TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

HIMALAYA COLLEGE OF ENGINEERING

[CT-755]

A

FINAL YEAR PROJECT

ON

TECH ARTICLE AGGREGATOR AND SUMMARIZATION SYSTEM

BY:

ANISH RIJAL (25704)

SANAM GHIMIRE (25729)

SNEHEE MAHARJAN (25736)

SONAL ADHIKARI (25737)

A PROJECT SUBMITTED TO DEPARTMRENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR BACHELOR'S DEGREE OF COMPUTER ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

April, 2022

TECH ARTICLE AGGREGATOR AND SUMMARIZATION SYSTEM

BY:

ANISH RIJAL (25704)

SANAM GHIMIRE (25729)

SNEHEE MAHARJAN (25736)

SONAL ADHIKARI (25737)

PROJECT SUPERVISOR:

Er. Ramesh Tamang

A report submitted in partial fulfillment of the requirements for the degree of Bachelor in Computer Engineering

Department of Electronics and Computer Engineering

HIMALAYA COLLEGE OF ENGINEERING

Tribhuvan University

Lalitpur, Nepal

April, 2022

ACKNOWLEDGEMENT

We would like to express our gratitude to the department of Electronics and Computer Engineering, our head of the department associate professor **Er. Ashok Gharti Magar**, DHOD **Er. Devendra Kathayat** and IOE for providing us the opportunity to do this major project.

We are very grateful to our project coordinator **Er. Narayan Adhikari Chettri** and our major project supervisor **Er. Ramesh Tamang** who guided and motivated us throughout the project.

We are also thankful to **Er. Suroj Maharjan, Er. Madhu Nyoupane, Er. Sudharsan Subedi** and **Er. Himal Chandra Thapa** for providing us the support and insight in our project. We would also like to thank our friends and seniors for helping us in the project.

We have written this report in hopes that we can hear the reviews of our project and also provide us the constructive suggestion for further improving our project.

Group Members

Anish Rijal (25704)

Sanam Ghimire (25729)

Snehee Maharjan (25736)

Sonal Adhikari (25737)

ABSTRACT

With the development of the internet there are many content that makes the web more interesting and useful for the users but it also brings along the problem of information overload. Tech Article Aggregator and Summarization System is a web application, which collects latest technology related articles around the world from various sources and present them in one place. The main objective of our system is to reduce the time consumption, as all of the articles that would be explored through more than one website will be placed in a single site. In addition, summarizing this aggregated content will save reader's time. The project is composed of two parts: Article collection and Summarization. For article collection web scraping technique is applied which scrape the content of different news articles and for summarization extractive summarization technique called the Text Rank algorithm is used. Text Rank is an extractive and unsupervised text summarization technique which is based on PageRank. After the implementation of our design we performed some testing to verify if the news articles were extracted in real time and the final result was satisfactory as we obtained real time news articles from different website into one place and also obtained their summarized version.

Keywords: *aggregator, scrapper, text summarization, text enhancement, text rank algorithm*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	: Application Programming Interface
HTTP	: Hyper Text Transfer Protocol
NLP	: Natural Language Processing
NLTK	: Natural Language Toolkit
NumPy	: Numerical Python
RSS	: Rich Site Summary

CHAPTER 1. INTRODUCTION

1.1 Background

In recent years, the world had incredible and huge growth in the rate of articles that is published. People live in a time full of information, data, and news articles. Nowadays tech news has an important part and position within the community. As people read the tech articles daily to keep up with the most recent data and inputs in tech world. Information gain is everything in today's world. With the development of the internet, and lot of websites that provide the same data and information, getting this has become simpler. More content makes the web more interesting for more users, who in turn create more content. In this manner, the end user has gained access to an enormous volume of information, which apart from its clear positive side brings along the problem of information overload. So, users frequently discover it troublesome to decide which of these websites can provide the specified data within the most valuable and effective way.

Despite the pros of the presence of lots of information to the people through the internet, it will get us another problem, which is information overload. There will be too much information that is in front of the user. This problem can be solved through the proposed system. A tech aggregator makes this task easier. It collects the articles from various websites and presents it in a single site. It simplifies reader's search and reading time for tech articles. Using tech aggregation is one of the best ways to stay on top of the tech news. It offers convenience and time saving features.

Another important and major feature of the proposed system is summarization. Summarization is to create a shorter and smaller form of a text by protecting its meaning and the key substance of the initial content. It is the process of generating short, fluent, and most importantly accurate summary of a respectively longer text document. It has many pros like reducing the time of reading to the user and getting only useful and real information. As online textual data grows, automatic text summarization methods have potential to be very helpful because more useful

information can be read in a short time. Content summarization methods can be categorized into extractive summarization and abstractive summarization. Extractive summarization depends on extracting a few parts, such as phrases and sentences, from a piece of text and gathers them together to form a summary. Therefore, identifying the right sentences for summarization is of the most extreme importance in an extractive method. But abstractive summarization utilizes advanced NLP methods to generate a completely new summary. A few parts of this summary may not indeed appear within the original text. In our system, extractive summarization technique will be followed which gives better output and right sentence for summarization. For this, text rank algorithm will be used which is an extractive and unsupervised text summarization technique.

1.2 Problem Statement

As of late, there has been a blast in the measure of text data from an assortment of sources. With the dramatic growth of the internet, people are overwhelmed by the tremendous amount of online information and documents. The accessibility of these news sources generates a large wave of information, which often times can be contradicting and confusing. This volume of text is a priceless source of information and knowledge, which should be effectively aggregated and summarized to be useful. This expanding availability of documents has demanded exhaustive research in automatic text summarization. Automating such a process can help parse through a lot of data and help humans better use their time to make crucial decision. With the sheer volume of media out there, one can be very efficient by reducing the fluff around the most critical information.

1.3 Objectives

The objectives of our project are as follow:

- To collect tech related articles in a single site
- To present summarized version of the articles to the user

1.4 Project Scope

News collector and summarization system mainly focuses on helping users to provide an easy interface to checkout top headlines from tech related websites. Moreover, in today's world market, there are infinite scopes and applications of news collector system. Here are some applications of our system:

- Quick and easy access to articles from different sites.
- Summary of the articles helps users in fast reading and understanding of variety of topics.
- Tech article collector and summarization system can be further expanded to gather news about many other fields.

1.5 Project Features

The news collector and summarization system have following features:

- It provides the latest tech news that is occurring around the world.
- Users can get the story to follow up and dig deeper from different sources.
- It fetches articles in real time.
- It also presents summarized version of articles to the user.

CHAPTER 2. LITERATURE REVIEW

In this section, we will discuss other news/article aggregator websites, which are based on summarization:

In [1], the authors were focusing on gathering news using matrix-based analysis (MNA) with five main steps as follows: the first and second steps are data gathering and extracting the article from the websites and save it in the database. The third step is grouping where they categorize the articles. The last two steps are summarization and visualization that view the important article to the user. Before the grouping step, they added the matrix-based analysis where the matrix has entity as row and the column is the states about the entities. When starting analysis, the user defines what he's looking for where MNA prepare the default values for this purpose. After that, the initialization of the matrix extends a matrix over the two required chosen dimension and look in each cell for the cell documents. The summarization phase is done according to the following steps: topic summary, cell summary and summarizing both by using TF-IDF for each cell in the matrix [2].

According to [3], the authors were aiming to accumulate the content from diverse websites such as articles fond moreover news headlines from blogs and websites. The belief that Rich Site Summary (RSS) gives us summarized and short data, which is preferable for the news aggregator that they are still a successful solution for indexing articles. As reducing the time required for visiting some websites, subscribed users can quickly utilize Rich Site Summary feeds without wasting time going to numerous websites. Creating HTTP requests from the web-server is the primary step in the application and these requests are received from clients. At that point, they utilize Python to download Rich Site Summary feeds and extract articles from it according to the input. After periods, the web-server gives some requests to the subscribed users and in case there are any upgrades, it will be stored and downloaded. Finally, it is possible to say that a decline in newspaper and broadcasted articles is observable. Various factors play here an important role; nevertheless, the method of translating news for the youth is the biggest issue.

Young people do not feel connected to the article any more due to their presentation and language.

Author in [4] was aiming to use Rich Site Summary integrated with HTML by using wrappers (programs) and parser in order to extract the information from a specific source, and then adjust them according to news categories and personalized web views via a web-based interface. They explain how they do the content scanner by using HTML and Rich Site Summary. The first step is wrapping (HTML/Rich Site Summary wrapper) which involves identifying the URL address of the new items from the source with category per the news, and the address is stored in the database as for each category pair and also combined with the corresponding wrapper. The second step of wrapping is getting information from the new items that will be used for getting and indexing the article, for each article they obtain the first sentence and pass it to the corresponding HTML page.

According to [5], the authors were aiming to collect the news from multiple sites, newspapers, magazines, and television and merge them all in one summarized website. It progresses the goodness of results because the contents and data in it are brief and summarized. So, their work based on the Rich Site Summary fetcher for recovering Rich Site Summary reports from specific websites at a certain time. They also use web Crawling (Scraping) besides Rich Site Summary to get more accurate results. Web scraping may be a method utilized to collect huge amounts of information from websites.

