REAL TIME AUTOMATED SPARE PARTS INFORMATION RETRIEVAL

A PROJECT REPORT

Submitted by

MOULIRAJ A K

(2023176001)

A report for the phase-I of the project
submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment
for the award of the degree
of

MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

SPECIALIZATION IN AI & DS

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600 025

DECEMBER 2024

ANNA UNIVERSITY

CHENNAI - 600 025

BONA FIDE CERTIFICATE

Certified that this project report titled ”Real Time Automated Spare Parts Information Retrieval ” is the bonafide work of Mouliraj A K (2023176001) who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE:	**DR. S. BAMA**
DATE:	**ASSOCIATE PROFESSOR**
	PROJECT GUIDE
	DEPARTMENT OF IST, CEG
	ANNA UNIVERSITY
	CHENNAI 600025

COUNTERSIGNED

DR. S. SWAMYNATHAN
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

In the digital marketplace, vast amounts of product information are spread across numerous e-commerce platforms, making manual data collection and analysis both challenging and time-consuming. This project presents an automated pipeline aimed at streamlining the data collection process for competitive market analysis and product comparison. The proposed solution leverages web scraping tools such as Selenium and BeautifulSoup to dynamically gather product details—such as product names, prices, descriptions, and specifications by parsing relevant HTML elements.

The scraped data undergoes preprocessing to ensure consistency and quality, followed by a summarization phase using a BART-based model to create concise summaries of product information. This step helps reduce the complexity of the extracted data, ensuring that the relevant information is easily accessible for further analysis. The system aims to enhance the efficiency of data handling, reducing manual intervention and errors.

An interactive question-answering component is integrated into the pipeline using a DistilBERT model, allowing users to query specific details from the summarized content. The extracted and summarized data is stored in a structured format, facilitating further analysis and reporting. This project demonstrates the feasibility and efficiency of an automated, end-to-end system for large-scale product data collection, processing, and retrieval, offering significant value for applications in e-commerce analytics and competitive intelligence.

TAMIL ABSTRACT

ACKNOWLEDGEMENT

It is my privilege to express my deepest sense of gratitude and sincere thanks to **Dr. S. BAMA,** Associate Professor, Department of Information Science and Technology, College of Engineering, Guindy, Anna University, for her constant supervision, encouragement, and support in my project work. I greatly appreciate the constructive advice and motivation that was given to help me advance my project in the right direction.

I am grateful to **Dr. S. SWAMYNATHAN,** Professor and Head, Department of Information Science and Technology, College of Engineering Guindy, Anna University for providing us with the opportunity and necessary resources to do this project.

I would also wish to express my deepest sense of gratitude to the Members of the Project Review Committee: **Dr. S. SRIDHAR,** Professor, **Dr. G. GEETHA,** Associate Professor, **Dr. D. NARASHIMAN,** Teaching Fellow Department of Information Science and Technology,College of Engineering Guindy, Anna University, for their guidance and useful suggestions that were beneficial in helping me improve my project.

I also thank the faculty member and non teaching staff members of the Department of Information Science and Technology, Anna University, Chennai for their valuable support throughout the course of our project work.

Mouliraj A K
(2023176001)

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

In today's highly competitive e-commerce environment, businesses and consumers face significant challenges in accessing accurate and up-to-date product information from multiple sources. The vast amount of unstructured product data scattered across websites complicates the tasks of market analysis, product comparison, and pricing strategy development. Manually collecting, cleaning, and organizing this data is labor-intensive, time-consuming, and prone to inconsistencies. To address this issue, there is a need for an automated solution that can efficiently extract comprehensive product information from diverse online sources, summarize it into a manageable format, and allow for interactive querying. Such a system would facilitate real-time insights into product attributes, availability, and pricing. By enabling on-demand information retrieval, businesses could make informed decisions and improve operational efficiency in market analysis and inventory management.

1.2 OBJECTIVE

The objective of this project is to develop an automated system that efficiently extracts, summarizes, and provides interactive querying of product information from multiple online sources. This system aims to retrieve detailed product data such as names, prices, descriptions, specifications, and availability using web scraping techniques, while ensuring data consistency and accuracy through processing and structuring. Leveraging advanced text summarization models, the system will condense large datasets into concise, interpretable summaries, enabling users to gain insights quickly. Additionally, an interactive question-answering interface will allow users to query specific details within the summarized information, offering a flexible and user-friendly way to engage with product data. This processed information will be stored in a structured format, facilitating further analysis, reporting, and integration with business intelligence tools, ultimately supporting real-time, data-driven decision-making in market analysis and competitive intelligence

1.3 OVERVIEW

This project addresses the need for an automated system that can streamline the collection,processing, and analysis of product information from various e-commerce platforms. As the digital marketplace grows, businesses require efficient ways to access up-to-date data on product attributes, pricing, and availability to make informed decisions. Traditional manual methods of data gathering and comparison are time-consuming, error-prone, and resource-intensive. To overcome these challenges, this project introduces a comprehensive, automated pipeline for extracting, summarizing, and querying product data in real time. The project leverages web scraping tools, primarily Selenium and BeautifulSoup, to retrieve product details such as names, prices, descriptions, specifications, and availability. Extracted data is then preprocessed and organized to ensure accuracy and consistency. Using a NLP based summarization model, the system condenses extensive data into brief, coherent summaries, making large volumes of information more accessible and interpretable. Additionally, a question-answering interface powered by a fine-tuned NLP model allows users to interact with the data, retrieving specific information on demand. This system also includes data storage capabilities in a structured format, enabling further analysis and facilitating reporting or integration with other business intelligence tools. The project demonstrates a practical, scalable solution for businesses and analysts, offering insights into market trends, competitive pricing, and product availability with minimal manual intervention. This automated approach to data collection and analysis empowers users to make timely, data-driven decisions in the fast-paced world of e-commerce.

1.4 ORGANIZATION OF REPORT

This section gives an overview of the project structure. It outlines what each chapter will cover and the ordering sequence. This serves as a roadmap to understand how the project develops.

Chapter 2: This chapter provides a summary of the literature review that highlights the earlier research in the field of Web Information retrival, Text Summarization and Question Answering.

Chapter 3: This chapter covers the project's system design. And also defines the different modules of the project.

Chapter 4: This chapter covers the algorithms of different modules along with the their sample output.

