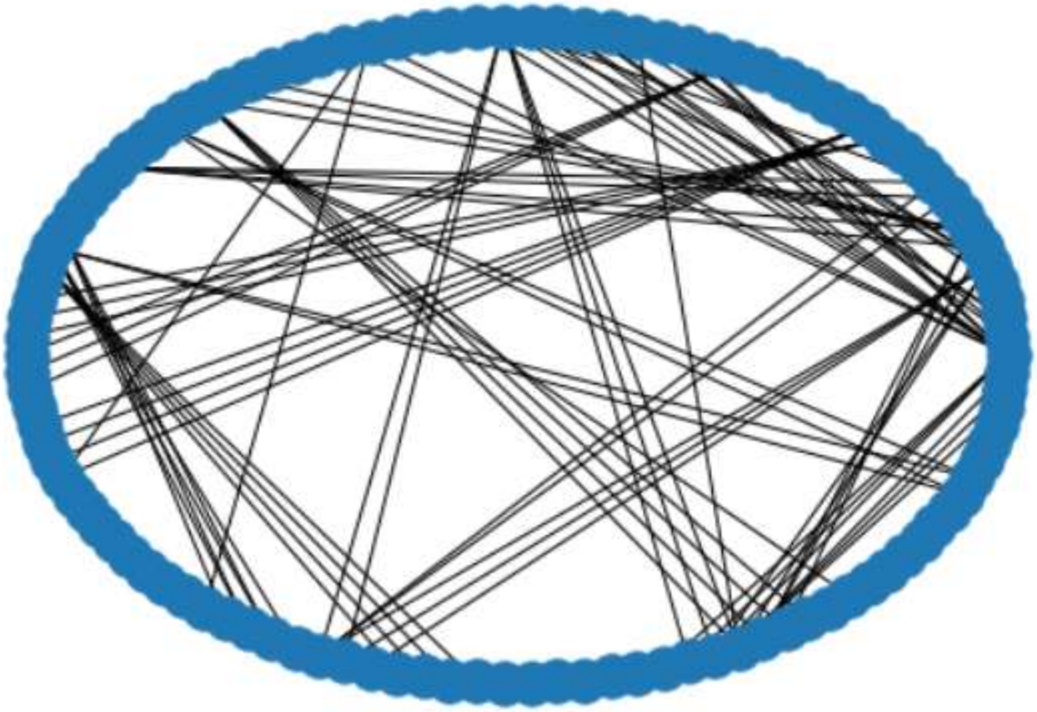
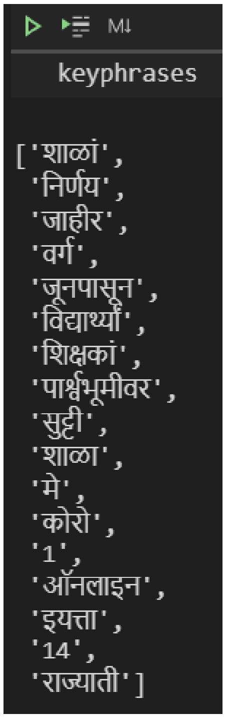


Fig.4.3. TextRank Graph

After creation of textrank graph, we find the highest ranking keyphrases.

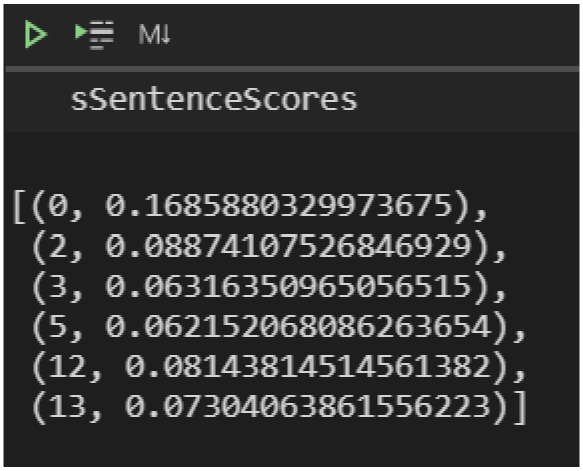


```
keyphrases
```

```
['शाळां',
 'निर्णय',
 'जाहीर',
 'वर्ग',
 'जूनपासून',
 'विद्यार्थ्यां',
 'शिक्षकां',
 'पार्श्वभूमीवर',
 'सुट्टी',
 'शाळा',
 'मे',
 'कोरो',
 '1',
 'ऑनलाइन',
 'इयत्ता',
 '14',
 'राज्याती']
```