From all the above-mentioned researches on the news/article aggregator, the quality of the aggregator system is still an open area to be introduced.

Web scraping; however, can also be used to simply extract public data. Applications are many: from extracting the content of a website to use it for Data Mining, Data Indexing, to extracting offers of competitors in order to make a comparison for online analysis of E-commerce websites. The process of web scraping includes three steps; first, fetching or downloading of a page and it is similar to what the browser does when you view the page using one of the scraping libraries. Second, web crawling process or fetching pages for later

processing [5]. Finally, the content of the selected page will be parsed, searched, or reformatted; all data will be copied into a spreadsheet or JSON files, and so on.

Data scraping is a term used to describe the extraction of data from an electronic file using a computer program. Web scraping describes the use of a program to extract data from HTML files on the internet. Typically, this data is in the form of patterned data, particularly lists or tables. Programs that interact with web pages and extract data use sets of commands known as application programming interfaces (APIs). These APIs can be 'taught' to extract patterned data from single web pages or from all similar pages across an entire web site. Alternatively, automated interactions with websites can be built into APIs, such that links within a page can be 'clicked' and data extracted from subsequent pages. This is particularly useful for extracting data from multiple pages of search results. Furthermore, this interactivity allows users to automate the use of websites' search facilities, extracting data from multiple pages of search results and only requiring users to input search terms rather than having to navigate to and search each website first. One major current use for web scraping is for businesses to track pricing activities of their competitors: pricing can be established across an entire site in relatively short time scales and with minimal manual effort. Various other commercial drivers have caused a large number and variety of web scraping programs to have been developed in recent years.

Some of these programs are free, whilst others are purely commercial and charge a one off or regular subscription fee. These web scraping tools are equally as useful in the research realm. Specifically, they can provide valuable opportunities in the search for grey literature, by: i) making searches of multiple websites more resource-efficient; ii) drastically increasing transparency in search activities; and iii) allowing researches to share trained APIs for specific websites, further increasing resource-efficiency. A further benefit of web scraping APIs relates to their use with traditional academic databases, such as Web of Science. Whilst citations, including abstracts, are readily extractable from most academic databases, many databases hold more useful information that is not readily exportable, for example corresponding author information. Web scraping tools

can be used to extract this information from search results, allowing researchers to assemble contact lists that may prove particularly useful in requests for additional data, calls for submission of evidence, or invitations to take part in surveys.[6]

Text summarization automatically produces a summary containing important sentences and includes all relevant important information from the original document. One of the main approaches, when viewed from the summary results, is extractive and abstractive. An extractive summary is heading towards maturity and now research has shifted towards abstractive summation and real-time summarization. Although there have been so many achievements in the acquisition of datasets, methods, and techniques published, there are not many papers that can provide a broad picture of the current state of research in this field. This paper provides a broad and systematic review of research in the field of text summarization published from 2008 to 2019. There are 85 journal and conference publications which are the results of the extraction of selected studies for identification and analysis to describe research topics/trends, datasets, preprocessing, features, techniques, methods, evaluations, and problems in this field of research. The results of the analysis provide an in-depth explanation of the topics/trends that are the focus of their research in the field of text summarization; provide references to public datasets, preprocessing and features that have been used; describes the techniques and methods that are often used by researchers as a comparison and means for developing methods. At the end of this paper, several recommendations for opportunities and challenges related to text summarization research are mentioned [7].

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 is very difficult for human beings to manually extract the summary of large documents of text. There are plenty of text materials available on the internet. Therefore, there is a problem of searching for relevant documents from the number of documents available, and absorbing relevant information from it. In order to solve the above two problems, the automatic text summarization is very much necessary. Text summarization is the process of

identifying the most important meaningful information in a document or set of related documents and compressing them into a shorter version preserving its overall meanings. Before going to the Text summarization, first we, have to know that what a summary is. A summary is a text that is produced from one or more texts, that conveys important information in the original text, and it is of a shorter form. The goal of automatic text summarization is presenting the source text into a shorter version with semantics. The most important advantage of using a summary is; it reduces the reading time. Text Summarization methods can be classified into extractive and abstractive summarization. An extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form. An abstractive summarization is an understanding of the main concepts in a document and then expresses those concepts in clear natural language. There are two different groups of text summarization: indicative and informative. Inductive summarization only represents the main idea of the text to the user. The typical length of this type of summarization is 5 to 10 percent of the main text. On the other hand, the informative summarization systems give concise information of the main text. The length of informative summary is 20 to 30 percent of the main text [8].

CHAPTER 3. SYSTEM ANALYSIS

3.1 Requirement Analysis

3.1.1 Functional Requirement

- Collect Article: The system should be able to fetch technology related article from different sites and present them in one location.
- Generate Summary: The system should generate the summary of the collected articles.

3.1.2 Non Functional Requirement

- User friendly: The system should be user friendly and easy to use.
- Reliability: The system should be reliable and should not crash
- Performance: It should not take excess time to open. The user experience should be smooth and response time should be quick.
- Compatibility: It should be compatible and should run on all devices.
- Scalability: It should be to handle various ranges of users.

3.1.3 Tools and Techniques

- Frontend: HTML, CSS, Bootstrap
- Backend: Python, Django framework
- Database: Dbsqlite3
- Library: NLTK(3.6.7), NumPy(1.22.1)

CHAPTER 4. SYSTEM DESIGN

4.1 System Overview

System overview provides the overall concept of the system. It describes what the system is and what it does. The system overview diagram of our system is:

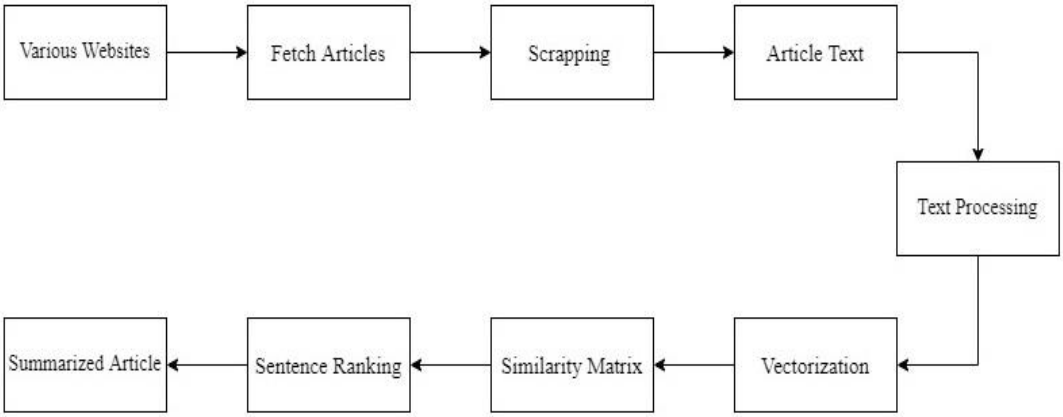


Figure 4. 1 System Overview of Tech Article Aggregator and Summarization

The system collects tech articles from various websites. The articles elements like title, image and url are fetched. The article contents are scrapped to obtain contents in text form. Then those text are processed to remove stop words and whitespaces for tokenization process. The tokenized text is passed through vectorization process. After that similarity matrix is generated using cosine distance formula. Now Text rank algorithm is applied for sentence ranking. Finally, top five sentences are selected for obtaining summarized article.

4.2 Use Case Diagram

A use case diagram is a way to summarize details of a system and the users within that system. It is generally shown as a graphic depiction of interactions among different elements in a system. The use case diagram of our system is:

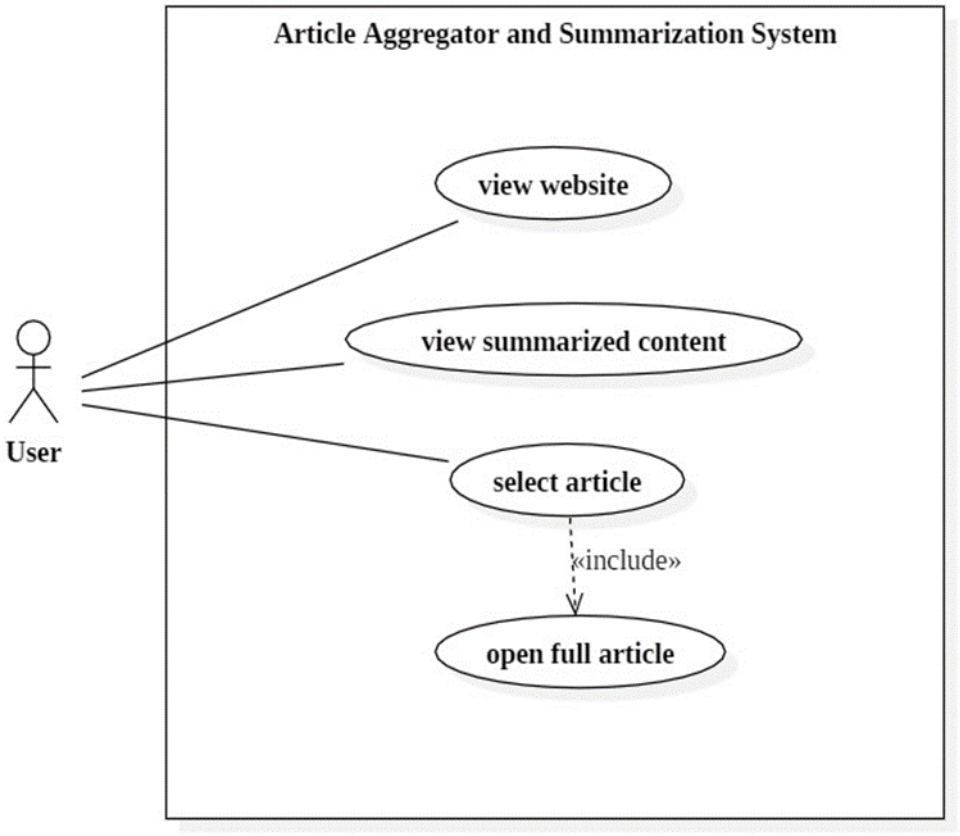


Figure 4.2 Use Case Diagram of Tech Article Aggregator and Summarization

The user views the website that shows top articles aggregated from six tech websites along with the summarized content. The user can also select the article through provided link to read full news.