Chapter 5: This chapter displays the project's final outcome, comparing the models to identify the best among them.

Chapter 6: This chapter summarizes the key findings of the project, highlighting the effectiveness of using automated information retrieval systems to collect data. It also discuss the limitations that were observed. Additionally, this chapter outlines future work that could further enhance the prediction capabilities.

CHAPTER 2

LITERATURE SURVEY

2.1 OVERVIEW

In this chapter of the literature review, the application of web scraping and natural language processing (NLP) techniques in the domain of e-commerce data extraction, summarization and interactive querying. It explore studies that use web scraping frameworks like Selenium and BeautifulSoup to collect structured data from online platforms, as well as the role of advanced NLP models such as BART, DistilBERT, and Roberta in tasks that involve text summarization and question-answering. The review will cover research focused on automated product information retrieval, the impact of text summarization on data interpretability, and interactive query systems for enhanced user engagement

Automated information retrieval systems have undergone significant advancements, integrating technologies for web scraping, preprocessing, summarization, and question-answering. These systems collectively aim to streamline information extraction and enhance usability across domains.

2.1.1 Web Scraping

Web scraping is a fundamental module that involves extracting structured data from unstructured web content. Traditional approaches often relied on predefined rules and knowledge of webpage DOM structures(Bhardwaj et al., Liu et al.) [1][2]. However, modern systems incorporate machine learning and NLP techniques, enabling dynamic content scraping and overcoming challenges like CAPTCHA and dynamic rendering(Bhardwaj et al., Reang et al.,Reddy and Guha) [1][3][4]. Bhardwaj et al. demonstrated the integration of Named Entity Recognition (NER) with web scraping, while Reddy and Guha highlighted the use of web crawlers for automated text collection from diverse sources[4][1].

2.1.2 Text Preprocessing

Text preprocessing ensures that raw data extracted through scraping is cleaned and structured for downstream tasks. Techniques such as tokenization, stemming, removal of HTML tags, and stop-word elimination are standard practices (Bhardwaj et al.; Liu et al.)[2]. Advanced systems use deep learning for preprocessing to better manage unstructured content, as described in the works by Reang et al. and Bhardwaj et al.[3][1]. These preprocessing techniques significantly reduce noise, improving the performance of summarization and QA models(Liu et al., Reddy and Guha)[2][4]

2.1.3 Summarization

Text summarization methods have advanced from extractive approaches like TextRank and LexRank to abstractive models powered by transformers like BART, T5, and PEGASUS (Reddy Guha, 2023; Bhardwaj et al., Liu et al.)[4][1][2].Extractive approaches rely on clustering techniques and ranking algorithms (Liu et al., 2024; Bhardwaj et al., 2021)[1][2]. In abstractive summarization generates coherent, human-like summaries and has seen widespread use in chatbots and automated customer service systems (Reddy and Guha, 2023; Reang et al., 2024)[4][3]. Hybrid approaches combining these two methods have also gained traction, leveraging the strengths of both for robust summarization (Liu et al., Reddy and Guha)[4][2].

2.1.4 Question-Answering

Question-answering (QA) systems provide an interactive layer for information retrieval, allowing users to query systems for specific insights. Early systems relied on rule-based approaches and keyword matching, but recent advancements have adopted transformer models like RoBERTa and DistilBERT (Reang et al., Reddy and Guha)[3][4]. These systems excel in understanding context and generating precise answers, making them integral to conversational AI platforms like chatbots (Reang et al., Reddy and Guha, Liu et al.)[3][4][2]. Additionally, the integration of summarization with QA enhances their ability to provide concise responses to complex queries (Reddy and Guha)[4]

2.1.5 Integration and Challenges

While these modules independently showcase robust functionalities, their integration poses challenges in scalability, latency, and accuracy. Systems like those proposed by Liu et al. and Bhardwaj et al. address these issues by combining dynamic scraping, efficient preprocessing, and state-of-the-art NLP models[2][1].Reddy and Guha further demonstrate the value of cohesive pipelines in building real-time, user-friendly systems tailored for specific applications[4].

2.2 SUMMARY OF THE STUDY

The literature review highlights significant advancements in web scraping, text summarization, and question-answering systems, showcasing their importance in automating large-scale data extraction and processing. Traditional web scraping methods, which depend on predefined DOM structures, have been significantly enhanced with the adoption of machine learning techniques such as Named Entity Recognition (NER) and NLP-based summarization. These innovations, as demonstrated by Bhardwaj et al.[1] enable the dynamic and automated extraction of structured data from diverse and complex web page layouts. This advancement is particularly critical for applications like e-commerce, where product page designs and data structures often vary widely, requiring systems to adapt dynamically to these challenges.

In the field of text summarization, there has been a paradigm shift from extractive methods to hybrid and abstractive approaches. Advanced models like BERT, BART, and T5 have enabled the rephrasing and condensation of content, ensuring greater coherence and relevance. As reviewed by Reang et al.[3] these techniques are particularly valuable for processing large volumes of unstructured data into concise and interpretable formats. Such advancements are essential for systems that aim to provide meaningful summaries of extensive data, allowing users to quickly grasp critical information without wading through excessive details.

Conversational systems have also seen remarkable progress through the integration of summarization and question-answering capabilities. Studies by Bhardwaj et al.[1] illustrate how chatbots and interactive systems now leverage models like LexRank and T5 to enable users to query extensive datasets and retrieve specific insights in real-time. These advancements showcase the potential of AI-driven systems to provide scalable, automated solutions for real-time data analysis. By integrating web scraping, summarization, and interactive querying, these systems demonstrate how modern technologies can effectively process and present vast amounts of information in a user-friendly manner.

CHAPTER 3

SYSTEM ARCHITECHTURE OF REAL TIME AUTOMATED SPARE PARTS INFORMATION RETRIEVAL

The system architecture for real time automated spare parts information retrival involves four modules - data input and site retrieval module, Web scraping module, Summarization module, Question Answering module. The process begins with data input, where the user and URLs give products part number and descriptions are retrieved for that and accessed using automated browsing with Selenium. The web data extraction module parses the HTML content from these pages, which utilizes BeautifulSoup to locate specific product attributes like name, price, and specifications through targeted element matching. In the summarization module, extracted information undergoes cleaning and chunking, and is then summarized using a BART-based model to generate concise descriptions of the product data. Finally, the Question Answering module incorporates a question answering system, allowing users to query the summarized data for specific details through a question answering model.

