



Tribhuvan University

Institute of Science and Technology

A Final Year Project Report

On

AUTOMATIC TEXT SUMMARIZER

Submitted To:

Department of Computer Science and Information Technology

Kathford International College of Engineering and Management

**In partial fulfillment of the requirement for the Bachelor Degree in Computer
Science and Information Technology**

Submitted By:

Sanjay Parajuli (5036/071)

Sanjiv Chitrakar (5037/071)

Sudip Basnet (5043/071)

Vijay Shrestha (5051/071)

September, 2018

Supervisor's Recommendation

I hereby recommend that this report has been prepared under my supervision by Sanjay Parajuli (TU Exam Roll No. 5036/071), Sanjiv Chitrakar (TU Exam Roll No. 5037/071), Sudip Basnet (TU Exam Roll No. 5043/071) and Vijay Shrestha (TU Exam Roll No. 5051/071) entitled “**AUTOMATIC TEXT SUMMARIZER**” in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for evaluation.

.....

Mr. Bijay Mishra

Project Supervisor

Kathford International College of Engineering and Management

Balkumari, Lalitpur

CERTIFICATE OF APPROVAL

This is to certify that this project prepared by Sanjay Parajuli (TU Exam Roll No. 5036/071), Sanjiv Chitrakar (TU Exam Roll No. 5037/071), Sudip Basnet (TU Exam Roll No. 5043/071) and Vijay Shrestha (TU Exam Roll No. 5051/071) entitled “**AUTOMATIC TEXT SUMMARIZER**” in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

<hr style="width: 20%; margin-left: 0;"/> <p>Mr. Bijay Mishra Supervisor Visiting Faculty Department of Computer Science and IT Kathford International College of Engineering and Management Balkumari, Lalitpur</p>	<hr style="width: 20%; margin-left: 0;"/> <p>Er. Suwas Karki Head Department of Computer Science and IT Kathford International College of Engineering and Management Balkumari, Lalitpur</p>
<hr style="width: 20%; margin-left: 0;"/> <p>(External Examiner) Tribhuvan University</p>	

ACKNOWLEDGEMENT

It is a great pleasure to have the opportunity to extend our heartfelt gratitude to everyone who helped us throughout the course of this project. We are profoundly grateful to our supervisor **Mr. Bijay Mishra**, for his expert guidance, continuous encouragement and ever willingness to spare time from his otherwise busy schedule for the project's progress reviews. His continuous inspiration has made us complete this project and achieve its target.

We would also like to express our deepest appreciation to **Er. Suwas Karki** Head of Department, Kathford International College of Engineering and Management, Department of Computer Science and Information Technology, for his constant motivation, support and for providing us with a suitable working environment. We would also like to extend our sincere regards to **Mr. Madhav Subedi** and all the faculty members for their support and encouragement.

At last our special thanks go to all staff members of B.Sc.CSIT department who directly and indirectly extended their hands in making this project works a success.

ABSTRACT

In the modern Internet age, textual data is ever increasing. Need some way to condense this data while preserving the information and meaning. We need to summarize textual data for that. To address that need, the purposed system was developed. The purposed system uses extractive algorithm to summarize long textual data. In addition, two extractive algorithms namely Word Frequency and Sentence Scoring were used to generate summary. Furthermore, a mechanism to generate summarized news from popular websites was developed and users were given the option to save the summary.

Keywords: *Automatic text summarizer, Extractive algorithm, Word Frequency, Sentence Scoring, Natural language summaries, Summarized news*

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION	1
1.1. Project Introduction.....	1
1.2. Problem Statement	2
1.3. Objectives.....	2
1.4. Scope	3
1.5. Report Organization	3
CHAPTER 2: REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS	5
2.1. Literature Review	5
2.2. Requirement Analysis	6
2.2.1. Functional Requirements	6
2.2.2. Non Functional Requirements	7
2.3. Feasibility Analysis	8
CHAPTER 3: SYSTEM ANALYSIS.....	9
3.1. Structuring System Requirements.....	9
3.1.1. Data Modeling	9
3.1.2. Process Modeling	10
CHAPTER 4: SYSTEM DESIGN.....	13
4.1. Database Schema.....	13
4.3. Class Diagram	15
4.4. Sequence Diagram.....	15
4.5. Activity Diagram.....	16
CHAPTER 5: IMPLEMENTATION	18
5.1. Tools Used.....	18
5.2. Methodology	19
5.2.1. Essential Methods of Sentence Scoring Algorithms	19
5.2.2. Essential Methods of Word Frequency Algorithms	20
5.2.3. Methods for Generating Machine Learning Models	21
CHAPTER 6: SYSTEM TESTING.....	24
6.1. Unit Testing.....	24
6.2. Integration Testing	25
6.3. ROUGE Evaluation.....	25
CHAPTER 7: CONCLUSION AND RECOMMENDATION	27

6.1. Conclusion.....	27
6.2. Future Scope.....	27
REFERENCES	28
APPENDIX.....	29

LIST OF FIGURES

Figure 2.1: Use Case Diagram for Automatic Text Summarizer.....	7
Figure 3.1: ER Diagram for Automatic Text Summarizer	9
Figure 3.2: Context Diagram for Automatic Text Summarizer	10
Figure 3.3: Level 0 DFD for Automatic Text Summarizer	11
Figure 3.4: Level 1 DFD for process 3.0 of level 0 DFD	12
Figure 4.1: Database Schema for Automatic Text Summarizer	13
Figure 4.2: Input Design of Automatic Text Summarizer	14
Figure 4.3: Output Design of Automatic Text Summarizer	14
Figure 4.4: Class Diagram for Automatic Text Summarizer	15
Figure 4.5: Sequence Diagram for Automatic Text Summarizer	16
Figure 4.6: Activity Diagram for Automatic Text Summarizer	17
Figure 5.1: Encoder Decoder Architecture with Attention	23

LIST OF TABLES

Table 6.1: Unit Testing of Automatic Text Summarizer	24
Table 6.2: Rouge Metrics for Sentence Scoring Algorithm	25
Table 6.3: ROUGE Metrics for Frequency Word Algorithm	26
Table 6.4: ROUGE Metrics for Machine Learning Model.....	26

ABBREVIATIONS

API	Application Program Interface
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
ER	Entity Relationship
GloVe	Global Vectors
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
LSTM	Long Short Term Memory
LCS	Longest Common Subsequence
ML	Machine Learning
OOV	Out of Vocabulary
PDF	Portable Document Format
RNN	Recurrent Neural Network
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
Seq2Seq	Sequence-to-Sequence
URL	Uniform Resource Locator
UML	Unified Modeling Language
Word2Vec	Word-to-Vector
XML	Extensible Markup Language

CHAPTER 1: INTRODUCTION

1.1. Project Introduction

Automatic text summarizer is a web based application that converts any long text into short summary. The system described in this paper achieved text summarization by distilling the most important information from different sources to produce an abridged version for a particular user. Automatic text summarization methods are greatly needed to address the ever-growing amount of text data available to both better help discover relevant information and to consume relevant information faster. Think of the internet, comprised of web pages, news articles, status updates, blogs and so much more. There is a great need to reduce much of this text data to shorter, focused summaries that capture the salient details.

Text summarization is the process of automatically generating natural language summaries from an input document while retaining the important points. It would help in easy and fast retrieval of information. There are two prominent types of summarization algorithms:

- Extractive summarization systems form summaries by copying parts of the source text through some measure of importance and then combine those part/sentences together to render a summary. Importance of sentence is based on linguistic and statistical features.
- Abstractive summarization systems generate new phrases, possibly rephrasing or using words that were not in the original text. Naturally abstractive approaches are harder. For perfect abstractive summary, the model has to first truly understand the document and then try to express that understanding in short possibly using new words and phrases. Much harder than extractive. Has complex capabilities like generalization, paraphrasing and incorporating real-world knowledge.

In this project, majority of the work was done using the extractive algorithms as they promise grammatically correct and coherent summary. But they often didn't provide correct summary for long and complex texts which resulted us to train a machine learning model. This abstractive approach was mainly based on Tensorflow sequence to sequence library and also embraced different machine learning mechanisms.

1.2. Problem Statement

Everybody wanted to gain more knowledge by spending less time on reading. Who doesn't love to read summary of a large text material? Which gives mostly the same information as if reading the whole text spending lots of time. Automatic text summarization methods were greatly required to address the ever-growing amount of text data available online to both better help discover relevant information and to consume relevant information faster.

In this new era, where tremendous information is available on the internet, it is most important to provide the improved mechanism to extract the information quickly and most efficiently. It was very difficult for human beings to manually extract the summary of a large documents of text. There are plenty of text materials available on the internet. So there was a problem of searching for relevant documents from the number of documents available, and absorbing relevant information from it.

In order to solve such problems, the automatic text summarizer implements two different extractive approaches: word frequency counter and sentence matching. These two algorithms were used in the system to create a summary of long textual data. However, in extractive based text summarization, the extracted sentences usually have much higher length with respect to information it acquires. Due to this reason, segments which are unnecessary for summary also occupy space. Relevant information is normally spread across many sentences which extractive method do not carry in summary. So to solve this problem, a machine learning model was trained on toy datasets to generate summary based on abstractive method. However, the deployment of this ML model in production has some requirements and limitations due to which its implementation was restricted. Though the ML model was trained and tested which resulted in bit skeptical summary.

