A Major Project Final Report on

**Article Summarizer**

Submitted in Partial Fulfillment of the Requirements for the Degree of

**Bachelors of Engineering in Software Engineering**

Under Pokhara University

Submitted by:

**Priyesh Gautam, 15723**

**Divij Devkota, 15741**

**Anup Binod Khanal, 15799**

Under the supervision of

**Mr. Chandra P. Bhatta**

…………………..

Date:

**December 31st, 2019**

|  |  |
| --- | --- |
|  | **Department of Software Engineering**  **NEPAL COLLEGE OF**  **INFORMATION TECHNOLOGY**  Balkumari, Lalitpur, Nepal |

# ACKNOWLEDGEMENT

We would like to express our sincere thankfulness and gratitude to all of those who helped us during this project at several different phases to lead it to completion. We owe special thanks to a number of people who have devoted much of their time and expertise, without which it would have been very difficult for us to complete our project “Article Summarizer”.

We would like to take the opportunity to express our humble gratitude to our supervisor **Mr. Chandra. P. Bhatta**, under whom we undertook our project. His constant guidance andwillingness to share his vast knowledge made us enhance our knowledge and helped us to complete the tasks. Without his effort and full support this project may not have succeeded.

Moreover, we would like to extend our special thanks to **Dr. Roshan Chitrakar** for providing technical support with suggestions which helped our project to foster. It would be our privilege to thank **Er. Birendra Bista**, Head of Department of Software Engineering for his support. We feel obliged in submitting this project as part of our curriculum.

Priyesh Gautam

Divij Devkota

Anup Binod Khanal

Nepal College of Information Technology,

Balkumari, Lalitpur

**Abstract**

Data analytics and data manipulation has been the major topic of this technological era. The need of data manipulation is increasing every year, where the need for data filtration, indexing and summarization is ever increasing. The project entitled Article Summarizer is an application that enables to summarize multiple articles to a brief description. This application accepts articles collected from different sources on the basis of an input keywords provided and manipulates the collected articles to obtain a brief summary of that particular topic. In this beta version, it accepts only up to 10 articles which will be extracted by a web crawler. For this project, two techniques have been employed i.e. “The Centroid Based Method (CBM)” and “Graph Based Method (GBM)”. Comparisons between results have been obtained by implementing only one of the mentioned techniques and also by implementing both techniques, in which, the result obtained by implementing both techniques have been considered as an optimum and efficient result. This application enables to save time and effort of an individual willing to obtain a summary from the collection of multiple articles. The idea of this project can also be implemented in other areas such as in Search Engine Optimization, sampling and so on.

Keywords: Article Summarization, Centroid-Based Method, Graph Based method, Multi-document Summarizer, Iterative SDLC method.

Table of Contents

[ACKNOWLEDGEMENT I](#_Toc28688962)

[Table of Figures: IV](#_Toc28688963)

[1 INTRODUCTION 1](#_Toc28688964)

[1.1 Problem Statement: 2](#_Toc28688965)

[1.2 Project Objectives: 2](#_Toc28688966)

[1.3 Project Scope and Limitation: 3](#_Toc28688967)

[1.4 Significance of Study 4](#_Toc28688968)

[2 LITERATURE REVIEW 5](#_Toc28688969)

[3 Proposed Methodology 7](#_Toc28688970)

[3.1 Iterative Model: 7](#_Toc28688971)

[3.2 Incremental phases of the project 9](#_Toc28688972)

[4 Proposed Performance Analysis: 11](#_Toc28688973)

[5 Requirement Analysis 12](#_Toc28688974)

[6 DESIGN 13](#_Toc28688975)

[6.1 Use-case Diagram 13](#_Toc28688976)

[6.2 Activity Diagram 14](#_Toc28688977)

[7. Time Schedule: 15](#_Toc28688978)

[7 TESTING AND REVIEW 16](#_Toc28688979)

[8 CONCLUSION 17](#_Toc28688980)

[9 RECOMMENDATIONS 18](#_Toc28688981)