4.3 Sequence Diagram

Sequence diagram represents the details of a UML use case and models the logic of a sophisticated procedure, function, or operation. It also shows how objects and components interact with each other to complete a process.

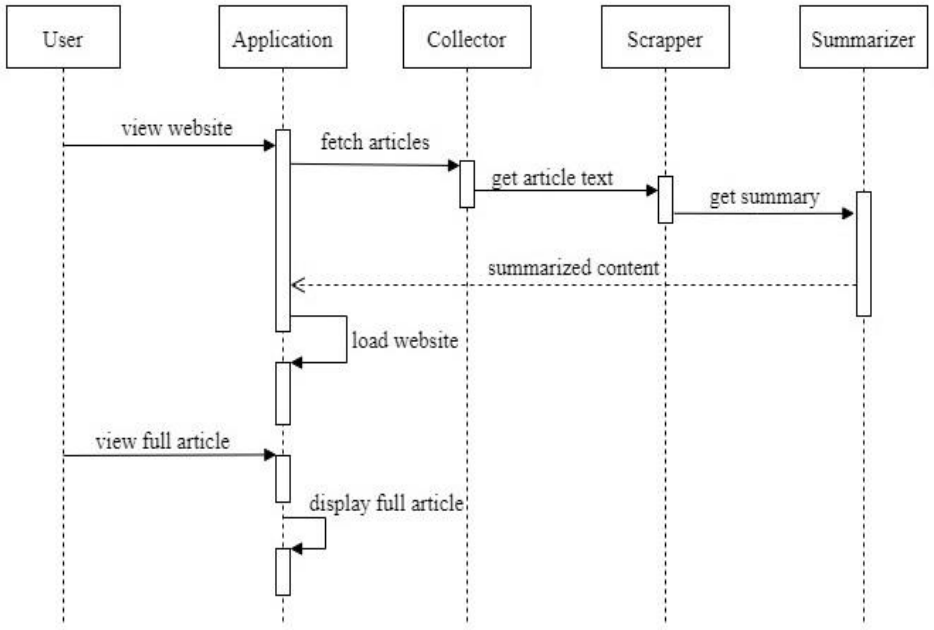


Figure 4.3 Sequence Diagram of Tech Article Aggregator and Summarization

4.4 DFD

DFD-level 0

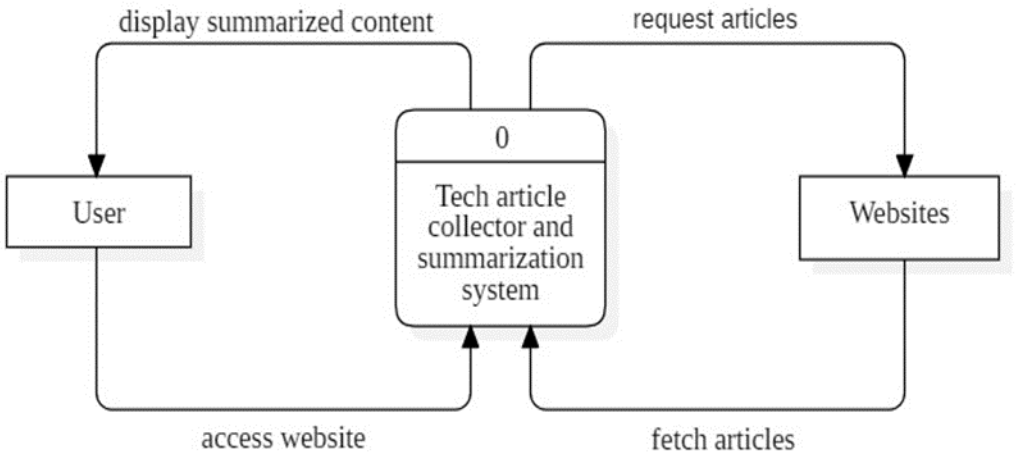


Figure 4.4 DFD 0 of Tech Article Aggregator and Summarization

DFD-level 1

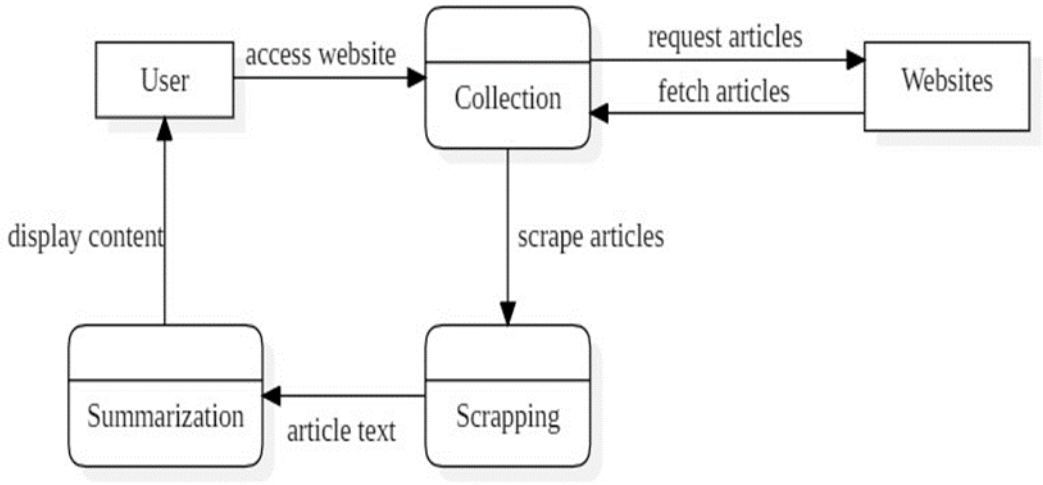


Figure 4.5 DFD 1 of Tech Article Aggregator and Summarization

CHAPTER 5. METHODOLOGY

The development of the project is divided into the following key stages:

5.1 Background Research and Investigation

This stage, as expected to be the one of the most time-consuming phases, involved compiling, analyzing and understanding materials about different subjects like: Natural Language Processing, Vectorization, Tokenization, Similarity matrix and Sentence scoring. Research included all forms of media, tools, programming libraries, application programming interfaces, external samples and efforts, scientific papers as well as textual content.

5.2 Project Methods

Any research topic, software or otherwise, requires certain methodologies to be used and adhered to, in order to succeed. As the project encompasses different technical fields and requires specific handling for each subsection, the project is divided into two different sections:

- Article collection and summarization system development; and,
- Deployment and successful user interaction.

5.2.1 Article Collection

The first step is to collect tech related articles from various tech websites and show them in our website. For collecting the articles, we used Newsapi for collecting the latest top headlines from various websites.

News API is a simple HTTP REST API for searching and retrieving live articles from all over the web. In order to use this api, a special type of key is required called api_key. This is a unique identifier used to authenticate a user, developer or calling program to an API. By using this api, we are able to fetch the required data. This api provides feature of pulling various article elements to our site like author, title, description, url, image, published date etc. We only fetched article

elements like title, url and image of latest top headlines to our website using newsapi. Although it says to retrieve fresh live articles, there is certain amount of time delay. The articles recently published in original website do not immediately shows in our website.

This completes article-fetching process of our project. We have manually fetched the articles from six different tech websites like techcrunch, techradar, theverge, the next-web, engadget and wired. After this comes the aggregation part. For this, we created the layout and presented the gathered articles in the web application.

Up to this point, the user can view the website containing aggregated latest tech news articles along with the title, image and url. The user can also visit the original site of the article through the provided link.

5.2.2 Scrapping Article

Scraping simply means to capture information from online sources. Generally, web scraping involves accessing numerous websites and collecting data from them. However, the aim of this project is to scrape news articles from different websites using Python. After collecting the required articles from different sites, we then extracted article content in text form. The article content is required in text form for summarization purpose.