1.3. Objectives

The objectives of this automatic text summarizer are:

1. To create automatic text summarization system.
2. To implement extractive methods for text summarization.
3. To provide summarized news from various popular websites feeds.

1.4. Scope

Automatic text summarizer can be used to summarize textual document which can either be provided by the user or extracted from the URL provided by the user. A Summarizer API can be built from this project. Such API can be implemented by any third party website to add summarize option in their respective website which contains large text. Talking about retrieving short and relevant information for large text, it can also be used in data mining. Although extractive approach may not give accurate summary to fully implement in data mining, if abstractive approach is implemented in future, it will be effective in data mining from large data warehouse. Thus, based on abstractive approach a machine learning model was trained and tested. However, this model was not deployed in the production due to various reasons. In future, this machine learning model could be implemented into production which can be used to generate abstractive summary directly from the application.

1.5. Report Organization

This report is organized into 6 chapters as follows:

Chapter 1:

The project is introduced in detail with its objectives and scope.

Chapter 2:

It contains the functional and non-functional requirements of the project. Analysis and evaluation of project is done by feasibility analysis.

Chapter 3:

Data modeling and process modeling of the project is done to analyze the data and working mechanism of the system in detail.

Chapter 4:

Explains the methods and tools used to implement the project.

Chapter 5:

Different Test Cases used during testing of each component of the system.

Chapter 6:

It contains conclusion and recommendations based on the project.

CHAPTER 2: REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS

2.1. Literature Review

There are two main approaches to summarizing text documents; they are [1]:

1. Extractive Methods.
2. Abstractive Methods.

Extractive text summarization involves the selection of phrases and sentences from the source document to make up the new summary. Techniques involve ranking the relevance of phrases in order to choose only those most relevant to the meaning of the source [2].

Abstractive text summarization involves generating entirely new phrases and sentences to capture the meaning of the source document. This is a more challenging approach, but is also the approach ultimately used by humans. Classical methods operate by selecting and compressing content from the source document [2].

Recently deep learning methods have shown promising results for text summarization. Approaches have been proposed inspired by the application of deep learning methods for automatic machine translation, specifically by framing the problem of text summarization as a sequence-to-sequence learning problem. These deep learning approaches to automatic text summarization may be considered abstractive methods and generate a wholly new description by learning a language generation model specific to the source documents [3].

The results of deep learning methods are not yet state-of-the-art compared to extractive methods, yet impressive results have been achieved on constrained problems such as generating headlines for news articles that rival or out-perform other abstractive methods. The promise of the approach is that the models can be trained end-to-end with specialized data preparation or sub-models and that the models are entirely data-driven, with the preparation of specialized vocabulary or expertly pre-processed source documents [4].

2.2. Requirement Analysis

Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project. Requirement analysis is mainly categorized into two types:

2.2.1. Functional Requirements

The functional requirements for a system describe what the system should do. Those requirements depend on the type of software being developed, the expected users of the software. These are statement of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situation.

Automatic text summarizer has a solo thing to achieve that is to create summary after the user has provided long textual data or a URL from which textual data is to be extracted and then summarized. After the user has provided all required inputs the summarizer provides summary as a result. If the user has given any malicious input, the summarizer returns particular exception or any sort of error message to the user.

The functional requirements of this system are as follows:

- Allow user to input long textual data or URL
- Allow user to upload the pdf file for summarization
- Allow user to select one of the extractive algorithms
- Allow user to summarize the textual data
- Allow user to login in or sign up in the system
- Allow user to save the summary
- Allow user to view summarized news from popular websites
- Allow the admin to manage the users

The following Use Case Diagram describes the major actions of the system and interaction between actors (user, admin) and the system in our application:

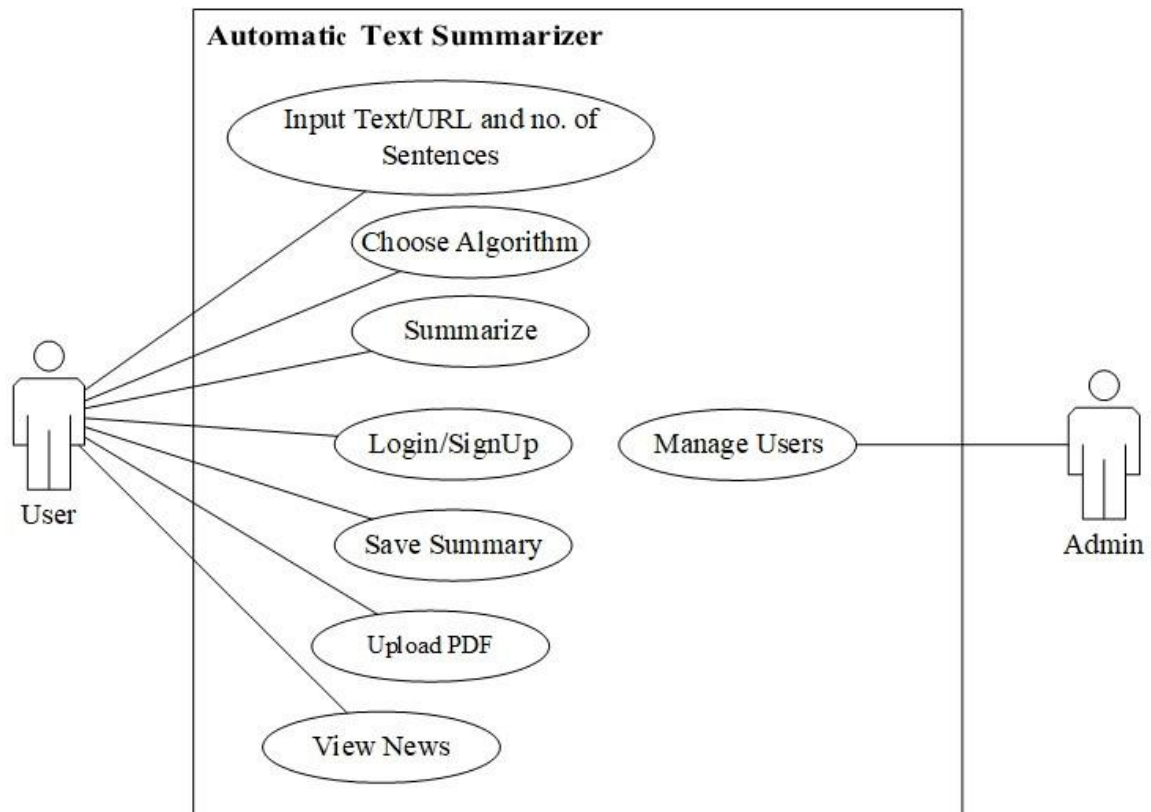


Figure 2.1: Use Case Diagram for Automatic Text Summarizer

The use case diagram of the automatic text summarizer is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. This diagram shows that the user can provide textual data or URL as an input and can also choose from the available algorithms. The user can also upload the textual data in pdf file format. In addition, the user can also save the summary and view the summarized news from popular websites.

2.2.2. Non Functional Requirements

Non-functional requirements are requirements that are not directly concerned with the specified function delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Some of the non-functional requirements related with this system are hereby below:

- **Reliability:**
Reliability based on this system defines the evaluation result of the system, correct summary of long textual data and summary generation in minimum time period. The system must be able to generate summary for all possible inputs.
- **Ease of Use:**
The system is simple, user friendly and graphics user interface implemented is easy to navigate so that anyone can use this system without any difficulties.
- **Scalability:**
The automatic text summarizer must be able to scale not only to English language but also to Nepali language simply by adding some regular expressions for tokenization and stop word removal.

2.3. Feasibility Analysis

Before starting the project, feasibility study is carried out to measure the viable of the system. Following are the feasibility that is concerned in this project:

- **Technical Feasibility:**
Technical feasibility is one of the first studies that must be conducted after the project has been identified. Technical feasibility study includes the hardware and software devices. The required technologies (Python, Django and PyCharm IDE) existed. Furthermore, to train the ML model advance technologies such as Tensorflow and other machine learning libraries existed. For training the ML model large datasets were required and most of this datasets were licensed which resulted in restriction in its usage. However, toy dataset was used to train ML model in this project.
- **Operational Feasibility:**
The purposed system is compatible and easy to use. The proper guidance from the supervisor existed. Any user with basic knowledge of computer and internet can access it.
- **Economic Feasibility:**
The required resources were freely available for the purposed system. It can be assured that the system is economically feasible.

CHAPTER 3: SYSTEM ANALYSIS

3.1. Structuring System Requirements

The field of system analysis relates closely to requirements analysis. It is also an explicit formal inquiry carried out to help a decision maker identify a better course of action. It contains the unpacking of the system requirements from data modeling and process modeling of the system.

3.1.1. Data Modeling

Data modeling is a process used to define and analyze data requirements needed to support the business process within the scope of corresponding information systems in organizations.

ER – Diagram

An entity-relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. The following ER model shows the entities, their attributes and relationships between them in our application.