Fig 4.4. KeyPhrases

After finding highest ranking text phrases, we use them to find Sentence scores.



```
sSentenceScores
```

```
[(0, 0.1685880329973675),
 (2, 0.08874107526846929),
 (3, 0.06316350965056515),
 (5, 0.062152068086263654),
 (12, 0.08143814514561382),
 (13, 0.07304063861556223)]
```

Fig 4.5. Sentence Scores

These are the top sentence scores. Hence, the summary for given article is:

शाळांना 1 मे पासून सुट्ट्या जाहीर, 14 जूनपासून नवे शैक्षणिक वर्ष सुरू होणार.
राज्यातील प्राथमिक, माध्यमिक, उच्च माध्यमिक शाळांना 1 मे पासून सुट्ट्या जाहीर करण्यात आल्या आहे.
1 मे ते 13 जून दरम्यान शाळांना सुट्टी राहणार आहे.
राज्यातील शिक्षक संघटनाकडून वारंवार केल्या जात असलेल्या मागणीनंतर शाळांना सुट्टी जाहीर करण्यात आल्या आहेत.
याची दखल घेत, आज शाळांना सुट्टी देण्याबाबत निर्णय जाहीर करण्यात आला.
नवे शैक्षणिक वर्ष जरी 14 जूनपासून सुरू होणार असले तरी प्रत्यक्ष शाळा सुरू होण्याबाबतचा निर्णय कालांतराने कोरोना परिस्थितीचा आढावा घेऊन त्याबाबत कळविले जाणार आहे.

Chapter 5

Summary and Future Scope

5.1 Summary

Our main aim for designing the project was to summarize news articles so that readers can easily read summaries of articles without need of a human to write those summaries. For this task, we opted for TextRank Algorithm. The product of the textrank algorithm will give us the keyphrases from the news articles. This gives us the most important stemmed words. The textrank algorithm is initialised on position of word after stemming. We perform sentence scoring to find the highest scoring sentences. For Preprocessing, we create our stopwords list, tokenized the article and cleaned the text to give us stemmed dictionary. This stemmed dictionary is then passed onto Textrank algorithm that gives us a short summary of the article.

5.2 Future Scope

1. Tagging for Marathi could allow for better results in our summary.
2. To make our news summarization abstractive, we could add NER for finding Noun phrase and Verb Phrases.
3. Further, we can use Sentence Feature Extraction to generate sentences that are not in the news articles.

References

[1] V. Giri, Dr. M. Math, Dr. U. Kulkarni, “ A Survey of Automatic Text Summarization System for Different Regional Language in India”, Bonfring International Journal of Software Engineering and Soft Computing(BIJSESC), October 2016.

[2] S. Shimpikar, S. Govilkar, “ Abstractive Text Sumarization using Rich Semantic Graph for Marathi Sentences”, Journal of Applied Science and Computations(JASC), December 2018.

[3] M. Majgaonkar, T. Siddiqui, “Discovering Suffixes, A case study for Marathi Language”, International Journal on Computer Science and Engineering(IJCSE), 2010.

[4] P. Gawade, D. Madhavi, J. Gaikwad, S. Jadhav, “Natural Language Processing Tasks for Marathi Language”, International Journal of Engineering Research and Development(IJERD), April 2013

[5] R. Mihalcea, P. Tarau, “TextRank: Bringing Order to Texts”, 2004 Conference on Empirical Methods in Natural Language Processing, July 2004.

Acknowledgement

We wish to express our deep sense of gratitude and thanks to our Professor and Guide for this Mini Project, Prof. Pranita Mahajan for her guidance, help and useful suggestions, which helped in completing our project work in time.

We also thank our H.O.D., Dr. Aparna Bannore for her support in completing the project. We are also extremely grateful for her guidance provided whenever required.

We also thank to our Principal Dr. Atul Kemkar, for extending his support to carry out this project. Also, we would like to thank the entire faculty of the Computer Department for their valuable ideas and timely assistance in this project, last but not the least, we would like to thank our teaching and non-teaching staff members of our college for their support, in facilitating timely completion of this project.

Project Team

Saail Ganesh(117A1061)

Santosh Krishnan(117A1065)

Nabeel Sarguroh(117A1066)