For scrapping, an api named extract news api was used. It pulls structured data from online news articles. The working of this api is similar to newsapi. A key is provided for identifying the user. Requests python module is used for extracting. Requests is an elegant and simple HTTP library for Python. Requests library is one of the integral part of Python for making HTTP requests to a specified URL. When one makes a request to a URI, it returns a response. Python requests provides inbuilt functionalities for managing both the request and response. The HTTP request returns a response object with all the response data (content, encoding, status, etc). Python requests module has several built-in methods to make HTTP requests to specified URI using GET, POST, PUT, PATCH or HEAD requests. The HTTP request is meant either to retrieve data from a

specified URI or to push data to a server. It works as a request-response protocol between a client and a server. Since we only need article content from the websites, GET requests is used. GET method is used to retrieve information from the given server using a given URI. The GET method sends the encoded user information appended to the page request. Therefore, the url of collected articles are provided to this scrapper api and article content is extracted in text form. Those texts are preceded to summarization process afterwards.

5.2.3 Summarization

Summarization is a brief and accurate representation of input text such that the output covers the most important concepts of the source in a condensed manner. Text summarization in NLP is the process of summarizing the information in large texts for quicker consumption. Text summarization methods can be grouped into two main categories: Extractive and Abstractive methods. Extractive summarization aims at identifying the salient information, extract them and group together to form a concise summary whereas abstractive summary generation rewrites the entire document by building internal semantic representation, and then a summary is created.

In our project, we used extractive method for summary generation. Extractive summarization techniques involve: the construction of an intermediate representation of the input text (text to be summarized), the creation of a scoring of the sentences and the selection of the top K most important sentences. The main objective is to identify the significant sentences of the text and add them to the summary. The summary obtained contains exact sentences from the original text. For extractive method of summary generation, we implemented Text Rank Algorithm. TextRank is an extractive and unsupervised text summarization technique.It is based on the PageRank Algorithm. It includes following steps:

- Extract all the sentences from the text document, either by splitting at whitespaces or full stops, or any other way.

- A graph is created out of the extracted sentences. The nodes represent the sentences, while the weight on the edges between two nodes is found by using a Similarity function, like Cosine Similarity.
- Find importance (scores) of each node by iterating the algorithm until convergence i.e. until consistent scores are obtained.
- The sentences are sorted in a descending order based upon their scores. The first *k* sentences are chosen to be a part of the text summary.

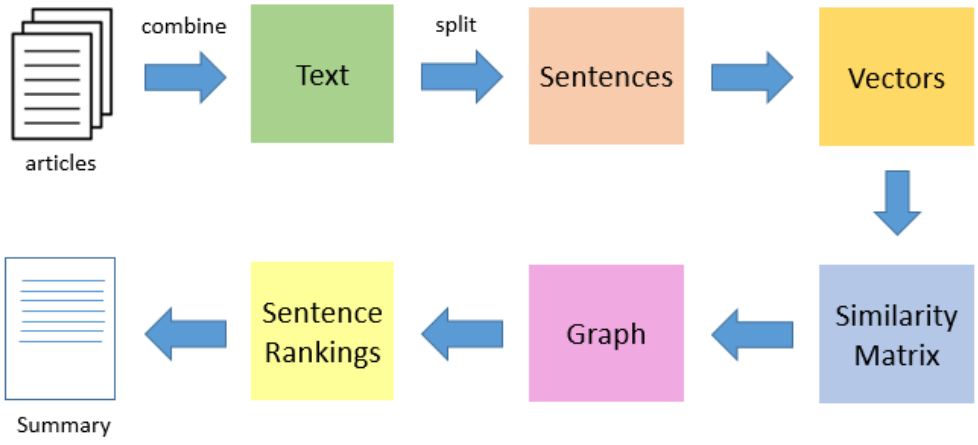


Figure 5.1 Steps in TextRank Algorithm

The steps can be illustrated below:

Obtaining articles: The articles are obtained in text form from scrapping process.

Tokenization: Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens.Sentence tokenization is the process of splitting text into individual sentences i.e. tokenizing the original text into sentences. These tokens could be paragraphs, sentences, or individual words. NLTK provides a number of tokenizers in the tokenize module. The demo below shows how they work using the PunktSentence Tokenizer. Then each sentence is tokenized into words using 4 different word tokenizers:

- Treebank WordTokenizer
- WordPunctTokenizer
- PunctWordTokenizer
- WhitespaceTokenizer

Example

Tokenize text: “Hello everyone, it’s an example of tokenization.”

Treebank WordTokenizer

Hello everyone ,it ‘s an example of tokenization .

WordPunctTokenizer

Hello everyone , it ‘ s an example of tokenization .

PunctWordTokenizer

Hello everyone ,it ‘s an example of tokenization.

WhitespaceTokenizer

Hello everyone, it ‘s an example of tokenization.

NLTK contains a module called *tokenize()* which further classifies into two sub-categories:

Word tokenize: We use the word_tokenize() method to split a sentence into tokens or words. word_tokenize() function is a wrapper function that calls tokenize() on an instance of the TreebankWordTokenizer class.

Sentence tokenize: We use the sent_tokenize() method to split a document or paragraph into sentences.

Vectorization: The next step is vectorization. In this step, each word is represented by a vector based on the co-occurrence of a word with the others in a single sentence.

Consider a Corpus C of D documents {d1,d2…..dD} and N unique tokens extracted out of the corpus C.

D1: He is a lazy boy. She is also lazy.

D2: Neeraj is a lazy person.

The dictionary created may be a list of unique tokens(words)

= [‘He’,’She’,’lazy’,’boy’,’Neeraj’,’person’]

Here, D=2, N=6

The count matrix M of size 2 X 6 will be represented as –

	He	She	lazy	boy	Neeraj	person
D1	1	1	2	1	0	0
D2	0	0	1	0	1	1

The two vectors are:

D1 = [1, 1, 2, 1, 0, 0]

D2 = [0, 0, 1, 0, 1, 1]

Similarity Matrix: Then we obtain a similarity matrix for all sentences using cosine similarity. It measures the cosine of the angle between two vectors (D1, D2) projected in an N-dimensional vector space. The similarity here refers to common content in sentences. “Smaller the angle, the higher the similarity” — Cosine Similarity.

$$\boldsymbol{cos\theta} = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|\,\|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.1$$

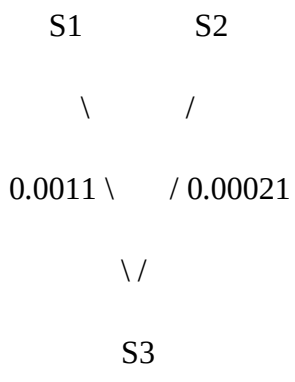
Where, $\vec{a}.\vec{b} = \sum_1^n a_i\, b_i = a_1 b_1 + a_2 b_2 \ldots\ldots\ldots + a_n b_n$ is the dot product of two vector.

Consider three sentences and after applying cosine similarity between them, their respective value be:

Sentence	S1	S2	S3
S1	1	0	0.0011
S2	0	1	0.00021
S3	0.0011	0.00021	1

Graph: After obtaining the similarity matrix in the previous step, we convert it into a Graph where the edges determined by a similarity relation between them and sentences become node. Those edges are used to obtain the vertices weight.

Forming a graph with sentences as nodes and edges represented by the similarity metric for above example:



Now apply PageRank to the above graph. The importance of a sentence is based on the number of edges that represented as a score for each vertex using PageRank algorithm. According to the PageRank algorithm, we can treat the sentences

(nodes) as webpages and the edges as links to the webpages. Applying PageRank to the above graph concludes that node S3 is the most important/highly ranked sentence since it contains most links to other nodes.

PageRank: PageRank (PR) is an algorithm used by Google Search to rank websites in their search engine results. Lets see an example on how it works:

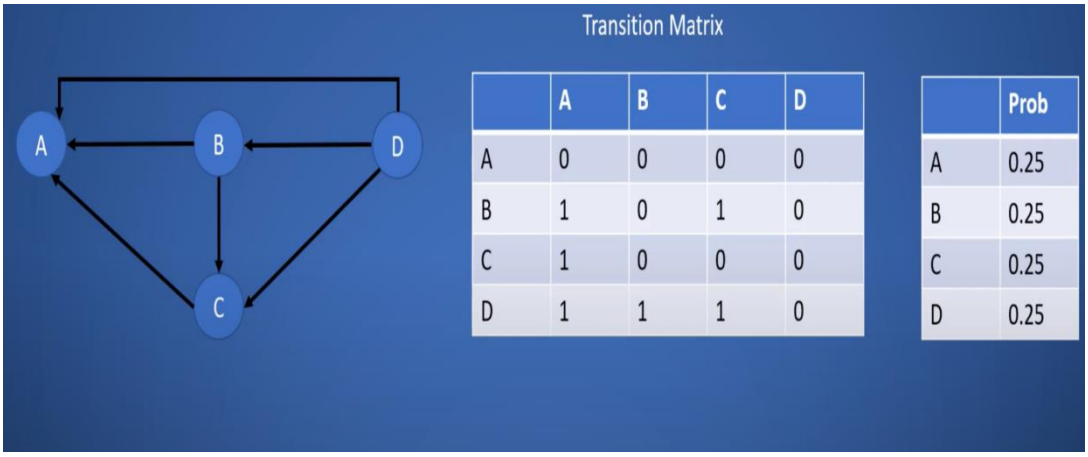


Figure 5.2 PageRank-Transition matrix

Consider four pages A, B, C and D. According to graph, find transition matrix as shown above. At start, all pages have equal probability of 0.25. The page rank theory holds that an imaginary surfer who is randomly clicking on link will eventually stop clicking. The probability that the person will continue is a damping factor d. from various tests damping factor is generally assumed to be around 0.85. PageRank takes damping factor, epsilon value and transition matrix as input to calculate new probability. It is an iterative algorithm. The calculation is repeated lots of time until the numbers stops changing much.