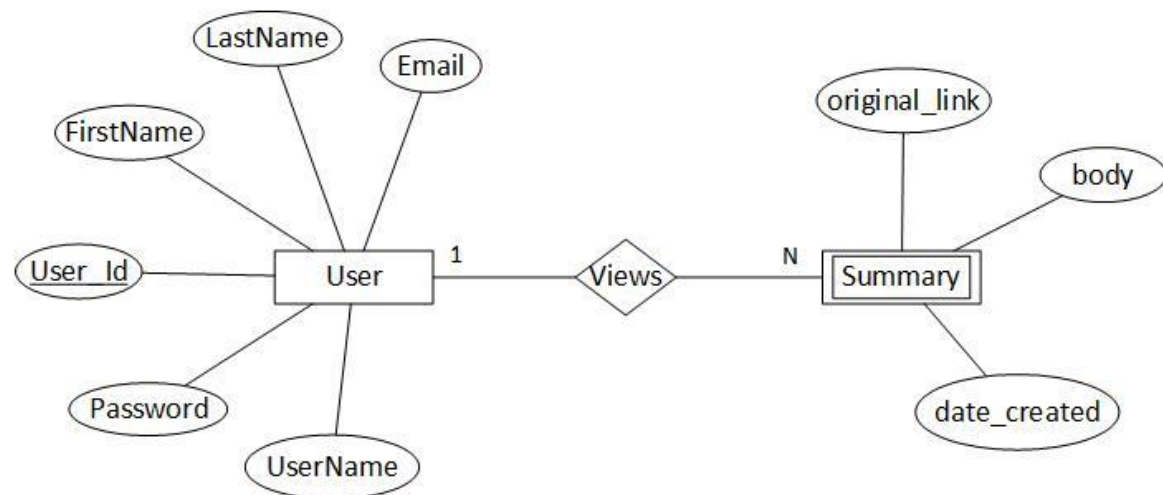


Figure 3.1: ER Diagram for Automatic Text Summarizer

The above ER diagram of the purposed system is simple. There are two main entities namely user and summary. A user can view multiple summaries so there existed one to many relationships between them. Likewise, the user entity has several attributes namely

first name, last name, username, email, etc. In addition, the summary entity also has several attributes namely original link, body and date created.

3.1.2. Process Modeling

Process modeling is a technique for organizing and documenting the structure and flow of data through a system's processes and/or the logic, policies, and procedures to be implemented by a system's processes.

Data Flow Diagram

Data flow diagram (DFD) is an analysis tool to represent the flow of data through an information system. The data flow diagram of our system is divided into context diagram and level 0 DFD.

Context Diagram

The context diagram of our system is shown as below:

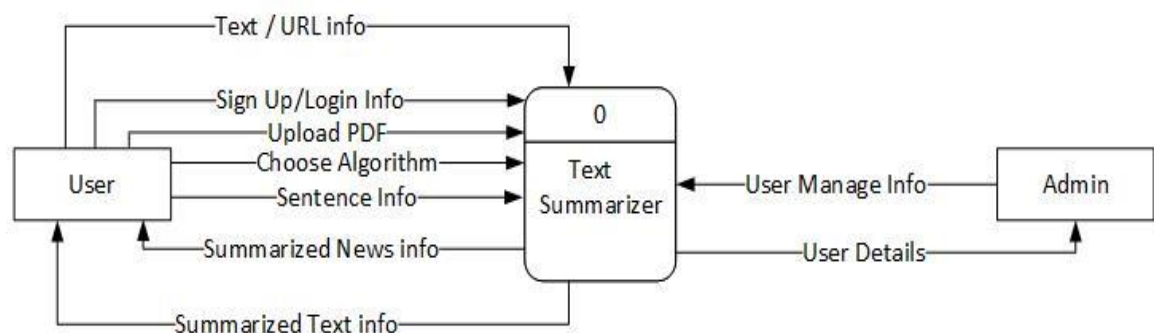


Figure 3.2: Context Diagram for Automatic Text Summarizer

A context diagram defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it. This diagram is a high level view of a system. It is similar to a block diagram. The above context diagram shows that the purposed system has two distinct entities namely user and admin with a process text summarizer. All of these entities interact with each other in several ways as shown in the figure above.

Level 0 DFD

The level 0 DFD of the above context diagram is shown as below:

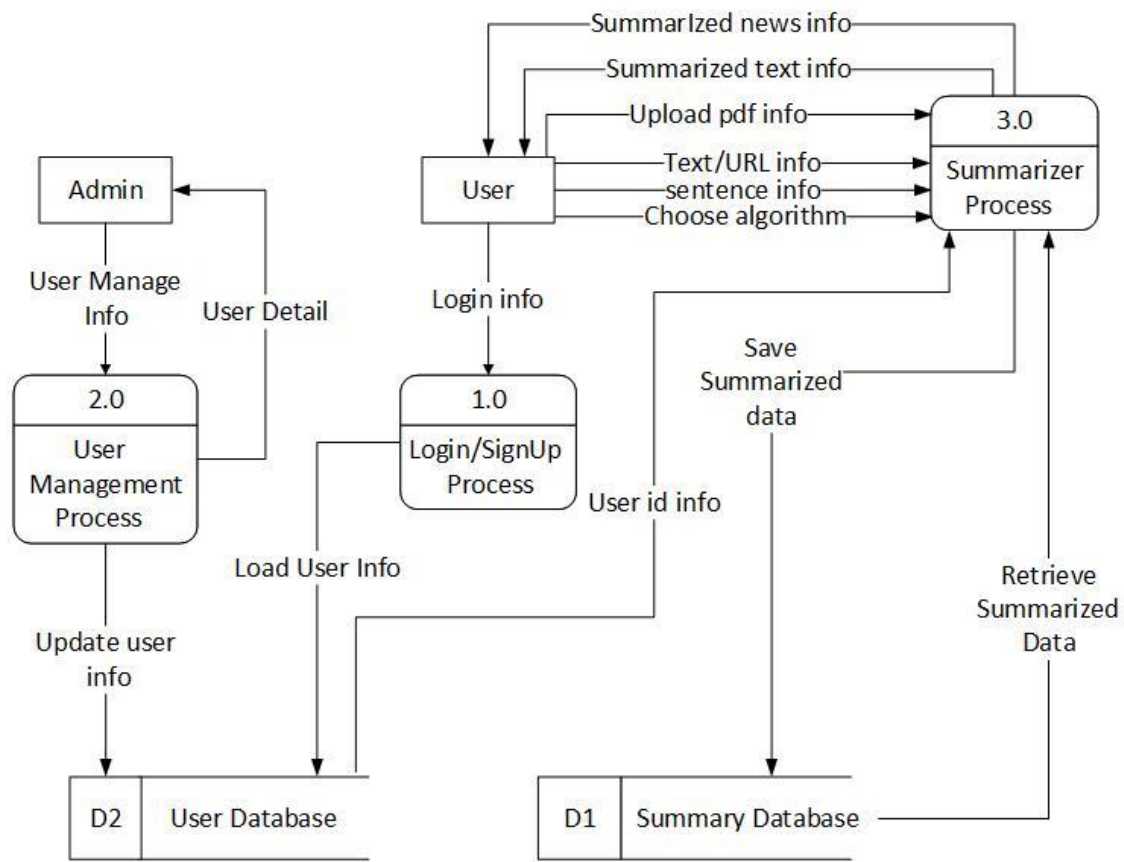


Figure 3.3: Level 0 DFD for Automatic Text Summarizer

The above level 0 DFD diagram of the purposed system shows the data flow in the automatic text summarizer. The admin can manage the user info and update the user database accordingly. Likewise, the user can provide login info which is authenticated by accessing the user database. The user is also allowed to provide textual data, number of sentences and algorithm as an input to the summarizer process. The summarizer process can generate the summary as per the input which can be save by the user in the summary database.

Level 1 DFD

The level 1 DFD of process 3.0 is shown as below:

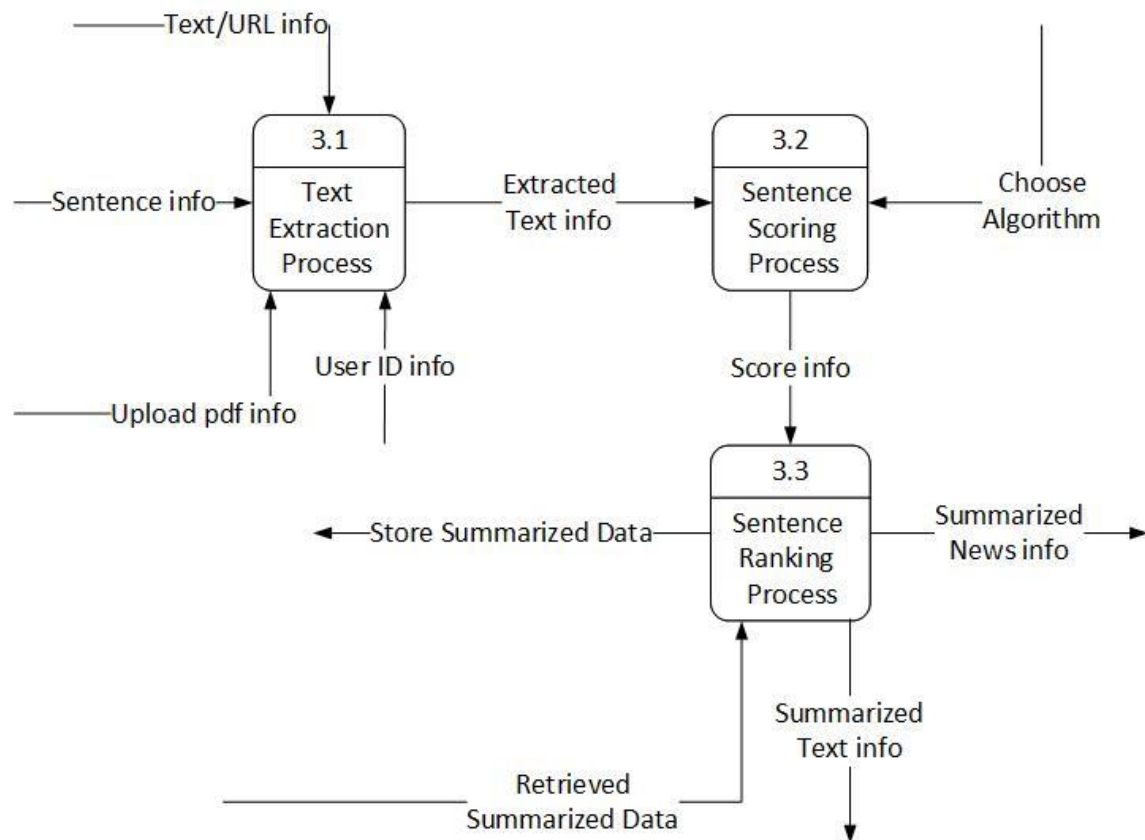


Figure 3.4: Level 1 DFD for process 3.0 of level 0 DFD

In above level 1 DFD for process 3.0 of level 0 DFD, there are three major processes namely text extraction process, sentence scoring process and sentence ranking process. The user can provide text or URL info, sentence info, pdf info and user id info to the text extraction process which generates extracted text info as an output. This output act as an input to the sentence scoring process which also receive selected algorithm as an input and generates score info as an output. This output act as an input to the sentence ranking process which can load summarized data in database and also retrieve the summarized data store in the database. This process can generate summarized news info and summarized text info as an output.

CHAPTER 4: SYSTEM DESIGN

System design is basically a process of defining the components, modules, interfaces and data for a system in order to satisfy specified requirements. It can also be defined as a process of creating or altering systems along with the processes, practices, models and methodologies that can be used to develop them. The main objective of the detailed system design is to prepare a blueprint of a system that meets the goals of the conceptual system design requirements. The system designs used for building this project include database schema, input output design, class diagram, sequence diagram and activity diagram.

4.1. Database Schema

The following database schema of our application is the structure of the database used in the system in a formal language.

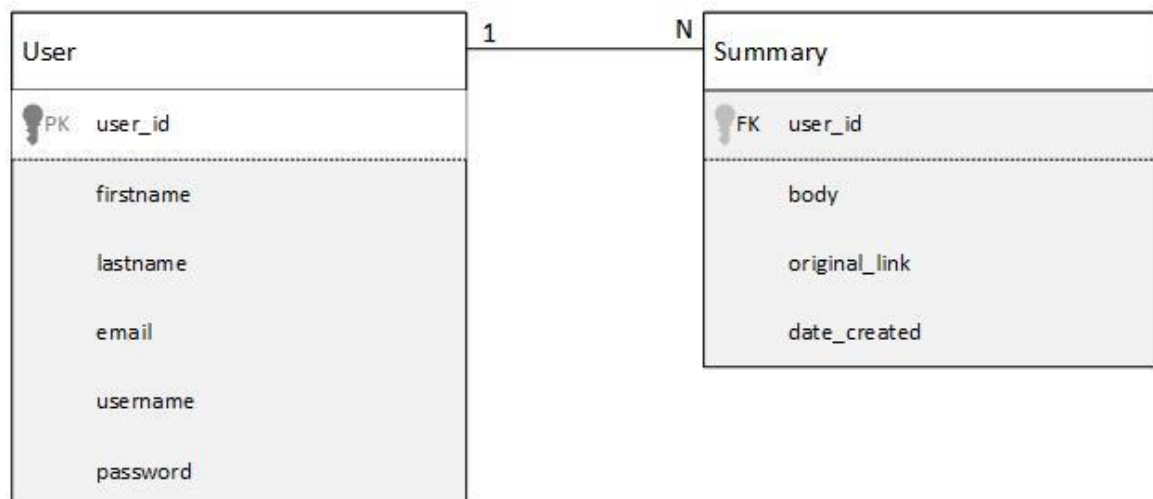
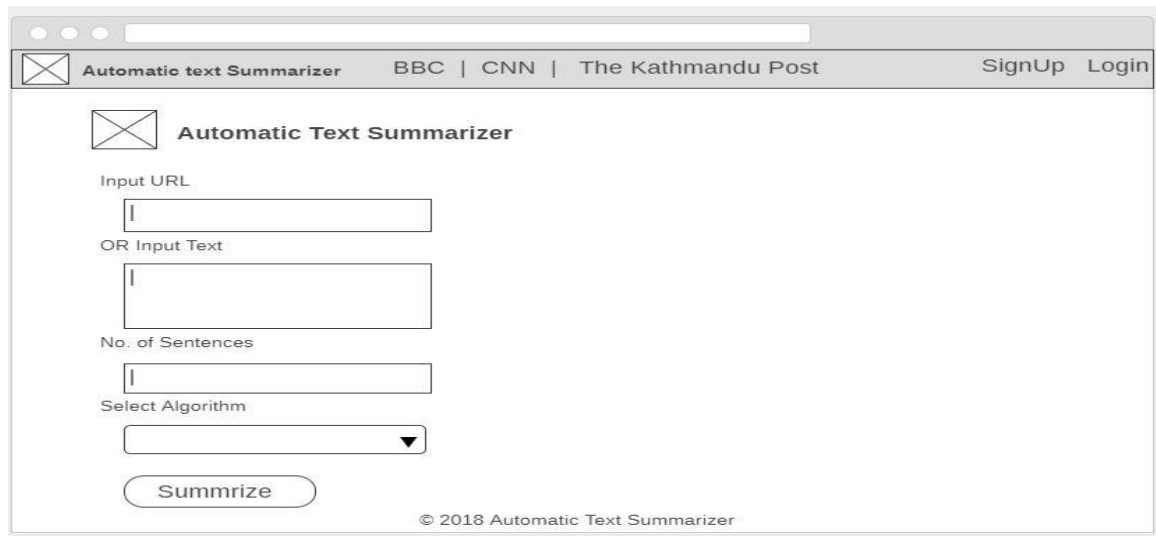


Figure 4.1: Database Schema for Automatic Text Summarizer

The above database schema of the purposed system describes the actual structure of the database. It clearly shows the implementation of two tables namely user and summary with respective attributes.

4.2. Input Output Design

Input design is the process of converting a user-oriented description of the input into a computer-based system. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. The following design shows the major input and output design of our system.

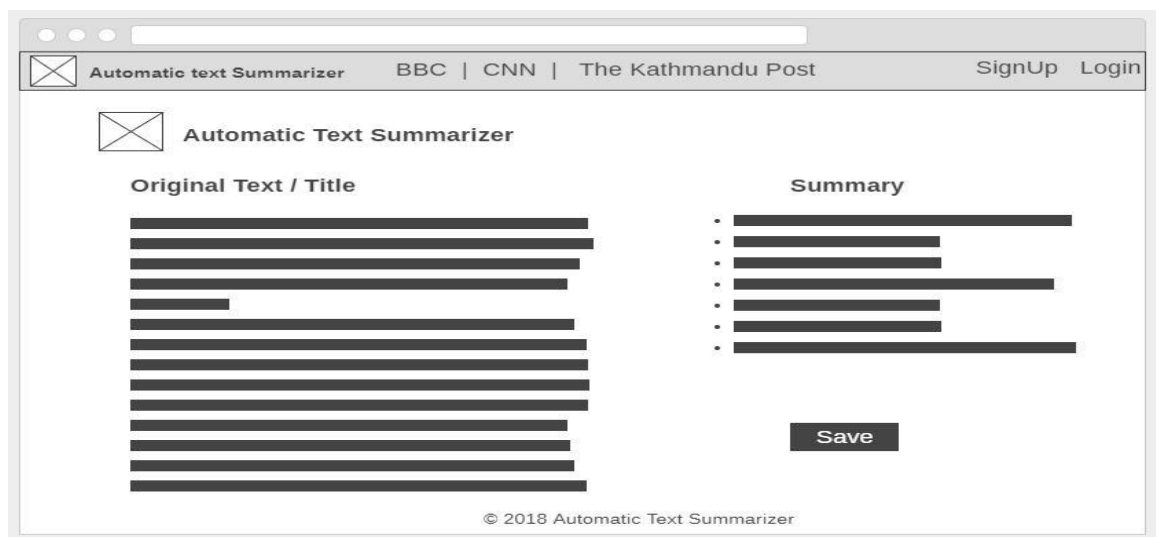


The image shows a web browser window with the title "Automatic text Summarizer". The browser's address bar contains "BBC | CNN | The Kathmandu Post". The page has a navigation bar with "SignUp" and "Login" links. The main content area is titled "Automatic Text Summarizer" and contains the following input fields:

- Input URL:** A text input field.
- OR Input Text:** A text input field.
- No. of Sentences:** A text input field.
- Select Algorithm:** A dropdown menu.
- Summize:** A button.

At the bottom of the page, there is a copyright notice: "© 2018 Automatic Text Summarizer".