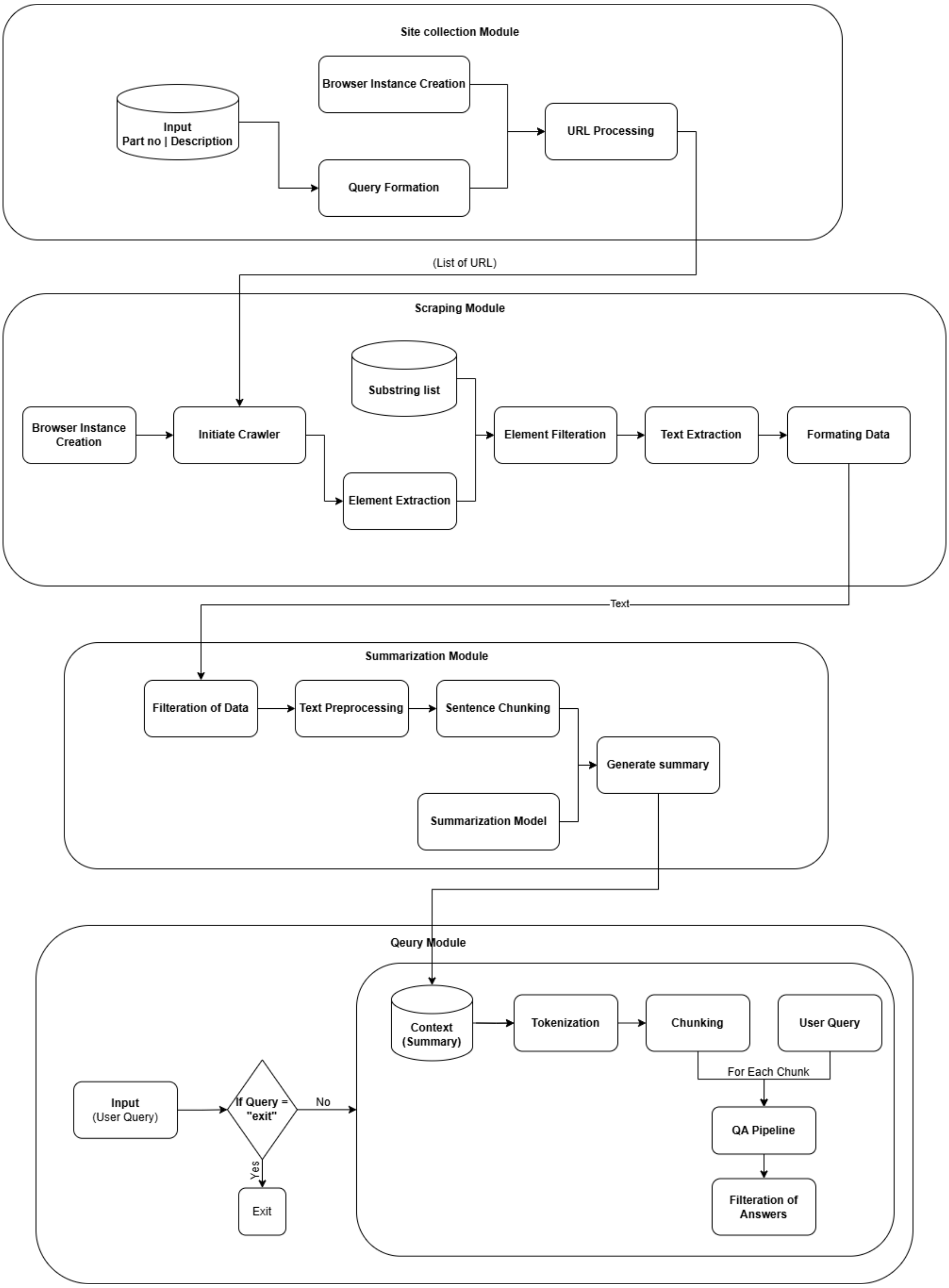


Figure 3.1: Architecture of eal time automated spare parts information retrival

3.1 MODULES

3.1.1 SITE COLLECTION MODULE

This module is responsible for accessing and gathering product pages from various e-commerce platforms. First the input is gathered from the user(Eg: C4368, LIFT SUPPORT), where the dataset consists of Part number, MFR, Description, Alternates and Links.2 Then the input is formed into a query to get the resulting from the browser.Utilizing Selenium WebDriver and BeautifulSoup the href(link) of all the resulting sites are gathered and if the number of gathered URLs is less than 20 then the crawler will move to the next result page and gets the hrefs from there nd this continues till 20 links are gathered. Then these gathered URL list is transferred to the scraping module to exract the data.

3.1.2 SCRAPING MODULE

This module focuses on extracting structured product data from the HTML content. Using BeautifulSoup, First it gets the list of URLs from the site collection module and for each URL present in the list it initiates a crawler the crawles through the URL and extracts the data and the crawler retuns all the ids and classes present in the HTML page. A list of substrings is used to define the fields that could contain the data for corresponding. In data extraction, substring is matched with the list of classes provided by the crawler to identify relevant HTML classes and IDs that correspond to the required data fields, ensuring that each product attribute is captured accurately. Once these elements are identified, BeautifulSoup is used to extract the text from the selected elements filtering all the HTML tags.

3.1.3 SUMMARIZATION MODULE

This module focuses on extracting structured product data from the HTML content.It gets the data extracted by the crawler and removes unnesasary fields like the URL present the extracted data and includes fields like Product Name, Product Description, Product Price, Alternates available so that the final summary would be precise. Once the fields are filtered the data from all the fields are normalized and cleaned to remove it from any unwanted and repetitive data. Then the data are tokenized uinng Bart tkokenizer and the data is chunked with maximum chunk size of 200 and a minimum size of 30. here the checking is done in context with the sentence present inorder to maintain the context of the data. Once over, Each chunk is given as a input to the summarize(BART,Flan-T5) to generate the summary. Here I have used Facebooks Bart large CNN model and Googles Flan-T5 Large model with the parameter-Chunks, Maximum size of the summary, Minimum size of the summary and both models gets same input and parameters to perform comparison.