[8. Reference 19](#_Toc28688982)

# Table of Figures:

[Figure 1: Iterative process 5](#_Toc26221235)

# List of tables

[Table 1: Tools and technologies used 10](#_Toc28689788)

[Table 2: Time Schedule 15](#_Toc28689789)

[Table 3: Devices used for testing the application 16](#_Toc28689790)

# INTRODUCTION

Multi-document summary is a process that aims to automatically extract information of a given query or a set of keywords from multiple sources using the internet as well as from the data that has been saved on the local machine. Initially it was designed to summarize static data that would be saved on the local memory of the machine, with further addition of APIs and the concepts of web-crawling, the application is capable of providing a summarized information about a user-input query that is related with a particular topic. The text summarization application has been designed with a goal of creating or extracting a summary or a short-concise document that describes the content of a given topic or an input query. The summary generated from the application enables the user to get familiar with the processed information from a large bundle of documents or contents. The automatic multi-document summarizer is capable of extracting descriptive and detailed information on the related topic. With various feelings being assembled and illustrated, every theme is portrayed in a solitary passage from numerous points of view, while the point of a concise rundown is to rearrange the scanned data and spare time by indicating the most pertinent source records, an exhaustive multi-report synopsis ought to contain the required data itself, along these lines restricting the need to get to unique documents to situations where refinement is important. Programmed rundowns present algorithmically, without article contact or abstract human mediation, data separated from various sources, subsequently making it totally fair. The program depends largely on the algorithms without any human mediation and makes a sensible summary.

## Problem Statement:

Data manipulation and data analytics has been one of the major topic in this technological era. Many information and data need to be filtered, synchronized or summarized into a concise form in order to obtain only the necessary data. Automatic Summarization has also been a major topic for data analytics as it enables to find keywords and converts large chunks of data to a concise form. According to research context, researching on a particular topic and summarizing them is much of a difficult task, as one has to go through many redundant and unnecessary information. One may even wish to find key words on the article which may be difficult without using the methodologies of the article summarizer. There is also a need of indexing in a document. The human made summary may not be concise as well as it may be biased, such problems are handled by the summarizer. There may also be a need of a provision for document manipulation for various commercial sectors. Hence, the Multi-Document summarizer enables to handle all such issues.

## Project Objectives:

With the implementation of the two main algorithms, the application enables:

* To reduce summarization time of any article or document
* To make the process of researching on a document easy.
* To support the process of Data analytics and data manipulation.
* Generate less biased summary than actual human made summary
* Obtain personalized summaries as they are useful since they provide personalized information in question-answering systems.
* To avoid unnecessary redundant information as far as possible.
* The use of automatic or semi-automatic summarization systems that enables to abstract commercial services to increase the number of texts they can process.

## Project Scope and Limitation:

Automatic data summary is part of data mining and machine learning. The main idea of summary is to find a subset of data that contains the whole set's "information." Today in industry, such techniques are widely used. Search engines are an example: document summaries, image collections and videos. Automatic text formulation is the problem of data science creating a short, efficient and fluent summary from a longer document. To consume the ever-increasing amount of text data available online, summary methods are much needed. The scope of the project is to develop an automatic multi document summarization in order to save the valuable time and to obtain an efficient brief description of a particular topic. The summarization is able to present a grammatically stable summary about any query found through the duck-duck go search engine. It captures the most relevant information found in the number of sources and through multiple process gives us the valid result.

However, there may be limitation of this application such as:

* Also has a vulnerability of getting attacked from multiple viruses that are attached to the articles or documents, while extracting information from the internet.
* Lack of ability to show the category of particular topic or query.
* The summary may or may not be precise enough to describe the topic.
* Need of few other algorithms to make the summary accurate enough to meet the user’s standard.