Figure 4.2: Input Design of Automatic Text Summarizer



The image shows a web browser window with the title "Automatic text Summarizer". The browser's address bar contains "BBC | CNN | The Kathmandu Post". The page has a navigation bar with "SignUp" and "Login" links. The main content area is titled "Automatic Text Summarizer" and contains the following output fields:

- Original Text / Title:** A large text area containing multiple lines of placeholder text.
- Summary:** A list of bullet points, each followed by a line of placeholder text.
- Save:** A button.

At the bottom of the page, there is a copyright notice: "© 2018 Automatic Text Summarizer".

Figure 4.3: Output Design of Automatic Text Summarizer

4.3. Class Diagram

The following class diagram is an illustration of the relationships and source code dependencies among classes in our system. Each class has three sections: Name, attributes and operations.

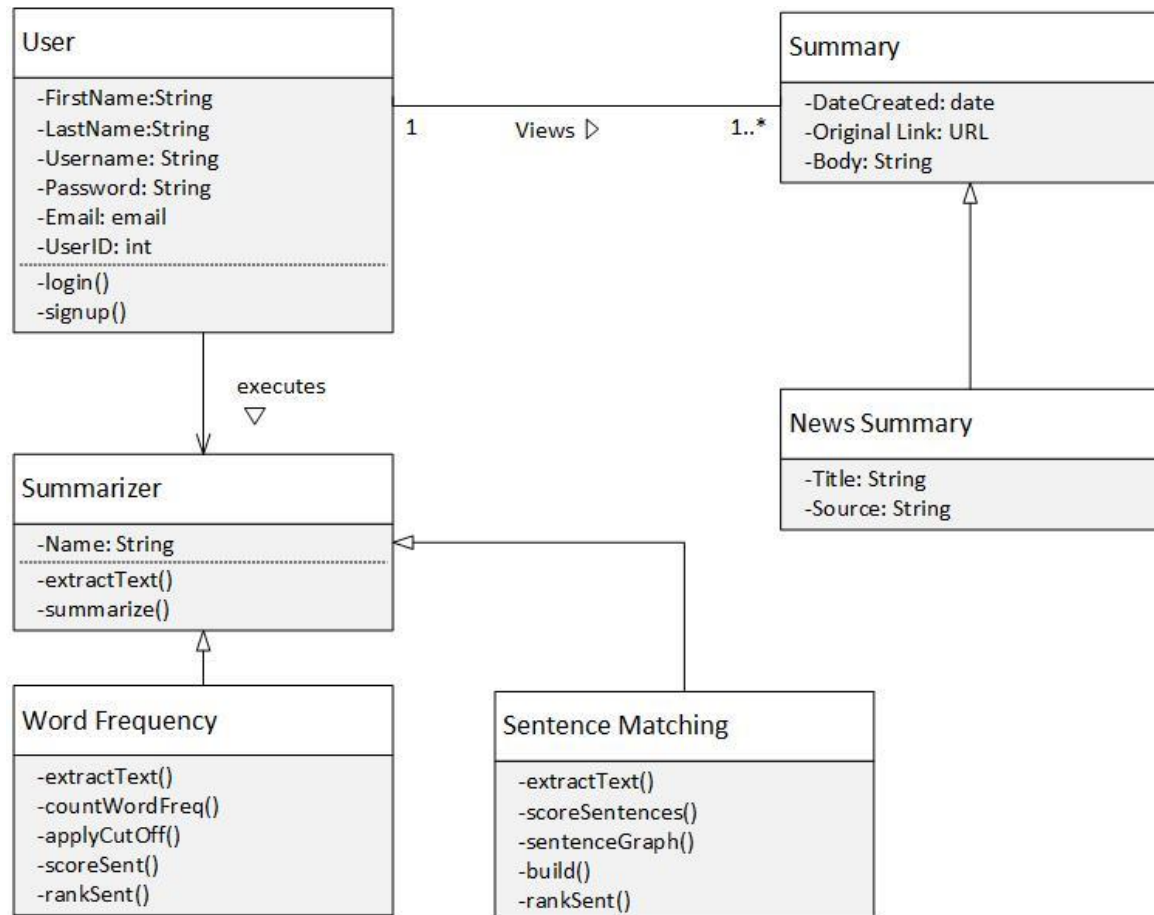


Figure 4.4: Class Diagram for Automatic Text Summarizer

4.4. Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. Sequence diagrams are sometimes known as event diagrams. The sequence diagram of our system is shown as below.

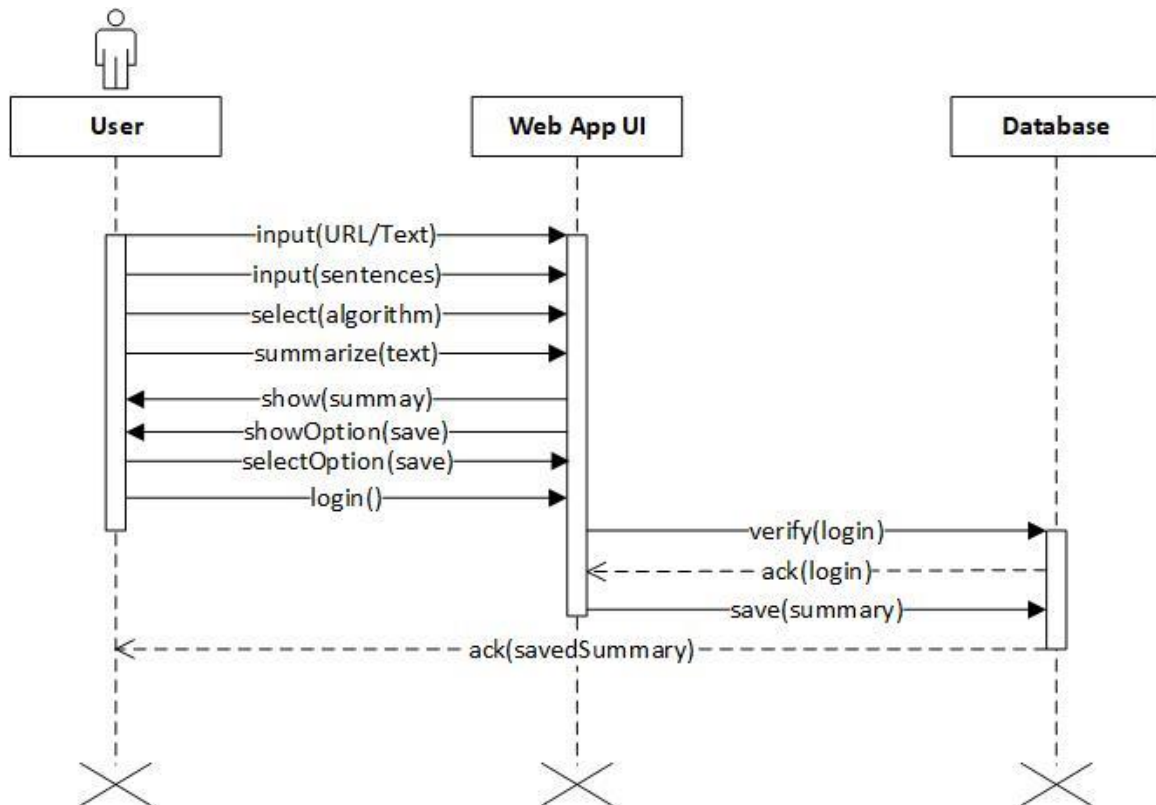


Figure 4.5: Sequence Diagram for Automatic Text Summarizer

4.5. Activity Diagram

Activity diagram is important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. The activity diagram of our system is shown as below:

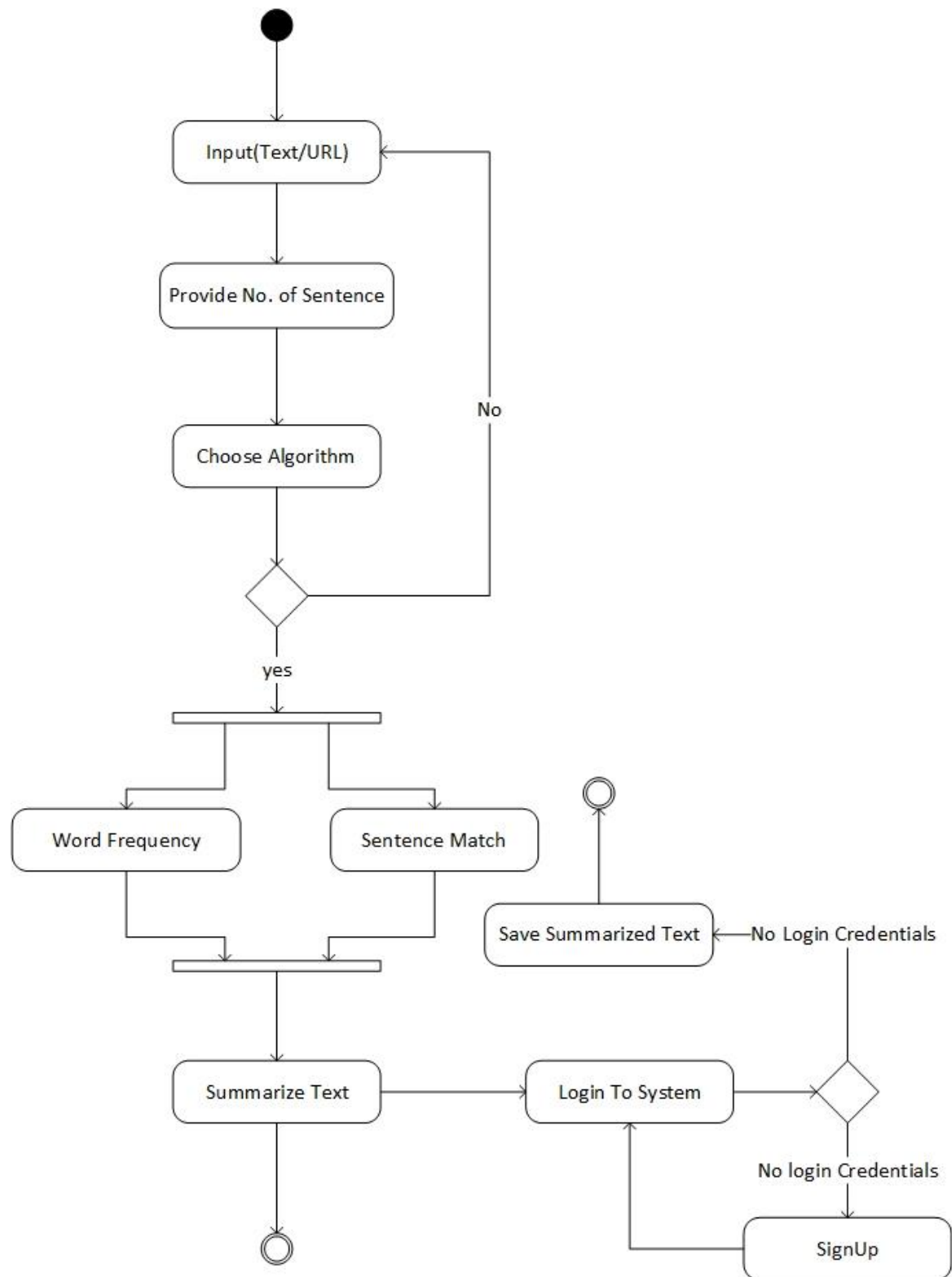


Figure 4.6: Activity Diagram for Automatic Text Summarizer

CHAPTER 5: IMPLEMENTATION

5.1. Tools Used

Following are the tools and framework used for the accomplishment of this project:

Python Language 3.6.4

Python is an interpreted high-level programming language for general-purpose programming. Python language was used for most of the backend operations such as: processing form data, authentication, summarization, pdf extraction, etc.

PyCharm 2018.1.1

PyCharm is an IDE used in computer programming, specifically for the Python language. The whole system was programmed in PyCharm including both front end and back end.

Beautiful Soup 4.6 and NLTK 3.2.5

Beautiful Soup is a Python package for parsing HTML and XML documents. It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping. It was used to extract text from the given URL.

The Natural Language Toolkit (NLTK) is a suite of libraries and programs for symbolic and statistical natural language processing. It was used for removing stop words, word tokenization and sentence tokenization in the system.

Django 2.0.6

Django is a free and open-source web framework, written in Python, which follows the model-view-template architectural pattern. In this system, model was used for database mapping, view was used for backend processing and template was used for front end design.

Jinja 2.0

Jinja is a template engine for the Python programming language. It's like client side scripting which also helps to display data contained in list and dictionary received from server. It allows to run python code in client side producing dynamic content.

Bootstrap 4.0

Bootstrap is a free and open-source front-end framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. It was used to design the User Interface components of web app like: Navigation Bar and form using built in CSS and JavaScript features.

MySQL 8.0.11

MySQL is an open-source relational database management system. It was used as a backend database engine. The database required for the system was created in MySQL.

Gensim 3.3.0

Gensim is a robust open-source vector space modeling and topic modeling toolkit implemented in Python. It was used to implement word2vec for constructing vector representation of words, also known as word embeddings.

Tensorflow 1.1

TensorFlow is an open-source software library for dataflow programming across a range of tasks. It was used to implement Seq2Seq encoder-decoder framework for training the model.

5.2. Methodology

The automatic text summarizer is based on the extractive approach which involved the selection of phrases and sentences from the source document to make up the new summary. Techniques involved ranking the relevance of phrases in order to choose only those most relevant to the meaning of the source.

There were two algorithms used in this automatic text summarizer namely: Sentence Scoring and Word Frequency. Both of these algorithms are based on the statistical approaches and worked fluently to find the desired length of summary. Below are some essential methods used in both of these algorithms:

5.2.1. Essential Methods of Sentence Scoring Algorithms

Sentence Scoring algorithm works by finding the common tokens in a pair of sentence. This is achieved in the following way:

1. Split the article into words or tokenization
2. Eliminate the stop words
3. Take two sentences and find average and score of sentences
4. Create a completely connected and weighted graph using score between all pair of sentences.
5. Calculate individual score of each sentence by summing up all its intersection with other sentences
6. Rank the sentence using Python's sorted function
7. Display the summary

Sentence scoring algorithm showed promising result in the purposed system and was used in summarizing news from popular websites and also to summarize pdf documents.

5.2.1.1. Scoring

The `score_sentences()` function receives two sentences, finds the intersection between the two i.e. the words/tokens common in both the sentences and then the result is normalized by the average length of the two sentence. The implementation was done in following way:

$$\text{avg} = \text{len}(s1) + \text{len}(s2) / 2.0$$
$$\text{score} = \text{len}(s1.\text{intersection}(s2)) / \text{avg}$$

After finding the score of sentences, a completely connected and weighted graph was created which contains scores between all the pairs of sentences in a paragraph. The function, `sentence_graph()` performed this task. Suppose `scoreGraph` is the obtained weighted graph. The `scoreGraph` consist of paired scores. So, to calculate individual score of each sentence, sum up all the intersection of a particular sentence with the other sentences in the paragraph and store the result in a dictionary with the sentence as the key and the calculated score as the value. The function, `build()` performed this task.

To build the summary from the final score dictionary, the result is first sorted and top sentences are displayed in the same order as they were present in the input text.

5.2.2. Essential Methods of Word Frequency Algorithms

Word Frequency algorithm works by finding the frequency of each word in the sentence. This is achieved in the following way:

1. Split the article into words or tokenization
2. Eliminate the stop words
3. Find how often each remaining word occurs i.e. compute frequency of each word
4. For each sentence find the score by summing up the frequency of existing words
5. Rank the sentences by that score
6. Finally display the important sentences as a summary

The tokenization and removing the stop words were quite similar in both the algorithms and those were the easy task. Following are some vital methods used in this algorithm:

5.2.2.1. Compute Frequency

This method took a list of sentences that were already tokenized and returned a dictionary that contained the frequency of particular word. The frequency of each word is normalized and if the normalized frequency is greater than and equal to the designated max_cut the particular frequency is deleted from the list or if the normalized frequency is less than and equal to min_cut, the particular frequency is also deleted from the list to maintain unbiased summary. Finally, a dictionary containing frequency of words is returned. All above task was performed by compute_frequencies() method. The implementation of above task was done in the following way:

```
m = float(max(freq.values()))  
  
freq[w] = freq[w]/m  
  
if freq[w] >= max_cut or freq[w] <= min_cut:  
  
    del freq[w]
```

The computed frequencies were sent to the summarize function which then returned a list of n sentences which is the summary of the text. The summarize function used the built-in Heap queue algorithm to rank the sentence which was done in rank method. This algorithm also included a text extractor which is shared with Sentence Scoring algorithm. The text extractor could extract text from any given URL based on the regular expressions. This text extractor removed all the HTML tags and syntax to get the pure text which is then used to summarize.

5.2.3. Methods for Generating Machine Learning Models

Abstractive method for text summarization basically involved training and testing of a machine learning model. Neural Sequence to Sequence attention models had shown promising results in Abstractive Text Summarization. The sequence to sequence architecture is widely used in the response generation and neural machine translation to model the potential relationship between two sentences. It typically consists of two parts: an encoder that reads from the source sentence and a decoder that generates the target

sentence word by word according to the encoder's output and the last generated word. Some techniques used to train the model are discussed below:

5.2.3.1. Word Embedding

Word Embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers. Conceptually it involves a mathematical embedding from a space with one dimension per word to a continuous vector space with a much lower dimension.

GloVe (Global Vectors) algorithm is used for word embedding. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

5.2.3.2. Encoder Decoder Architecture

The Encoder-Decoder LSTM (Long Short Term Memory) is a recurrent neural network (RNN) designed to address sequence-to-sequence problems, sometimes called seq2seq. Sequence-to-sequence prediction problems were challenging because the number of items in the input and output sequences can vary. The Encoder-Decoder architecture is a way of organizing recurrent neural networks for sequence prediction problems that had a variable number of inputs, outputs, or both inputs and outputs. The architecture involved two components: an encoder and a decoder.

- Encoder: The encoder reads the entire input sequence and encodes it into an internal representation, often a fixed-length vector called the context vector.

- Decoder: The decoder reads the encoded input sequence from the encoder and generates the output sequence.