$$PR(A) = (1-d) + d\,(PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn)) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 5.2$$

Where;

PR(A) is the Page Rank of page A,

PR(Ti) is the Page Rank of pages Ti which link to page A,

C(Ti) is the number of outbound links on page Ti, and

d is a damping factor which can be set between 0 and 1

After completion of page rank algorithm, the new probability looks like this:

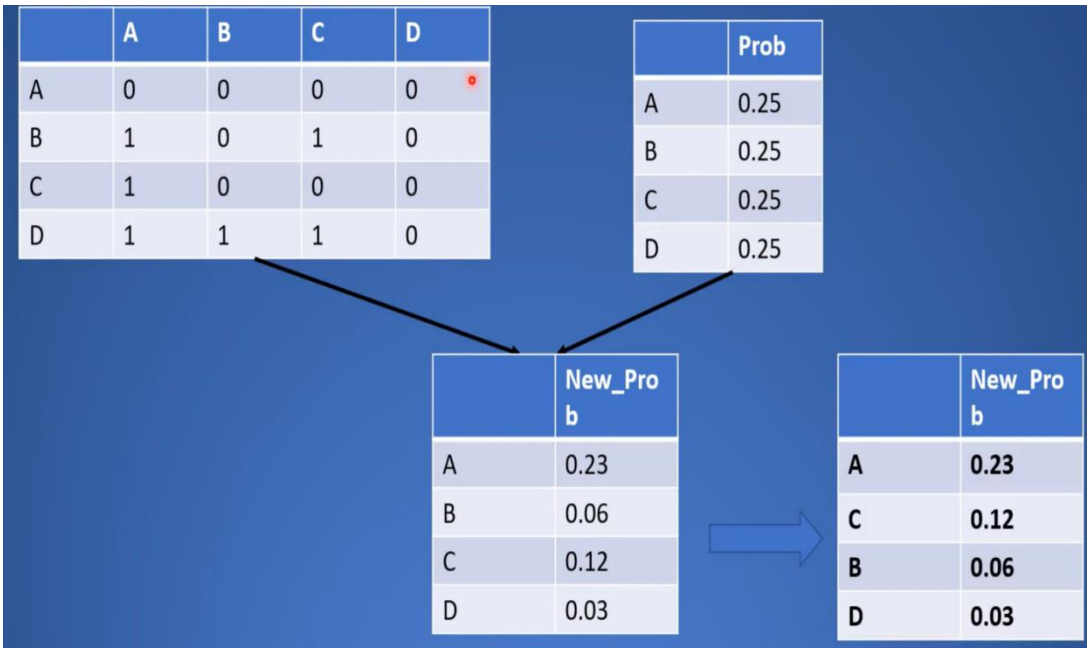


Figure 5.3 PageRank- calculate new probability

And finally the web pages are ranked according to their new probability. The page with highest probability is ranked first and so on.

PageRank for TextRank: TextRank also uses the same logic as in PageRank but with some subtle changes:

- Text sentences are used in place of webpages.
- The similarity matrix for index [A, B] is filled with similarity values between sentences A & B rather than 1/total_links from Page B to A which can be calculated using cosine distance.

So, PageRank is now used for ranking sentences. In place of transition matrix, here comes similarity matrix which is obtained from cosine similarity as explained

in previous steps. Now apply PageRank algorithm to obtain new probability. PageRank is applied after obtaining similarity matrix and graph.

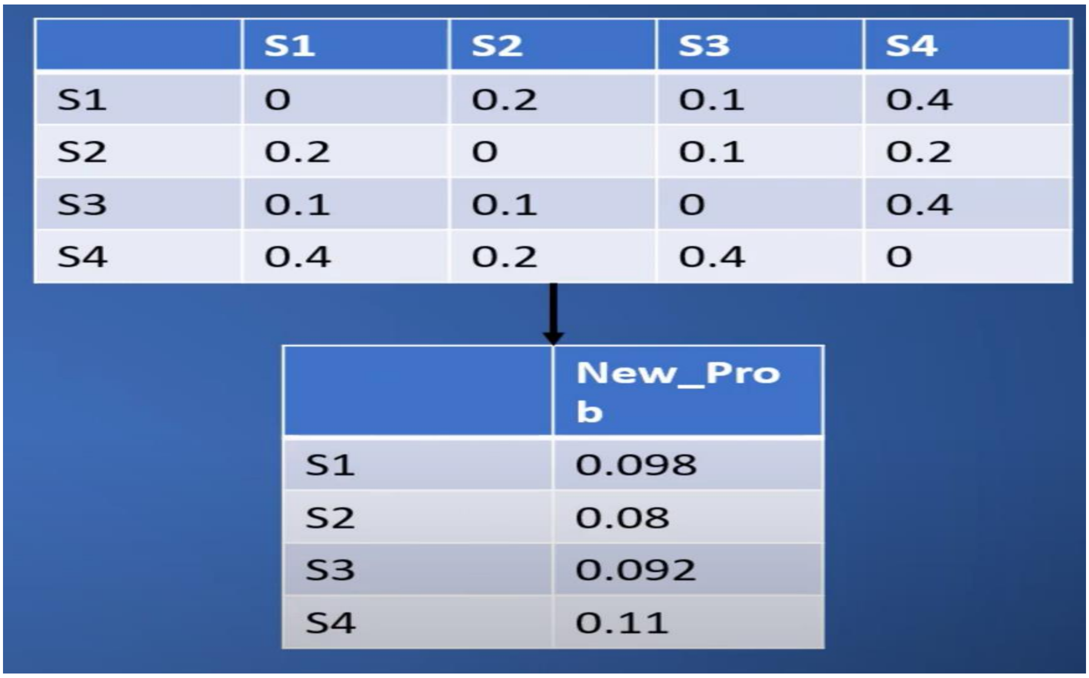
Let the directed graph, G = (V, E)

where V represents set of vertices and E represents set of edges.

The vertex score, Vi is defined as follows:

$$\boldsymbol{S}(\boldsymbol{V_i}) = (\mathbf{1} - \boldsymbol{d}) + \boldsymbol{d} * \sum_{j\epsilon|(V_i)|} \frac{\mathbf{1}}{|V_j|} \boldsymbol{S}(\boldsymbol{V_j}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. \; 5.3$$

Where, d is a factor that its value is between 0 and 1 and usually the value is 0.85, which represents the probability of going to another random vertex from a given vertex in the graph.



	S1	S2	S3	S4
S1	0	0.2	0.1	0.4
S2	0.2	0	0.1	0.2
S3	0.1	0.1	0	0.4
S4	0.4	0.2	0.4	0

	New_Prob
S1	0.098
S2	0.08
S3	0.092
S4	0.11

Figure 5.4 PageRank for sentence ranking

Sentence Ranking: The final step is sorting sentences in descending order according to their new probability. The top 5 sentences are extracted for summary formation.

Let's take an example:

Text = ‘He is a nice guy. He has a lot of friends. Raj is his best friend’.

Mark the sentences as A, B & C respectively for now where;

A=’He is a nice guy’

B=’He has a lot of friends’

C=’Raj is his best friend’

Assume after tokenization and vectorization, the similarity matrix from cosine similarity be:

Sentence	A	B	C
A	0	0.53	0.2
B	0.53	0	0.9
C	0.2	0.9	0

Initialize TextRank(A), TextRank(B), TextRank(C) as 1. Take d=0.85

Iteration 1:

TextRank (He has a lot of friends)

=(1–0.85) + 0.85*(TextRank(A)*M[B,A]+TextRank(C)*M[B,C])

=0.15+0.85*(1*0.53+1*0.90)

= 1.3655

TextRank (He is a nice guy)

=0.15+ 0.85* (TextRank(B)*M[A,B]+TextRank(C)*M[A,C])

=0.15+0.85*(1.3655*0.53+1*0.2)

=0.932

TextRank (Raj is his best friend)

=0.15+0.85*(TextRank(A)*M[C,A]+TextRank(B)*M[C,B])

=0.15+0.85*(0.932*0.2+1.3655*0.9)

=1.34

Now, repeat the update cycle for every sentence for 'n' iterations. Arranging these sentences based on their TextRanks will give us the most important sentences, which can be used as a summary. The top 'K' sentences are extracted to present summary.

5.3 UI and Deployment

5.3.1 Django

Django is a Python-based free and open-source web framework that follows the model–template–views (MTV) architectural pattern. Django is a high-level Python Web framework that encourages rapid development and clean pragmatic design. A Web framework is a set of components that provide a standard way to develop websites fast and easily. Django's primary goal is to ease the creation of complex database-driven websites.

Libraries:

1. **NumPy:** NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

2. **NLTK:** Natural Language Toolkit (NLTK) is a widely used, open-source Python library for NLP (NLTK Project, 2018). Several algorithms are available for text tokenization, stemming, stop word removal, classification, clustering, PoS tagging, parsing, and semantic reasoning. It also provides

wrappers for other NLP libraries. A notable feature of NLTK is that it provides access to over 50 corpora and lexical resources such as the Word Net.