3.1.4 QUERY MODULE

This module enables users to interactively query the summarized product data, providing an intuitive and responsive interface for on-demand information retrieval. First it gets the summaries generated by the previous module and uses it as the context to answer the queries of the user. The User will give the query and it be checked for escape tags inorder to stop the querying process. If the query is not a escape tag then the query is passed to the Question answering pipeline. Here in this pipeline I have used DistillBERT and deepset Roberta models that gets the parameters-Question, context. here the question will be the user query and context is the summary generated. Before using the summary as a context first it is Tokenized and chunked based on the sentence similar to the process used in the summarization module and for each chunk a answer is generated. For each answer a score is calculated based on the probability of the answer to the corresponding chunk and the answer with the highest confidence is provided as the final answer to the user.

CHAPTER 4

IMPLEMENTATION OF REAL TIME AUTOMATED SPARE PARTS INFORMATION RETRIEVAL

4.1 ENVIRONMENT SETUP

The following tools and technologies were used to develop and deploy the system:

- Programming Language: Python 3.X
- Web Scraping Frameworks:
  - Selenium: For dynamic web scraping and handling JavaScript-rendered pages.
  - BeautifulSoup: For parsing static HTML content.
- Data Processing and Analysis:
  - Pandas: For data manipulation and cleaning.
  - OpenPyxl: For reading, writing, and formatting Excel files.
- Natural Language Processing:
  - Hugging Face Transformers: For summarization and question-answering tasks.
  - NLTK: For text tokenization and preprocessing.
- WebDriver: ChromeDriver compatible with the installed Chrome browser version

4.2 OVERVIEW ALGORITHM

The proposed algorithm automates the process of extracting, summarizing, and analyzing product data from online sources. It begins by performing a Google search with a dynamic query, retrieving relevant URLs from the search results. Each URL is then processed using Selenium and BeautifulSoup to extract structured product details like names, prices, and part numbers, with dynamic content handled as needed. The extracted data is preprocessed and divided into manageable chunks for summarization using a pre trained NLP models. The summarized data is presented to the user and can be further queried interactively using a NLP based question answering pipeline. The results are stored in an Excel file, and the system concludes by releasing resources and logging any errors for future improvement. This algorithm ensures an efficient, scalable, and user-friendly approach to product data analysis.

4.2.1 ALGORITHM FOR SITE COLLECTION

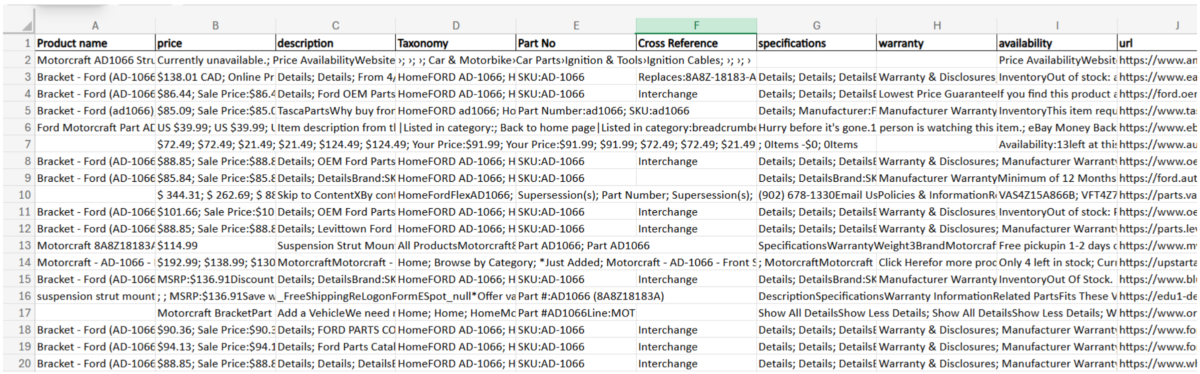
The site collection module is responsible for generating the list of URLs for the given part number. First, the part number and description is combined to create a query that will be used to get the results from google. Now the query is entered in the search bar and use the query to get a list of top 20 URLs from which data can be extracted. Create a list of all the URLs that can be used to extract the data.

Algorithm 4.1 Site Collection

```
Input: Part Number, Description
Output: List of url
Get the part number and description
combine part number and description as query x [ ]
initiate browser instance
identify search bar element
Enter the query in the search element
Create a list of URLs x [ ]
while length(x) ¡= 20 do
    Identify next page element
    Move to next page
end while
Return x [ ]
```

4.2.2 ALGORITHM FOR SCRAPING

It is responsible for extracting data from specific elements and distributing it into appropriate bins based on the keyword matching. The keywords are the substring that is used to filter the tag from which the data is extracted using Beautiful Soup as text. The sample output is given in 4.1