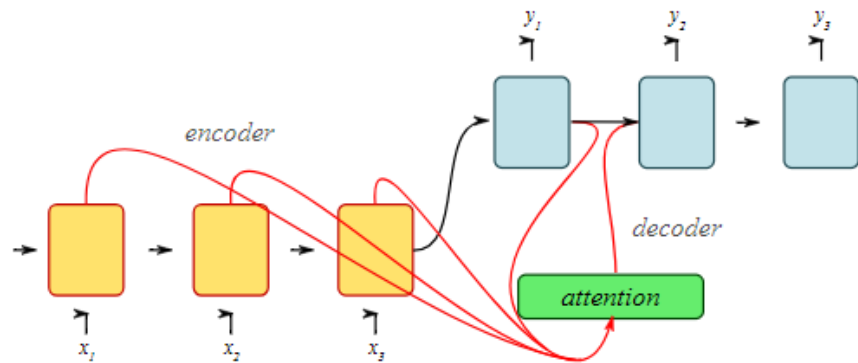


Figure 5.1: Encoder Decoder Architecture with Attention

5.2.3.3. Attention Mechanism

The entire encoded input is used as context for generating each step in the output. Although this works, the fixed-length encoding of the input limits the length of output sequences that can be generated. An extension of the Encoder-Decoder architecture is to provide a more expressive form of the encoded input sequence and allow the decoder to learn where to pay attention to the encoded input when generating each step of the output sequence. This extension of the architecture is called attention.

CHAPTER 6: SYSTEM TESTING

System testing was done by giving different training and testing datasets. This test was done to evaluate whether the system was providing accurate summary or not. During the phase of the development of the system our system was tested time and again. The series of testing conducted are as follow:

6.1. Unit Testing

In unit testing, we designed the entire system in modularized pattern and each module was tested. Till we get the accurate output from the individual module we worked on the same module. The input forms were tested so that they won't accept invalid input. Likewise, the text extraction or the feed extraction script was tested thoroughly so that it could only extract valuable text from the URL or the given website feed. Finally, all the algorithms used were tested to know the accuracy of the summary produced by the system.

Table 6.1: Unit Testing of Automatic Text Summarizer

Test Case No	Description	Expected Result	Actual Result	Status
1	URL summarization for English	Summarized English text	Summarized text obtained	Successful
2	URL summarization for Nepali	Summarized Nepali text	Summarized text obtained	Successful
3	URL summarization for Nepali	Summarized Nepali text	Summarized text with unwanted tokens	Failed
4	Upload pdf	Pdf uploaded	Pdf uploaded	Successful
5	Summarized News from popular websites feed	List of 5 summarized news from respective websites	List of 5 summarized news from respective websites	Successful

6.2. Integration Testing

After constructing individual modules all the modules were merged and a complete system was made. Then the system was tested whether the prediction given by training dataset to testing set was correct or not. We tried to meet the accuracy as higher as much as we can get. After spending a couple of days in integration testing the average accuracy of our system was 80%.

6.3. ROUGE Evaluation

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially of a set of metrics for evaluating automatic summarization of texts as well as machine translation. It worked by comparing an automatically produced summary or translation against a set of reference summaries (typically human-produced). ROUGE evaluation was carried out using English Gigaword dataset for machine learning model whereas for the Sentence Scoring and Word Frequency algorithm 17 timelines Summarization Dataset was used. The following evaluation metrics are available:

ROUGE-N: Overlap of N-grams between the system and reference summaries.

- ROUGE-1 refers to the overlap of 1-gram (each word) between the system and reference summaries.
- ROUGE-2 refers to the overlap of bigrams between the system and reference summaries.

ROUGE-L: Longest Common Subsequence (LCS) based statistics. Longest common subsequence problem takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically.

Table 6.2: Rouge Metrics for Sentence Scoring Algorithm

Rouge Measure	Precision	Recall	F-Measure
Rouge-1	0.529	0.272	0.359
Rouge-2	0.380	0.168	0.233
Rouge-L	0.529	0.272	0.303

Table 6.3: ROUGE Metrics for Frequency Word Algorithm

Rouge Measure	Precision	Recall	F-Measure
Rouge-1	0.335	0.468	0.391
Rouge-2	0.156	0.231	0.186
Rouge-L	0.290	0.404	0.320

Table 6.4: ROUGE Metrics for Machine Learning Model

Rouge Measure	Precision	Recall	F-Measure
Rouge-1	0.625	0.625	0.624
Rouge-2	0.285	0.285	0.285
Rouge-L	0.625	0.625	0.624

CHAPTER 7: CONCLUSION AND RECOMMENDATION

6.1. Conclusion

This project proposed an automatic system that could summarize a given textual data. The system developed in this project was able to summarize long textual data. The system took inputs from the user and process the input particularly to generate the summary. The user was able to provide either URL of the textual data or the long text to the system for the generation of the summary. The user was also able to select the quantity of summary by choosing the number of sentences of summary. In addition, the user was also able to select the algorithm. Two different algorithm was used in this project and both the algorithms generated the summary fluently based on different statistical method. Thereby the user could receive two different sets of summary from those two algorithms. The ultimate output of this project was a short and precise summary of a long text document. Furthermore, a machine learning model was successfully trained and tested to adopt the abstractive method for text summarization.

6.2. Recommendation and Future Scope

Text summarization is one of the leading topics in natural language processing. Many researches have be done in this field and the future of text summarization is bright. Keeping that in mind the text summarization proposed in this report has a wide future scope. The text summarizer can be used to display a list of trending news in websites. Likewise, this text summarizer can be subjected to API, which can be used by other third party application to generate desired result. In addition, the ML model could be deployed on production in upcoming days. The shortcomings of this machine learning model need to be addressed.

REFERENCES

- [1] J.-M. Torres-Moreno, Automatic Text Summarization (Cognitive Science and Knowledge Management), Montréal: Wiley-ISTE, 2014.
- [2] J. Brownlee, "A Gentle Introduction to Text Summarization," 29 11 2017. [Online]. Available: <https://machinelearningmastery.com/gentle-introduction-text-summarization/>.
- [3] P. J. L. C. D. M. Abigail See, "Get To The Point: Summarization with Pointer-Generator Networks," Cornell University Library, 14 4 2017. [Online]. Available: <https://arxiv.org/abs/1704.04368>. [Accessed 19 4 2018].
- [4] S. C. J. W. Alexander M. Rush, "A Neural Attention Model for Abstractive Sentence Summarization," Cornell University Library, 2 9 2015. [Online]. Available: <https://arxiv.org/abs/1509.00685>. [Accessed 19 4 2018].

APPENDIX


Outputs:

The screenshot shows the homepage of the Automatic Text Summarizer App. The header is blue with the ATS logo and navigation links: Automatic Text Summarizer, Summarize PDF, BBC, CNN, Nagarik, and a Sign In link. The main content area has a light gray background. On the left, there's a form with the following fields: 'Input Page URL' with a placeholder 'Paste Complete Valid URL', 'OR Paste Copied Text' with a placeholder 'Paste Long Text Here..', 'No. of Sentences' with a value of '5', and 'Select Algorithm' with a dropdown menu showing 'Sentence Scoring'. A blue 'Summarize' button is at the bottom of the form. On the right, there's a list of features: 'Summarize Text From Paragraph/URL', 'Save Summarized Text', 'Summarize PDF to Text File', and 'View Summarized News'. Below the list, there's a quote: 'Use Automatic Text Summarizer when you have less time for reading.' The footer is blue with the text '© 2018 Automatic Text Summarizer | ATS'.

Figure A: Homepage of Automatic Text Summarizer App

The screenshot shows the output of the Automatic Text Summarizer App. The header is blue with the ATS logo and the text 'Automatic Text Summarizer' and 'Summarize longer text online and save your time'. The main content area has a light gray background. On the left, there's a section titled 'Title' with the text '7 Tips To Look Handsome | Male Grooming Guide - ManStyleBox'. On the right, there's a section titled 'Summary' with the text: 'So, if you want to impress ladies, it's all about looking clean. Now when you have managed your facial hair, it's time to deal with your face. It's very simple, just keep your nails short.'

Figure B: Demonstration of Sentence Scoring Algorithm for URL



Automatic Text Summarizer

"Summarize longer text online and save your time"


Title

Haircuts affect your face structure, but more importantly, they affect how scruffy you look especially if you've got hair like mine which is a little bit rough it's not a great idea to grow out your hair unless you've got silky smooth hair. If you want to look neat and classy at the same time go for the golden haircut that's a slightly longer top and a slightly short side. Think of most good-looking athletes like Cristiano Ronaldo, Olivier Giroud, and Robin Van Persie they all follow this protocols with different ratios. What do I mean by ratios? Well, some people go for slightly longer tops and short sides and some people prefer slightly longer sides and not such long tops but at the end of the day, it's a basic blueprint. You need to figure out a haircut that suits your face and face structure perfectly but the most important rule is to just look neat and tidy. Remember women associate cleanliness with looking good and that's why we're moving on to tip number two.

Summary

If you want to look neat and classy at the same time go for the golden haircut that's a slightly longer top and a slightly short side. Well, some people go for slightly longer tops and short sides and some people prefer slightly longer sides and not such long tops but at the end of the day, it's a basic blueprint. You need to figure out a haircut that suits your face and face structure perfectly but the most important rule is to just look neat and tidy.

Figure C: Demonstration of Sentence Scoring Algorithm for long text



Automatic Text Summarizer

"Summarize longer text online and save your time"


Title

7 Tips To Look Handsome | Male Grooming Guide - ManStyleBox

Summary

Bonus skin care routine is only to be used on occasion like if you are going for a party or an important meeting and you don't want to look oily and grungy that's when you can use this bonus skin care tips. Haircuts affect your face structure, but more importantly, they affect how scruffy you look especially if you've got hair like mine which is a little bit rough it's not a great idea to grow out your hair unless you've got silky smooth hair.