## Significance of Study

The study on this concept provides a platform for research as well as big support in the field of Data Analytics and Data science. As we know, modern generation heavily relies upon the manipulation of data, as there will be tons of unnecessary data that need to be filtered, the concepts involved in the project ‘Article Summarizer’ will be much fruitful for enhancing the field of data analytics. The Application applies two main algorithms on a set of particular document or an article to obtain a concise and meaningful summary. In the context of the internet, there are hundreds of article related to a certain topic, and if one wishes to obtain a concise summary of that particular topic, they will have to go through every article, surfing through unwanted, redundant information. Hence, the “Multi-Document summarizer” or also called as “Article summarizer” enables to avoid such difficult activity by summarizing a set of related documents or articles into a concise sensible summary. Automatic summary generation of a query string is part of data-mining, data-analytics and machine learning. The main idea of the application is to find a subset of data that contains the whole set's "information." In today’s industry, such techniques are widely used. Search engines can also be taken as major example, where as some other examples are: document summaries, image collections and videos.

# LITERATURE REVIEW

This section consists of the constructs and concepts used for the development of this system. Many approaches have been proposed in the project of filtering relevant article from the noisy web page. For instance: learning from machines, scrapping based on rules, etc. Application of some of the above concepts by open source APIs is readily available. With content range erasure the framework endeavors to erase "less significant" ranges of content from the first document; the content that remaining parts is esteemed a summary. In general, there are two methods of achieving automatic summarization namely:

Abstractive Summarization: Abstractive techniques provides paraphrasing and shortens the parts of the source document. It also overcomes the grammatical inconsistencies of the extractive model. (2) It involves advanced deep learning techniques and sophisticated language modeling. It performs better than the Extractive summarization method.

Extractive Summarization:It produce summaries by selecting a subset of sentences from all the documents. There are a number of methods that can be used to select this subset. It usually involves weighing the essential sections of the sentences and using the results to generate summaries.

Here, we have incorporated Extractive Summarization methodology to implement the algorithm for summarizing the text contents. The two algorithms we have incorporated are:

Centroid based Algorithm:

* Create a centroid of words from the document cluster that are most central i.e., words having *tf × idf* values greater than a predefined threshold .  
   Tf = total frequency of each word ,   
   idf = log(N/Ni) where N is the total number of documents and Ni is the number of documents in which the word is present
* The threshold was obtained statistically.
* Sentences that have higher intersection with the centroid are considered more important.
* The intersection with the centroid was assigned as the score of each sentence.
* Ranks were assigned to each sentence based on their corresponding scores.

Graph based Algorithm:

* Create a graph in which each node represents a sentence and each edge represents connectivity between two sentences based on their similarity.
* Relation between two sentences is determined by the cosine similarity between the feature vector of each sentence.
* Threshold to discard low similarity values was obtained statistically.
* Scores are given to each sentence which is the number of nodes connected to that sentence.
* Each sentence was assigned a rank based on their scores.

# Proposed Methodology

In order to develop an application or to carry out any project successfully there is a need of a particular methodology which enables to manage the development process systematically and efficiently, such methodology work as a backbone to the development process that manages the efficiency of the developers and their effort. The methodology that our team will implement is the Iterative SDLC methodology that provides various advantages such as:

* Early detection of bugs and errors.
* Parallel development can be planned.
* Easily adaptable to the changes required in the project.
* Proper focus on all the components of the project.

## Iterative Model:

It is one of the most commonly used SDLC, as it breaks down the development process of a large software application into smaller pieces. Mostly, it focuses on simplifying initial implementation, which progressively gains more complexity and a broader feature until the final system is complete.

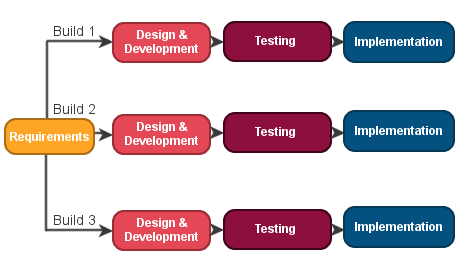


Figure 1: Iterative process

3.1.1. Requirement Analysis

In this phase, analysis will be performed in order to find out the requirements of the system. The initial goal at the beginning of the project is to identify and understand the fundamentals of the domain that the system is addressing, and throughout the project this model will be used to reflect how the construction of the application is being carried out.