CHAPTER 6. EVALUATION AND TESTING

Testing is the process of evaluating a system or its module(s) with the intent to find whether it fulfills the identified requirements or not. Moreover, testing is executing a system in order to recognize any gaps, errors, or missing necessities in contrary to actual requirements. Before actually implementing the new system into actions, a trial run of the system is done eliminating all the bugs, if any. After organizing the entire programs of the system, a test plan should be developed and run on a given set of test data. The output of the test run should meet the expected results. This project includes several stages of testing, some of them are mentioned below:

6.1 Unit Testing

During the development phase each module is tested independently to view whether the desired output is achieved or not. By unit testing the proper functioning of individual part of the system was verified.

Table 6.1 Article Collection

SN.	TEST CASE	EXPECTED OUTCOME	ACTUAL OUTCOME	REMARKS
1.	Validating scrapping in real time	Articles are collected in real time	Same as expected	Validated
2.	Validating news source	Articles are collected from provided websites	Same as expected	Validated

Table 6.2 Summarization

SN.	TEST CASE	EXPECTED OUTCOME	ACTUAL OUTCOME	REMARKS
1.	Matching summary to article	Respective summary of articles are obtained	Same as expected	Validated
2.	Correctness	Free from grammatical errors	Same as expected	Validated

6.2 Integration Testing

After unit testing is accomplished by proper functioning, each individual module was integrated and formed a compact system, then overall system was tested to identify whether there was any fault in integration or not.

Table 6.3 Integration Testing

SN.	TEST CASE	EXPECTED OUTCOME	ACTUAL OUTCOME	REMARKS
1.	Validating articles	Articles are collected from provided websites in real time	Same as expected	Validated
2.	Validating summary	Extractive summary generation	Same as expected	Validated

CHAPTER 7. RESULT ANALYSIS AND DISCUSSION

7.1 Result Analysis

After the completion of the project, we analyzed the result of our system to check if our system performs the way we expected or not. Firstly, the UI of the system was interactive for better user experience. Then we analyzed the article collection part of our project and made sure that the articles were being collected in real time.After that, we analyzed the summarization part of our project. For this we checked the accuracy of generated summary as well as the number of sentences in the summary.

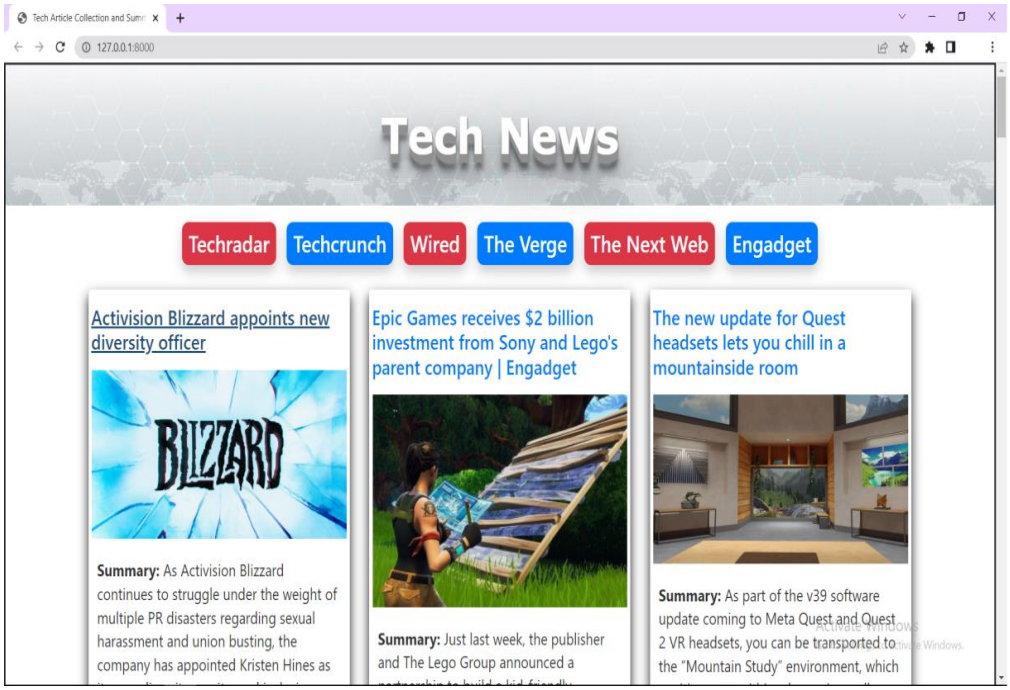


Figure 7.1 System Homepage

The above diagram depicts the homepage of the application. It presents tech articles from six different websites like techradar, techcrunch, wired, the verge, the next web and endgaget. The title, image and summarized content is seen in main page. If the user wants to read full article, he can read by clicking on provided link. Also the user can view separate articles and their description of each tech websites by clicking on the button.

7.2 Discussion

Tech article aggregator and summarization system is a web application presenting summarized latest tech news to the user collected from six different tech websites. This application was built using Django framework. Likewise, bootstrap was used for user interface layout. Newsapi was used to fetch top fresh news from those websites in real time. Then the url of each article was passed to the scrapper to extract the article content. This was done because article content was required in structured text form for summarization. So for scrapping process, extract news api was used.

Finally, we implemented text rank algorithm that is an extractive summarization technique. At first, the extracted texts from scrapper were tokenized into words and sentences. The stop words and whitespaces were removed in this process. Then those sentences were vectored to get numeric value for each tokens. After that cosine similarity was applied to get similarity value between sentences. Similarity matrix was formed and finally page rank was implemented for ranking sentences. The top five sentences were extracted to form summary.

CHAPTER 8. CONCLUSION AND FUTURE ENHANCEMENT

8.1 Conclusion

Tech Article Aggregator and Summarization System is simply a web application that collects tech articles around the world from various sources all in one place. It plays a very important role in reducing time consumption, as all of the articles that would be explored through more than one website will be placed in a single site. In addition, summarizing this aggregated content will save reader's time.With the completion of this project, the main aim of the project is achieved.

With the end of project, the members involved in project gained lots of experience on teamwork and they discovered various predicted and unpredicted problem and also implemented various idea to solve them various resources like video tutorial, text tutorial, internet and learning material were used to make project complete.

8.2 Future Enhancement

The system we designed works as intended but there is still room for improvement in the system. Some of the possible future enhancements that can be done in the system are:

- Option to download the articles and read them offline.
- Notification feature to alert user about latest news.

REFERENCES

[1] D.-C. C. C. O. a. S. T.-M. C. Grozea, ", “Atlas: News aggregation service,”," 16th RoEduNet Conference: Networking in Education and Research (RoEduNet), 2017.

[2] A. M. V. M. a. C. S. G. Paliouras, "A Personalized News Aggregator on the Web," 1970.

[3] D.-S. J. a. N. N. Esfahani, News aggregators and competition among newspapers in the internet (preliminary and incomplete), 2012.

[4] N. M. a. B. G. F. Hamborg, "Matrix-based news aggregation: exploring different news perspectives,”," IEEE Press, 2017.

[5] R. D. a. S. N. K. Sundaramoorthy, "“Newsone—an aggregation system for news using web scraping method,”," International Conference on Technical Advancements in Computers and Communications (ICTACC), 2017.