	A	B	C	D	E	F	G	H	I	J
1	**Product name**	**price**	**description**	**Taxonomy**	**Part No**	**Cross Reference**	**specifications**	**warranty**	**availability**	**url**
2	Motorcraft AD1066 Stru	Currently unavailable.; Price AvailabilityWebsite		›; ›; ›; Car & Motorbike›Car Parts›Ignition & Tools›Ignition Cables; ›; ›; ›					Price AvailabilityWebsit	https://www.an
3	Bracket - Ford (AD-1066	$138.01 CAD; Online Pr	Details; Details; From 4/	HomeFORD AD-1066; H	SKU:AD-1066	Replaces:8A8Z-18183-A	Details; Details; DetailsE	Warranty & Disclosures	InventoryOut of stock: a	https://www.ea
4	Bracket - Ford (AD-1066	$86.44; Sale Price:$86.4	Details; Ford OEM Parts	HomeFORD AD-1066; H	SKU:AD-1066	Interchange	Details; Details; DetailsE	Lowest Price Guarantee	If you find this product a	https://ford.oer
5	Bracket - Ford (ad1066)	$85.09; Sale Price:$85.0	TascaPartsWhy buy fror	HomeFORD ad1066; Ho	Part Number:ad1066; SKU:ad1066		Details; Manufacturer:F	Manufacturer Warranty	InventoryThis item requ	https://www.ta:
6	Ford Motorcraft Part AD	US $39.99; US $39.99; U	Item description from tl	|Listed in category:; Back to home page|Listed in category:breadcrumbe			Hurry before it's gone.1 person is watching this item.; eBay Money Back			https://www.eb
7		$72.49; $72.49; $21.49; $21.49; $124.49; $124.49; Your Price:$91.99; Your Price:$91.99; $91.99; $72.49; $72.49; $21.49					; 0Items -$0; 0Items		Availability:13left at thi:	https://www.au
8	Bracket - Ford (AD-1066	$88.85; Sale Price:$88.8	Details; OEM Ford Parts	HomeFORD AD-1066; H	SKU:AD-1066	Interchange	Details; Details; DetailsE	Warranty & Disclosures; Manufacturer Warranty		https://www.oe
9	Bracket - Ford (AD-1066	$85.84; Sale Price:$85.8	Details; DetailsBrand:SK	HomeFORD AD-1066; H	SKU:AD-1066		Details; DetailsBrand:SK	Manufacturer Warranty	Minimum of 12 Months	https://ford.aut
10		$ 344.31; $ 262.69; $ 88	Skip to ContentXBy cont	HomeFordFlexAD1066;	Supersession(s); Part Number; Supersession(s);		(902) 678-1330Email Us	Policies & InformationR	VAS4Z15A866B; VFT4Z7	https://parts.va
11	Bracket - Ford (AD-1066	$101.66; Sale Price:$10	Details; OEM Ford Parts	HomeFORD AD-1066; H	SKU:AD-1066	Interchange	Details; Details; DetailsE	Warranty & Disclosures	InventoryOut of stock: F	https://www.oe
12	Bracket - Ford (AD-1066	$88.85; Sale Price:$88.8	Details; Levittown Ford	HomeFORD AD-1066; H	SKU:AD-1066	Interchange	Details; Details; DetailsE	Warranty & Disclosures; Manufacturer Warranty		https://parts.lev
13	Motorcraft 8A8Z18183A	$114.99	Suspension Strut Moun	All ProductsMotorcraft8	Part AD1066; Part AD1066		SpecificationsWarrantyWeight3BrandMotorcraf		Free pickupin 1-2 days c	https://www.mv
14	Motorcraft - AD-1066 -	$192.99; $138.99; $130	MotorcraftMotorcraft -	Home; Browse by Category; *Just Added; Motorcraft - AD-1066 - Front S			; MotorcraftMotorcraft	Click Herefor more proc	Only 4 left in stock; Curı	https://upstarta
15	Bracket - Ford (AD-1066	MSRP:$136.91Discount	Details; DetailsBrand:SK	HomeFORD AD-1066; H	SKU:AD-1066	Interchange	Details; DetailsBrand:SK	Manufacturer Warranty	InventoryOut Of Stock.	https://www.bl
16	suspension strut mount	; ; MSRP:$136.91Save w	_FreeShippingReLogonFormESpot_null*Offer va		Part #:AD1066 (8A8Z18183A)		DescriptionSpecificationsWarranty InformationRelated PartsFits These V			https://edu1-de
17		Motorcraft BracketPart	Add a VehicleWe need r	Home; Home; HomeMc	Part #AD1066Line:MOT		Show All DetailsShow Less Details; Show All DetailsShow Less Details; W			https://www.or
18	Bracket - Ford (AD-1066	$90.36; Sale Price:$90.3	Details; FORD PARTS CO	HomeFORD AD-1066; H	SKU:AD-1066	Interchange	Details; Details; DetailsE	Warranty & Disclosures; Manufacturer Warranty		https://www.fo
19	Bracket - Ford (AD-1066	$94.13; Sale Price:$94.1	Details; Ford Parts Catal	HomeFORD AD-1066; H	SKU:AD-1066	Interchange	Details; Details; DetailsE	Warranty & Disclosures; Manufacturer Warranty		https://www.fo
20	Bracket - Ford (AD-1066	$88.85; Sale Price:$88.8	Details; Details; DetailsE	HomeFORD AD-1066; H	SKU:AD-1066	Interchange	Details; Details; DetailsE	Warranty & Disclosures; Manufacturer Warranty		https://www.wh

Figure 4.1: Output of Site collection and Scrapping Module

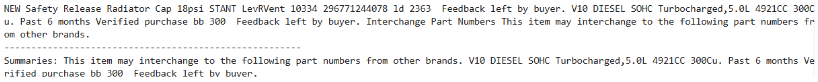
The output include the fields product name, price, description, taxonomy, part no, cross reference, specification, warranty, availability and URL. And these fields are filled with the data extracted from the sites and are arranged into appropriate fields along with the corresponding URL.

Algorithm 4.2 Scraping

```
Input: List of URL
Output: Dataframe containing Extarcted data
Get the list of URLs $x$ [ ]
Create DataFrame $df$ with keyword $k$ [ ] for each column
initiate Browser
for each $i$ of $x$ do
    open $i$ in browser
    for each $column$ in $df$ do
        for each $tag$ in $i$ do
            if $tag$ in $k$ then
                Extract data $d$ from that tag
            end if
            In $df$ append $d$ into appropriate column
        end for
    end for
end for
Return $df$
```

4.2.3 ALGORITHM FOR SUMMARIZATION

It Gets the Dataframe from the previous module and collects useful data from that, then the data cleaned of from any null values, excessive spaces, and any special characters. Then the cleaned data is tokenized and are divided into chunks with max size of 200, Here chunking is done based on the sentence to preserve the context of the sentence and if a chunk is too small then it is combined with the next chunk. then these chunks are used to generate summary and summary of each chunk is combined to form the final summary. In this summary is generated comparatively using two pretrained model namely BART and FLAN T5. The sample output is given in 4.2



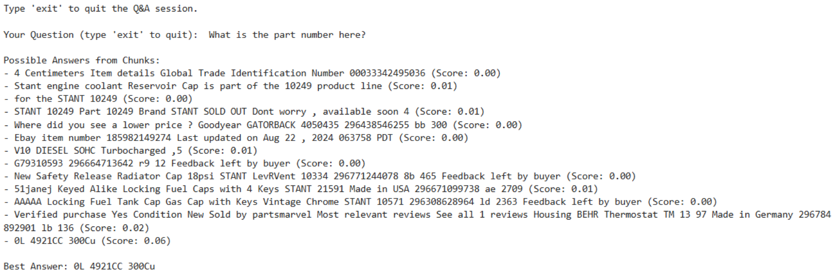
```
NEW Safety Release Radiator Cap 18psi STANT LevRVent 10334 296771244078 ld 2363  Feedback left by buyer. V10 DIESEL SOHC Turbocharged,5.0L 4921CC 300C
u. Past 6 months Verified purchase bb 300  Feedback left by buyer. Interchange Part Numbers This item may interchange to the following part numbers fr
om other brands.
-------------------------------------------------------
Summaries: This item may interchange to the following part numbers from other brands. V10 DIESEL SOHC Turbocharged,5.0L 4921CC 300Cu. Past 6 months Ve
rified purchase bb 300  Feedback left by buyer.
```

Figure 4.2: Output of Summary Module

This is an example output where summary generated for a single chunk is present and similarly for each chunk a summary is generated and are combined to form the final summary.