3.1.2. Design & Development

In this phase the system requirements specifications would be translated into system design. Context diagram, data flow diagram, entity-relation diagram, use-case diagram, sequence diagram and class diagram are developed. This phase starts with the process of refining the class diagrams which were developed in the first phase of the model. After this process an object model is developed of the system and the features are modeled. This step requires the most time and effort, more than the other steps of this lifecycle combined.

3.1.3. Coding Phase

In this phase coding is done according to the design and a working system is developed by the end of the process. The detailed design created in the phases ahead of this are iteratively programmed, tested and packaged. At the end of this phase small workable module will be ready to use.

3.1.4. Testing Phase

In this phase the system is tested with each testing list of changes to the system developed and is suggested and the changes are applied to the software and the software is delivered as a successive increment until a satisfying system is achieved.

## Incremental phases of the project

* INCREMENT 1: Developed User Interface and researched about the algorithms with the help of the objectives of our project that focused on analysis, design, coding and testing phase of the web app modules of the system which helped us to figure every aspects of the app.

In this phase several artifacts were produced and they are as follows:

* Use case Diagram
* Feasibility Study
* Software Architecture Document
* Activity Diagram
* INCREMENT 2: In this incremental phase we integrated the user-interface with the two main algorithms and added several static collection of documents where the algorithms could be applied to obtain a concise summary.
* INCREMENT 3: This incremental phase involved the research of implementing the concepts of “web-crawler” which depends upon the internet and provides user with the ability to obtain concise summary based upon the input query or keyword provided by the user.
* INCREMENT 4: The main aspects of this incremental phase involves the implementation of the two APIs that enables the application to provide concise summary by browsing the internet. This phase also involves the overall integration of the system, checking the overall performance of the application and its architecture. It also involved the documentation process.

## Tools and technologies used

Table 1: Tools and technologies used

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Tools/Language/Software** | **Purpose** |
| **1.** | Net beans | Code Editor IDE |
| **2.** | Java | To Integrate the algorithms and the concepts of summarization |
| **3.** | Microsoft windows 10 | OS for Development and testing |
| **4.** | MS word/MS PowerPoint | Documentation and slide preparation |

# Proposed Performance Analysis:

After the completion of the development, the testing was performed to ensure the performance of the application. At first, sample documents or collection of articles were saved on the local machine to obtain the summary of that particular keyword provided by the user which is related with that collection of documents saved on the local machine. For the case of such static data, the process involves implementation of two main algorithms, i.e. the ‘Centroid Based Algorithm’ and ‘Graph Based Algorithm’. If a process involves summarization of documents saved on the local machine, then the involvement of these two algorithms will obtain the concise summary based upon the keyword provided by the user.

At the second phase, two APIs were implemented in the application i.e. Duck-Duck Go API and Boiler-Pipe API to add the feature of web-crawling. After the implementation of those two APIs, the application is capable of browsing on the web on the basis of keyword provided by the user, where the Duck-Duck-Go API enables to browse through web and the Boiler-Pipe API enables to extract only the relevant text contents, related to the keyword provided by the user, ignoring the unnecessary contents such as headers, comments and other irrelevant section from the web pages. Various keywords are entered to check the synchronization of these APIs. After the text contents are obtained, all the collected contents go through the process of indexing and query processing and then the final concise form of summary is obtained.

# Requirement Analysis

The requirements that are necessary for the development of the application were listed so as to save time during the development process and also to enable the smooth construction of the application. Requirement analysis focuses on the operational part of the software and establishes constraints that are to be fulfilled after completion of the application. The requirements are given below:

* User should be able to obtain the summary of a static data by saving the collection of data on to the local machine.
* Users should be able to obtain the summary of a particular topic using the internet where they provide the name of the topic in the form of a query or keywords.
* Users should be able to express the keywords properly in the application.
* Users should be provided with the ability to select the size of the summary as well as to select the number of links that will be used to obtain the summary.