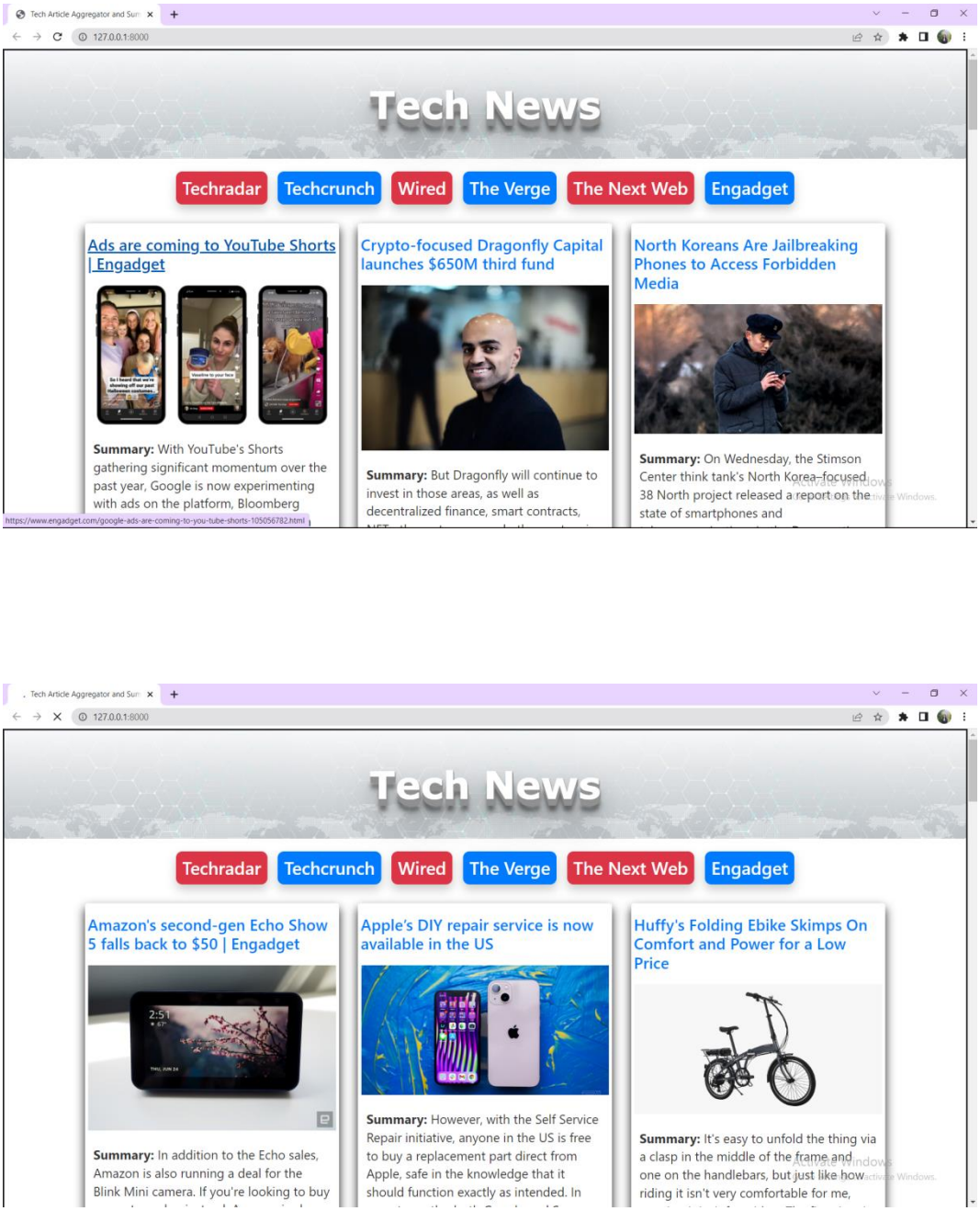
[6] N. Haddaway, The Use of Web-scraping Software in Searching for Grey Literature, Grey Journal, 2015.

[7] S. &. T.-C. M. &. R. Babar, Text Summarization:An Overview, 2013.

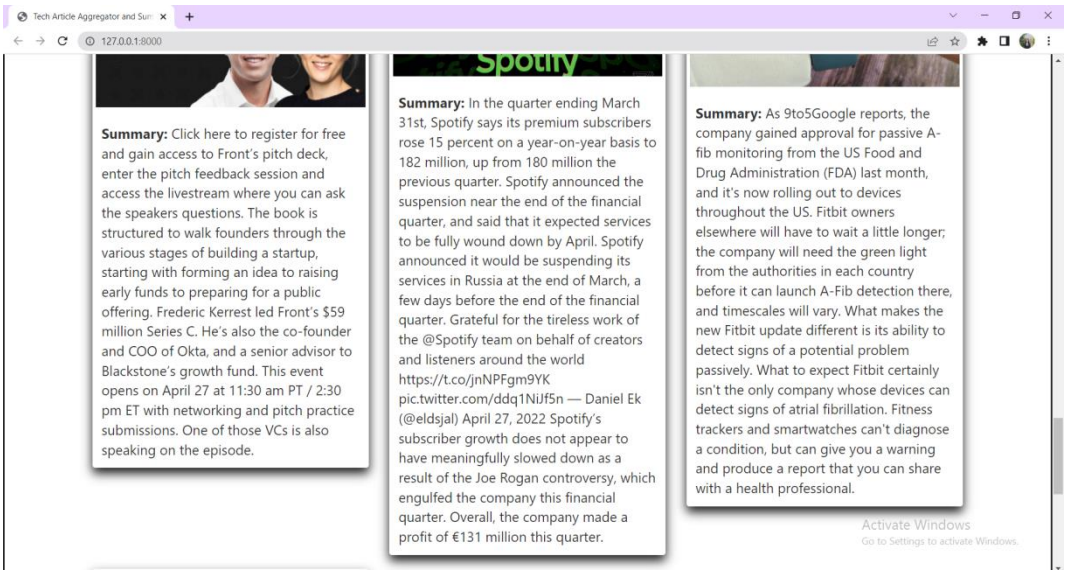
[8] V. S. L. Krotov, "Legality and ethics of web scraping," 2018.

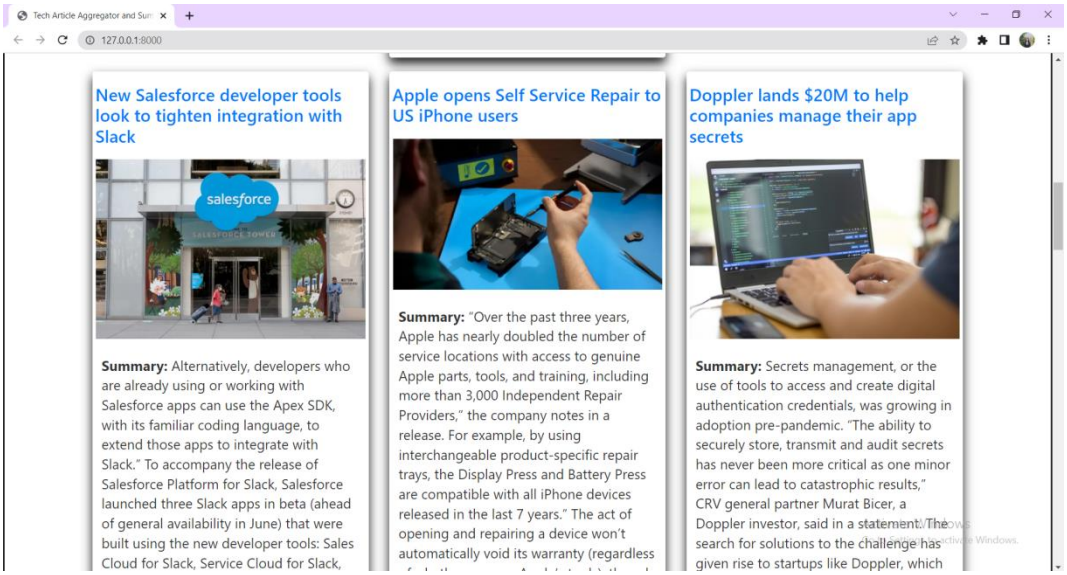
APPENDICES

SCREENSHOTS:

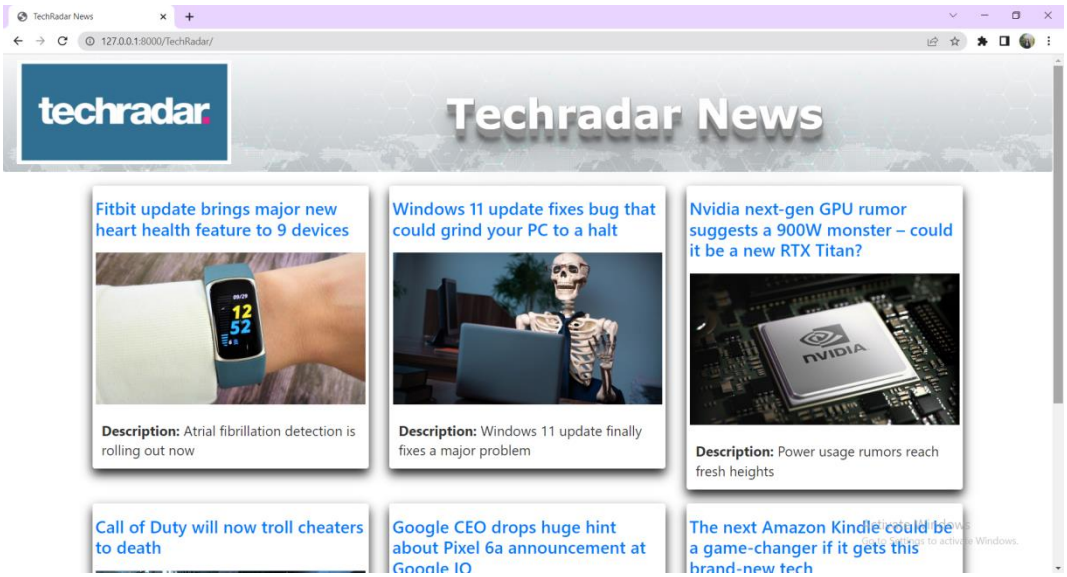


Home Page of system

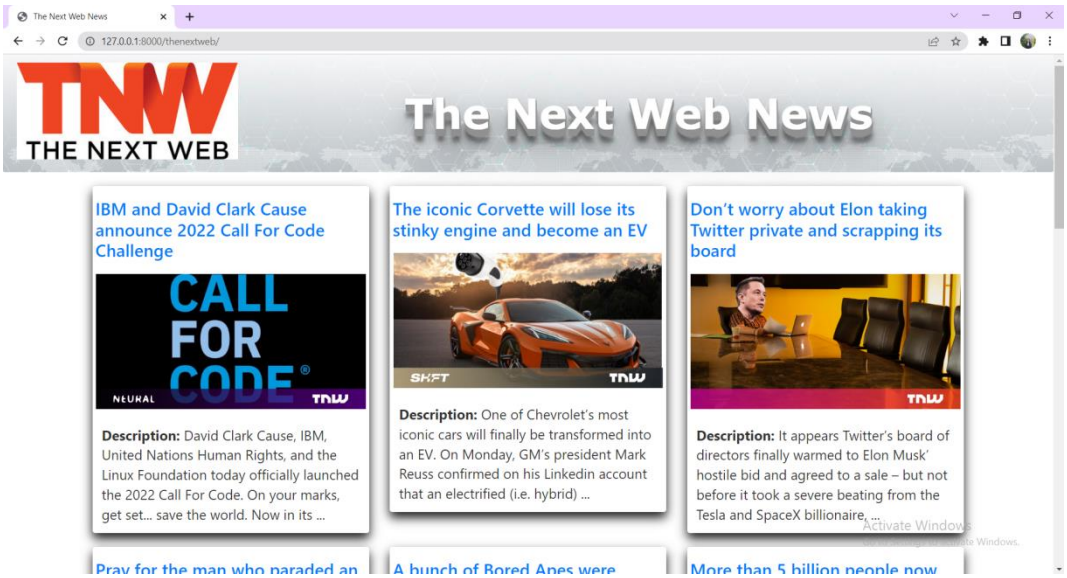




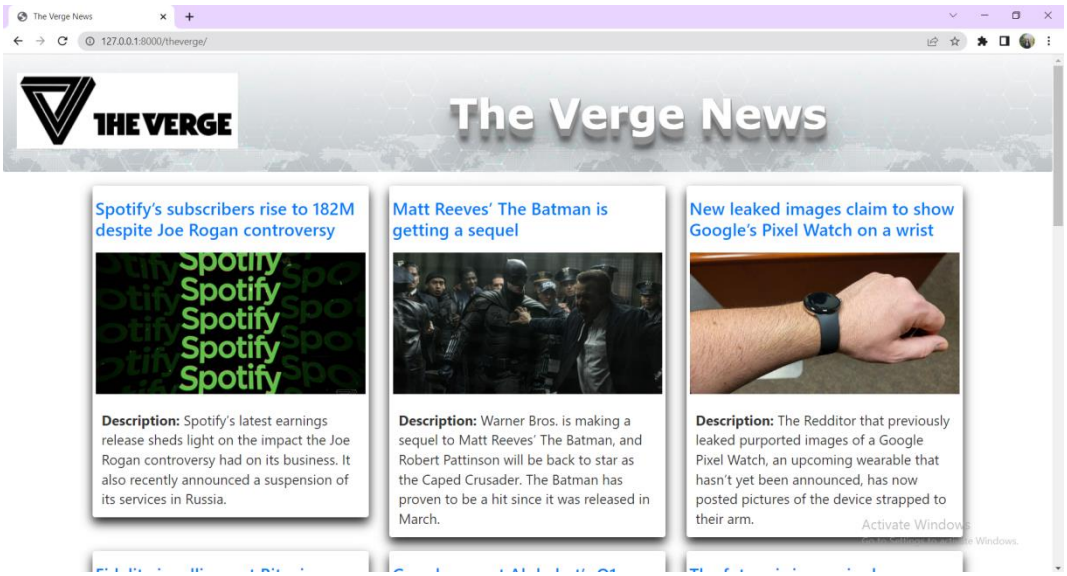
Summary of articles



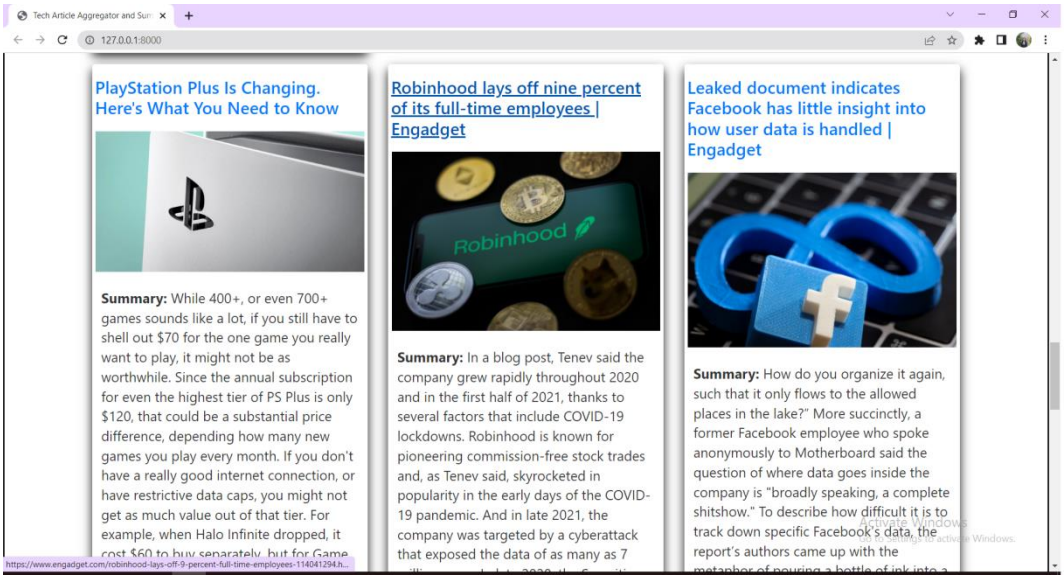
News section of Techradar



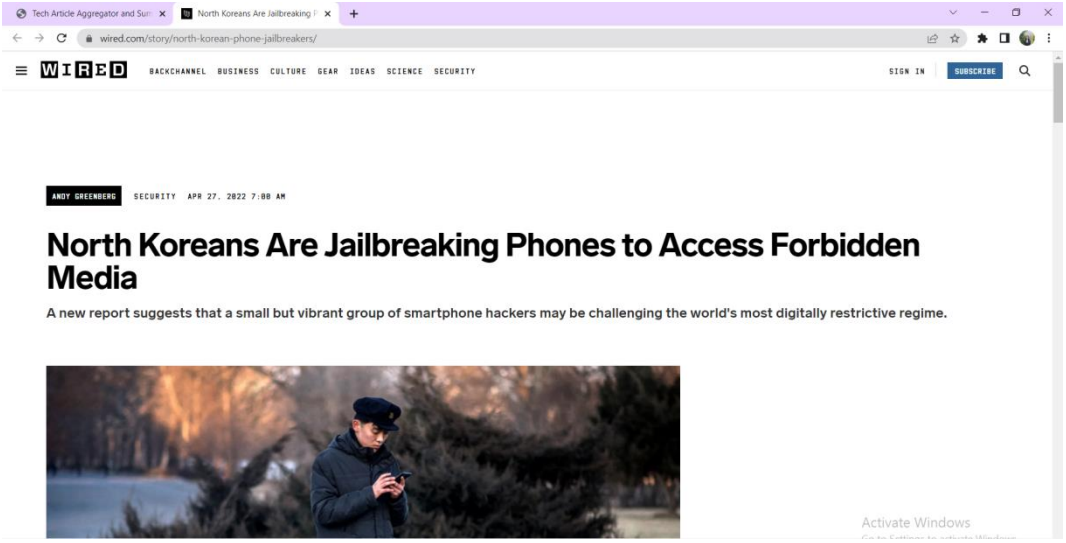
News section of The Next Web



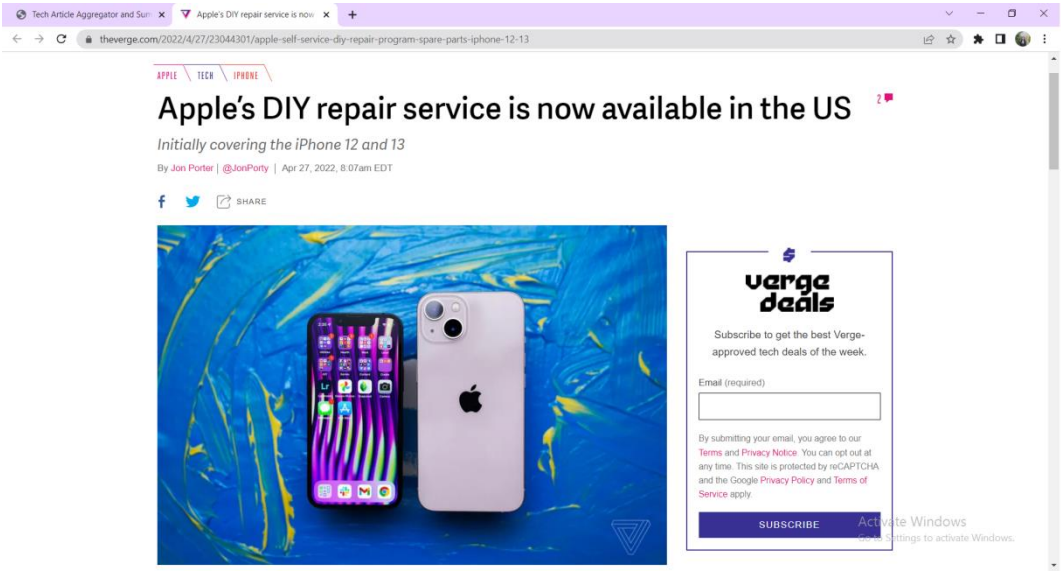
News Section of The Verge News



Summary of articles



Full Article



Full article