4.2.4 ALGORITHM FOR QUERYING

It Gets the generated summary from the previous module and uses it has the context to answer the users query. Here the summary is split into fixed sized chunks are then given into the qa pipeline along with the users query and each answer is scored based on its similarity to the chunk used to generate it and the answer with the highest score is selected as the final answer. The sample output is given in 4.3



```
Type 'exit' to quit the Q&A session.

Your Question (type 'exit' to quit):  What is the part number here?

Possible Answers from Chunks:
- 4 Centimeters Item details Global Trade Identification Number 00033342495036 (Score: 0.00)
- Stant engine coolant Reservoir Cap is part of the 10249 product line (Score: 0.01)
- for the STANT 10249 (Score: 0.00)
- STANT 10249 Part 10249 Brand STANT SOLD OUT Dont worry , available soon 4 (Score: 0.01)
- Where did you see a lower price ? Goodyear GATORBACK 4050435 296438546255 bb 300 (Score: 0.00)
- Ebay item number 185982149274 Last updated on Aug 22 , 2024 063758 PDT (Score: 0.00)
- V10 DIESEL SOHC Turbocharged ,5 (Score: 0.01)
- G79310593 296664713642 r9 12 Feedback left by buyer (Score: 0.00)
- New Safety Release Radiator Cap 18psi STANT LevRVent 10334 296771244078 8b 465 Feedback left by buyer (Score: 0.00)
- 51janej Keyed Alike Locking Fuel Caps with 4 Keys STANT 21591 Made in USA 296671099738 ae 2709 (Score: 0.01)
- AAAAA Locking Fuel Tank Cap Gas Cap with Keys Vintage Chrome STANT 10571 296308628964 ld 2363 Feedback left by buyer (Score: 0.00)
- Verified purchase Yes Condition New Sold by partsmarvel Most relevant reviews See all 1 reviews Housing BEHR Thermostat TM 13 97 Made in Germany 296784
892901 lb 136 (Score: 0.02)
- 0L 4921CC 300Cu (Score: 0.06)

Best Answer: 0L 4921CC 300Cu
```

Figure 4.3: Output of Querying Module

This is an example output where summary generated is used as context to answer the user query ”what is the part number?” which has been answer with part number in context with summary.

Algorithm 4.3 Summarization

```
Input: Dataframe
Output: Generated Summary
Initialize Tokenizer
Initialize Summarizer(BART—FLAN T5) pipeline
Get the list of useful Columns $c$ [ ]
Get the data $d$ from the specified columns of the dataframe
Strip $d$ of whitespace, Special characters
Tokenize $d$
Get Unique values from $d$
Join Unique values $u$
Initialize $current_c hunk$ and $chunk$ and $Temp$
Tokenize $u$ based on sentence
for $Sent$ in $u$ do
    append $Sent$ and $current_c hunk$, $Temp$ into $chunk$
    if Length of $chunk <= Maxlen$ then
        append $chunk$ into $current_c hunk$
    end if
    if Length of $chunk < Minlen$ then
        append $Sent$ into $Temp$
    end if
end for
Return d
```

Algorithm 4.4 Querying

```
Input: Generated Summary, User query
Output: Answer for user query
Get the Summary $s$
Initialize Tokenizer and QA pipeline
while $True$ do
    Get User input $q$
    if $q$ == ”Exit” then
        break
    end if
    Tokenize $s$ based on sentence Initialize $score$ to store answer and score as a dictionary
    for $sent$ in $s$ do
        append $Sent$ and $current_{c}hunk$, $Temp$ into $chunk$
        if Length of $chunk <= Maxlen$ then
            append $chunk$ into $current_{c}hunk$
        end if
        if Length of $chunk < Minlen$ then
            append $Sent$ into $Temp$
        end if
    end for
    for $chunk$ in $s$ do
        if $score$ ¿ $Final$ then
            $Final = score$
        end if
    end for
    Print $Final$
end while
```

CHAPTER 5

RESULTS AND ANALYSIS OF REAL TIME AUTOMATED SPARE PARTS INFORMATION RETRIEVAL

5.1 EVALUATION METRICS

This study uses two primary evaluation frameworks: ROUGE and BERTScore. Below is a detailed explanation of these metrics.

5.1.1 ROGUE(Recall-Oriented Understudy for Gisting Evaluation)

ROUGE is a set of metrics that compares the n-grams (e.g., unigrams, bigrams) in the generated text with those in the reference text. It is primarily focused on capturing lexical overlap, which serves as an indicator of similarity between the generated summary and the reference summary.

- ROUGE-1(Unigram Overlap) - It measures the overlapping of individual words between the generated and referenced content, this helps us understand how well the model captures the basic content of the reference summary.
- ROUGE-2(Bigram Overlap) - It measures the overlapping of two word sequence between the generated and referenced content, this helps us understand how well the model can generate coherent and contextually relevant phrases.

- ROUGE-L(Longest Common Subsequence) - It Considers the longest common subsequence of words between the generated and referenced content, this helps us to understand the model sentence-level structure and fluency by evaluating how well the order of words in the generated summary aligns with the reference
- Formulas
  - Precision(P) - It measures how many of the overlapping n-grams in the generated summary are also present in the reference summary. A high precision value indicates that most of the generated content is relevant to the reference.

$$\text{Precision} = \frac{\text{No of overlapping n-grams}}{\text{Total n-grams in generated text}}$$

  - Recall(R) - It measures how many of the overlapping n-grams in the reference summary are present in the generated summary. A high recall value suggests that the generated summary covers a significant portion of the reference content.

$$\text{Recall} = \frac{\text{No of overlapping n-grams}}{\text{Total n-grams in reference text}}$$

  - F1Score(F) - It is the harmonic mean of precision and recall, providing a balanced evaluation of the generated summary. A high F1 score indicates that the summary is both relevant and comprehensive.

$$F_1\text{-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.1.2 BERTScore

BERTScore is an advanced evaluation metric that measures the semantic similarity between a generated summary and a reference summary. Unlike traditional metrics like ROUGE, which rely on exact n-gram matches, BERTScore leverages contextual embeddings from pre-trained transformer models (e.g., BERT) to capture deeper, nuanced relationships between words and phrases.