# DESIGN

Visualization of the workflow or working of any application enables to develop the application in much more efficient and convenient way. It describes all the aspects of an application. Some diagrams used for the development of this application is given below:

## Use-case Diagram

The use-case diagram provides graphical description of the working of the elements of the application. It enables to identify, clarify and organize system requirements. We have implemented a simple use-case diagram that summarizes only the main relations between use cases, actors and the system.

User can input query or a set of keywords to get summary using the internet

User can save the collection of data on the local machine and obtain concise summary using only the algorithms

Users should be able to the provide limitations on selecting the number of articles as well as size of the article.

Users must be able to cancel the summarization process in an easier way.

## Activity Diagram

Activity diagram is defined as a UML diagram that focuses on the execution and flow of the behavior of a system instead of implementation. Here the activity diagram of the system is given below:

Select summary method, i.e. whether from static or dynamic data

Provide the summary after user inputs a keyword or a query

Provide the summary after selecting a static data from the local machine

If dynamic, provide an area to input query or keyword

If static, provide options of static data saved on the local machine

If static or dynamic method

# Time Schedule:

The time schedule for system development has been planned and divided according to its necessities and further limitations. Time has been divided depending on the load of work among the team members using iterative SDLC. The whole project and testing has been allocated a month time period in which documentation has been carried out constantly to make sure we fulfill all project requirements.

Table 2: Time Schedule

|  |  |
| --- | --- |
| TASK | APPROXIMATE DURATION ( in days ) |
| Feasibility Study | 5 |
| Develop Overall Model | 8 |
| Build feature list | 5 |
| Design by feature | 7 |
| Build by feature | 15 |
| Develop Documentation | 40 |

# TESTING AND REVIEW

This section provides the description about the devices used to test the application’s functionality and the review over the testing. The following table provides the list of devices used to test the application.

Table 3: Devices used for testing the application

|  |  |  |
| --- | --- | --- |
| SN | Device Name | Specifications |
|  |  |  |
| 1. | HP | Manufacturer: HP |
|  |  | Processor: AMD E-500 APU with Radeon™ HD Graphics 2.0 GHz |
|  |  | Installed memory(RAM):4.00 GB |
| 2. | Dell | Manufacturer: Dell |
|  |  | Processor: Intel Core™ i7-6500HQ CPU @2.30GHz |
|  |  | Installed memory(RAM):8.00 GB |
|  |  |  |

# CONCLUSION

The usage of this application enables the user to summarize chunks of text contents into a concise summary. This application will be capable to save time and effort of the users to obtain concise information about any particular topic using either the Web-crawler method or the static data summarization method (i.e. using only the two algorithms). The application “Article-summarizer” is considered as the major research in the field of data analytics and data science, so further research in the concepts of this application will be much fruitful as it can be applied in various sectors such as medical, military, education, finance, robotics and so on. Hence, this application can be considered as a research in the field of Natural Language processing and data science which can be extended to achieve greater discoveries in future.

# RECOMMENDATIONS

Due to various factors such as time limitations and complexity of algorithms, the application still has many places that need improvement. The section provides some recommendations that can enhance the application further into its better form. Some recommendations as mentioned below:

* The accuracy of the summary can be improved by integrating additional algorithms that will summarize the cluster of texts efficiently.
* Additional APIs can be implemented to obtain more accurate data in much quicker time period.

# Reference

1. En.wikipedia.org. (2019). *Multi-document summarization*. [online] Available at: https://en.wikipedia.org/wiki/Multi-document\_summarization [Accessed 20 May 2019].
2. <https://blog.floydhub.com/gentle-introduction-to-text-summarization-in-machine-learning/>
3. <https://www.digitalvidya.com/blog/the-top-5-clustering-algorithms-data-scientists-should-know/>
4. <https://towardsdatascience.com/a-quick-introduction-to-text-summarization-in-machine-learning-3d27ccf18a9f>

# Appendix