Figure D: Demonstration of Word Frequency Algorithm for URL



Automatic Text Summarizer

"Summarize longer text online and save your time"

Title

Haircuts affect your face structure, but more importantly, they affect how scruffy you look especially if you've got hair like mine which is a little bit rough it's not a great idea to grow out your hair unless you've got silky smooth hair. If you want to look neat and classy at the same time go for the golden haircut that's a slightly longer top and a slightly short side. Think of most good-looking athletes like Cristiano Ronaldo, Olivier Giroud, and Robin Van Persie they all follow this protocols with different ratios. What do I mean by ratios? Well, some people go for slightly longer tops and short sides and some people prefer slightly longer sides and not such long tops but at the end of the day, it's a basic blueprint. You need to figure out a haircut that suits your face and face structure perfectly but the most important rule is to just look neat and tidy. Remember women associate cleanliness with looking good and that's why we're moving on to tip number two.

Summary

Haircuts affect your face structure, but more importantly, they affect how scruffy you look especially if you've got hair like mine which is a little bit rough it's not a great idea to grow out your hair unless you've got silky smooth hair. Well, some people go for slightly longer tops and short sides and some people prefer slightly longer sides and not such long tops but at the end of the day, it's a basic blueprint.

Figure E: Demonstration of Word Frequency Algorithm for long text

ATS Automatic Text Summarizer
Summarize DocumentBeta
BBC CNN Nagarik

ATS
Automatic Text Summarizer
NP

"
Don't waste your time reading long texts.
"

Original Text/URL

प्रधानन्यायाधीशमा ओमप्रकाश मिश्रलाई सिफारिस गरेको छ। आज बसेको परिषदको बैठकले मिश्रलाई सर्वसम्मत सिफारिस गरेको हो। यस अघि परिषदले दिपकराज जोशीलाई प्रधानन्यायाधीशमा सिफारिस गरे पनि संसदीय विशेष सुनुवाई समितिले अस्वीकार गरेको थियो। जोशी अस्वीकृत भएपछि विदामा बसेका छन् भने कायममुकायमरूपमा मिश्रले काम गर्दै आएका छन्। प्रधानमन्त्रीको अध्यक्षतामा रहने परिषदमा प्रधानन्यायाधीश, प्रतिनिधिसभाका सभामुख, राष्ट्रिय सभाका अध्यक्ष, प्रतिनिधिसभाका विपक्षीदलका नेता र प्रतिनिधिसभाका उपसभामुख सदस्य रहने संवैधानिक व्यवस्था छ। प्रधानन्यायाधीश सिफारिस गर्ने बैठकमा कानुन मन्त्री पनि सहभागी हुने व्यवस्था छ।

Summary

प्रधानन्यायाधीशमा ओमप्रकाश मिश्रलाई सिफारिस गरेको छ। आज बसेको परिषदको बैठकले मिश्रलाई सर्वसम्मत सिफारिस गरेको हो। यस अघि परिषदले दिपकराज जोशीलाई प्रधानन्यायाधीशमा सिफारिस गरे पनि संसदीय विशेष सुनुवाई समितिले अस्वीकार गरेको थियो। जोशी अस्वीकृत भएपछि विदामा बसेका छन् भने कायममुकायमरूपमा मिश्रले काम गर्दै आएका छन्। प्रधानमन्त्रीको अध्यक्षतामा रहने परिषदमा प्रधानन्यायाधीश, प्रतिनिधिसभाका सभामुख, राष्ट्रिय सभाका अध्यक्ष, प्रतिनिधिसभाका विपक्षीदलका नेता र प्रतिनिधिसभाका उपसभामुख सदस्य रहने संवैधानिक व्यवस्था छ। प्रधानन्यायाधीश सिफारिस गर्ने बैठकमा कानुन मन्त्री पनि सहभागी हुने व्यवस्था छ।

Save Summary

Figure F: Demonstration of Sentence Scoring Algorithm for Nepali text

ATS Automatic Text Summarizer
Summarize DocumentBeta
BBC CNN Nagarik

Top 5 Headlines From 'Nagarik'

विश्वकर्मालाई रिहा गर्न सर्वोच्चको आदेश, परिसरबाटै पक्राउ

काठमाडौँ - सर्वोच्च अदालतले नेत्र विक्रम चन्द नेतृत्वको नेकपाका प्रवक्ता खड्गबहादुर विश्वकर्मालाई रिहा गर्न आदेश दिएको छ। न्यायाधीशद्वय केदारप्रसाद चालीसे र पुरुषोत्तम भण्डारीको संयुक्त इजलासले विश्वकर्मालाई सर्वोच्चको रजिष्ट्रारको रोहबरमा छाड्न आदेश दिएको हो। सर्वोच्चको आदेशमा रिहा भएका उनलाई प्रहरीले पुन पक्राउ गरेको छ। उनलाई भोजपुरमा भएको बम विष्फोटको आरोप लगाइएको छ। विश्वकर्मालाई प्रहरीले भदौ १ गते सर्वोच्च अदालत परिसरबाट दोश्रो पटक अगजनी मुद्दामा पक्राउ गरेको थियो। विश्वकर्मालाई गैरकानुनी रूपमा थुनामा राखिएको भन्दै उनका भाइ विरेन्द्र विकले गत आइतवार रिहाइको मागसहित सर्वोच्चमा बन्दी प्रत्यक्षीकरणको रिट निवेदन दर्ता गराएका थिए। यसअघि ७ करोड बराबरको चन्दा असुली अभियोगमा विश्वकर्मा साउन २२ गते चाल्नाखेलबाट पक्राउ परेका थिए। तर उनीमाथि लागेका आरोप पुष्टि हुने आधार नदेखिएको भन्दै सर्वोच्चले रिहा गर्न आदेश दिएको थियो। रिहा भएलगत्तै महानगरीय प्रहरी परिसर काठमाडौँले अदालत परिसरबाटै विश्वकर्मालाई पक्राउ गरी कम्प्रे प्रहरीलाई जिम्मा लगाएको थियो।

यो कस्तो चाला हो? (फोटो फिचर)

काठमाडौँ आइपुग्ने विशिष्ट अतिथिहरु हिड्ने सडकलाई सरकारले अहिले मर्मत गरिरहेको छ। मर्मतको नतिजा हेर्नुहोस् तस्वीरमा। विमानस्थलमा ओर्ले पछि अतिथिहरु हिड्ने सडकलाई चिटिक्क पार्न गएको साता देखि बरल काम थालिएको छ। खाल्डा परेका ठाउँलाई पिच गर्ने र सडक किनारमा पेन्टिङ गरेर चिटिक्क पारेर देखाउने प्रयासमा सरकार सक्रिय छ। तर अहिले मौसम सक्रिय भएका बेला, सडक कालोपत्रे गर्ने एवं अन्य मर्मतका काम धमाधम भइरहेका छन्। गएको बैशाख जेठमा यी काम गरेको भए सहज हुने जानकार बताउँछन्। अर्कोतर्फ चावहिल जोरपाटी सडकलाई टालटुल पार्न पनि नभ्याएपछि गोकर्ण रिसोर्टमा सरकारका तर्फबाट आगन्तुकलाई दिइने रिसेप्सन हुने नहुने दुंगो लागेको छैन। यतिवेला विमानस्थलदेखि होटल सोल्टी, होटल ह्याट, सिंहदरवार हुँदै बाजुवाटार र राष्ट्रपति निवाससम्मका सडक टालटुल गरेर रंगरोगनसहित चिटिक्क पार्ने प्रयास भइरहेको छ।

नेपालमा अब दन्त पर्यटन !

चितवन-अमेरिका र युरोपलागयतका विकसित देशमा दाँतको उपचार निकै महँगो हुने गर्दछ । नेपालमा करिब ७० हजार रुपैयाँमा हुने हुने दाँत प्रत्यारोपण ती देशमा पाँचदेखि सात हजार डलर खर्च हुन्छ । ती देशमा दाँतको उपचार गर्ने पैसाले नेपालमा उपचारसँगै आकर्षक पर्यटकीय गन्तव्य नै घुमेर फर्किन पुग्ने दन्त चिकित्सक बताउँछन्। युरोप, अमेरिका, अष्ट्रेलिया, बेलायतजस्ता विकसित देशमा दाँतको उपचार अत्यधिक महँगो पर्छ । उनले भनिन, "विकसित देशमा हुने दाँतको उपचारको गुणस्तर हामीले दिने गर्दछौँ ।" सो संस्थाले हालसम्म ३५ हजारभन्दा बढी बिरामीको उपचार गरिसकेको छ । विकसित देशमा दाँतको बीमा नहुने हुँदा उपचार महँगो पर्ने गर्छ। कम्बोडिया, थाइल्याण्ड, भारतजस्ता देशले पर्यटक लक्षित दाँतको अस्पताल शुरु गरिसकेका छन् ।

Figure G: Demonstration of Summarized Nepali News from Website

ai.txt - Notepad

File Edit Format View Help

Original Text: cow is like a mother for us as it gives us milk two times a day it cares us and nourishes us through its healthy and nutritious milk it is found in almost every regions of the world almost everyone keeps cow at home to get fresh and healthy milk daily it is very important and useful domestic animal

Gnereated Text: milk < unk > found in us cows

Figure H: Demonstration of summary generated by ML model

31