5.1.2.1 Metrics in BERTScore

- Precision(P) - It measures the proportion of tokens in the generated summary that are semantically similar to tokens in the reference summary.

$$\text{Precision} = \frac{1}{|S|} \sum_{x \in S} \max_{y \in R} \text{cos-sim}(x, y)$$

- Recall(R) - It measures the proportion of tokens in the reference summary that are semantically similar to tokens in the generated summary.

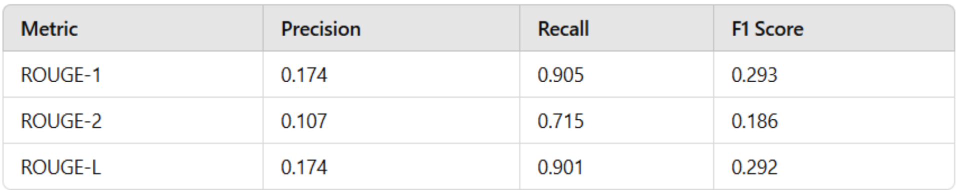
$$\text{Recall} = \frac{1}{|R|} \sum_{y \in R} \max_{x \in S} \text{cos-sim}(x, y)$$

- F1Score(F) - Combines precision and recall to provide a balanced evaluation of semantic similarity.

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

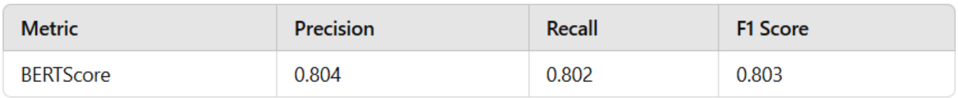
5.2 RESULTS

In this study, both models are tested on same data with the same parameters and follows the same steps with the models being replaced for getting the corresponding results for comparison. Here the data is cleaned of any special characters and unwanted whitespace and are then normalized. once done they are tokenized and chunked based on their sentence formation, Here each chunk is of a maximum size of 200 and a minimum size of 30.and for each chunk summary is generated separately and are then combined to form the final summary.



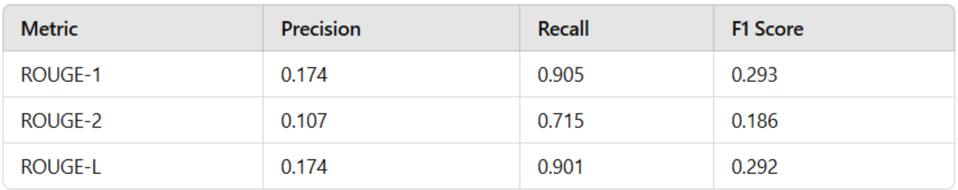
Metric	Precision	Recall	F1 Score
ROUGE-1	0.174	0.905	0.293
ROUGE-2	0.107	0.715	0.186
ROUGE-L	0.174	0.901	0.292

Figure 5.1: ROGUE Scores of Flan-T5



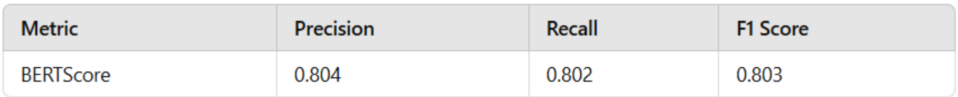
Metric	Precision	Recall	F1 Score
BERTScore	0.804	0.802	0.803

Figure 5.2: BERT Scores of Flan-T5



Metric	Precision	Recall	F1 Score
ROUGE-1	0.174	0.905	0.293
ROUGE-2	0.107	0.715	0.186
ROUGE-L	0.174	0.901	0.292

Figure 5.3: ROGUE Scores of BART



Metric	Precision	Recall	F1 Score
BERTScore	0.804	0.802	0.803

Figure 5.4: BERT Scores of BART

5.3 Comparison

The results of the comparative evaluation between BART (Model 1) and Flan-T5 (Model 2) reveal notable differences in their summarization performance, both quantitatively and qualitatively. BART achieved higher recall and F1 scores across ROUGE metrics, with ROUGE-1 F1 at 0.489, ROUGE-2 F1 at 0.351, and ROUGE-L F1 at 0.489, showcasing its ability to generate detailed, context-rich summaries. On the other hand, Flan-T5 demonstrated lower recall, with ROUGE-1 F1 at 0.293, ROUGE-2 F1 at 0.186, and ROUGE-L F1 at 0.292, indicating a more concise summarization approach. For BERTScore, BART outperformed Flan-T5 with an F1 score of 0.857 compared to 0.803, highlighting its superior semantic alignment with reference summaries. However, Flan-T5 showed strengths in producing brief and precise outputs, which may be advantageous in applications requiring concise summaries. While BART's the quality of using more words than necessary ensures comprehensive coverage, it may include redundant information, whereas Flan-T5 trades off some recall for higher computational efficiency and brevity. These findings underscore the trade-offs between the models, with BART being more suitable for detailed summarization tasks and Flan-T5 excelling in scenarios prioritizing clear and precise expresion.

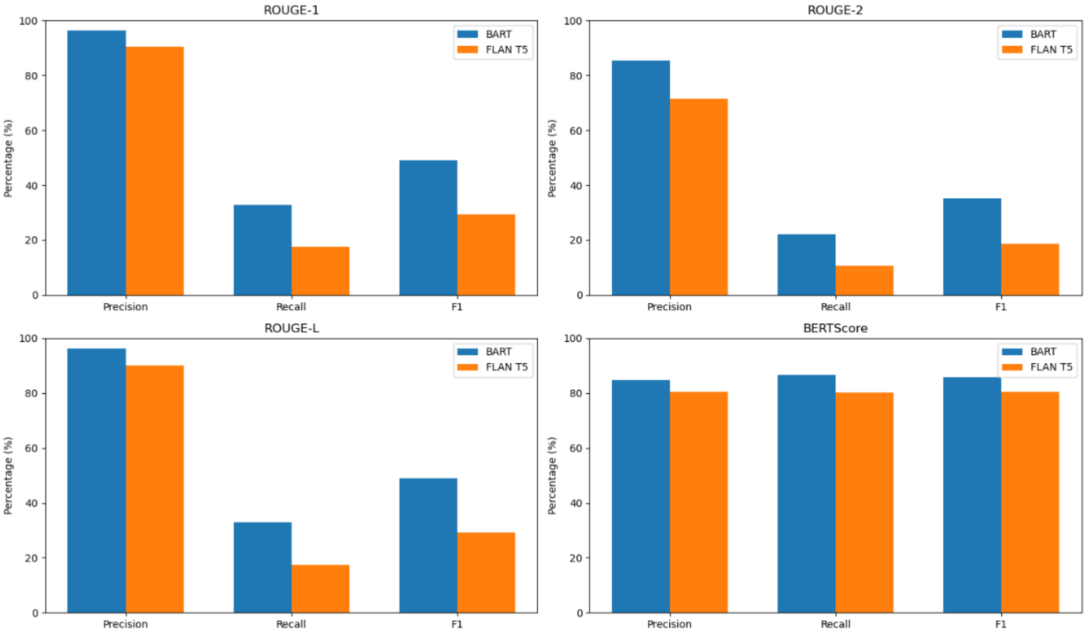


Figure 5.5: Comparison between Bart and Flan-T5

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The case study of “Real Time Automated Spare Parts Information Retrieval” indicate that what automation offers to offset the difficulties of acquiring, processing, and analysing product data from multiple online stores. In the current world, massive information floods product websites, and the traditional method of performing research by manually writing down notes and recording the necessary information is slow and prone to many mistakes. This work fills that gap by providing a convenient solution to the extraction, summarization and querying that could be so time-consuming using manual methods. With help of Selenium and BeautifulSoup tools, the system identifies and gathers structured and unstructured data from Web sources. Extensions in BART and Flan-T5 improve benefit of the completed system that rapidly compiles huge datasets into conveniently searchable form through summarization. These summaries will help users to selectively abstract certain information, without having to wade through mountains of text data. Furthermore, the inclusion of an NLP-based question answering module guarantees that particular details can be requested at run-time, giving the system real-time capability as well as user focus.

The outcome of this study shows how the proposed system helps minimize the time and mistakes arising from manual data processing. Comparative analyses of the NLP models presented here provided additional insights into their performance particularly BART which is very suitable for the generation of detailed summaries and Flan-T5 that was highly efficient when generating compact summaries. This makes the system flexible and appropriate for numerous vocations, such as for characterization of the market and inventory use.

Therefore it can be concluded that the “Real Time Automated Spare Parts Information Retrieval” proposal is spearheading a way towards the automation of data driven methods. Its solid design and approach to NLP technologies create a favorable basis for further development, providing great utility for companies functioning in the constantly growing and evolving environment of e-commerce.

6.2 FUTURE WORK

The “Real Time Automated Spare Parts Information Retrieval” system developed in this research works, though there are many directions for future enlargement of the system to enhance its efficiency and flexibility. It was found that statically fine-tuning the models for domain-specific data improves their accuracy and relevance, as well as incorporating more e-commerce platforms and APIs. Extending the functionality with interactive dashboards and visual reporting instruments would enhance readability of the data, while strict adherence to privacy standards should solidify user confidence and security of the undertaken process. The system could also use its ability to scale the system as the data grows bigger and to incorporate the outputs into enterprise scale applications and business processes along with a the ability integration into tools like Tableau or Power BI for instance. Moreover, having added functionalities such as the dynamic pricing analysis and the recommendation system that is based on AI algorithms, the system can be transformed into a true means of managing pricing models, as well as delivering valuable recommendations.

REFERENCES

[1] Bhavya Bhardwaj, Syed Ishtiyaq Ahmed, J Jaiharie, R Sorabh Dadhich, and M Ganesan. Web scraping using summarization and named entity recognition (ner). In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 261–265, 2021.

[2] Wenjun Liu, Yuyan Sun, Bao Yu, Hailan Wang, Qingcheng Peng, Mengshu Hou, Huan Guo, Hai Wang, and Cheng Liu. Automatic text summarization method based on improved textrank algorithm and k-means clustering. *Knowledge-Based Systems*, 287:111447, 2024.

[3] Rujeena Reang, Vasudev Dehalwar, and R. K. Pateriya. Deep learning techniques for automatic text summarization: A review. In *2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pages 1–6, 2024.

[4] Kethireddy Maheedhar Reddy and Radha Guha. Automatic text summarization for conversational chatbot. In *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, pages 1–7, 2023.

[5] Zaema Dar, Muhammad Raheel, Usman Bokhari, Akhtar Jamil, Esraa Mohammed Alazzawi, and Alaa Ali Hameed. Advanced generative ai methods for academic text summarization. In *2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI)*, pages 1–7, 2024.

[6] Sudharson D, K. S. Thrisha Vaishnavi, S. Hariprakassh, B. Abiram, K. Saranya, and Poonam Tanwar. An abstractive summarization and conversation bot using t5 and its variants. In *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, pages 432–437, 2023.

[7] Pascal Wilman, Talia Atara, and Derwin Suhartono. Abstractive english document summarization using bart model with chunk method. *Procedia Computer Science*, 245:1010–1019, 2024. 9th International Conference on Computer Science and Computational Intelligence 2024 (ICCSCI 2024).

[8] Turan Goktug Altundogan, Mehmet Karakose, and Onur Tokel. Bart fine tuning based abstractive summarization of patients medical questions texts. In *2023 4th International Conference on Data Analytics for Business and Industry (ICDABI)*, pages 174–178, 2023.

[9] Abdelrahman A. Mosaed, Hanan Hindy, and Mostafa Aref. Bert-based model for reading comprehension question answering. In *2023 Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 52–57, 2023.

[10] Lulu Zhang, Junru Li, Dacheng Feng, and Junjie Sun. Design and implementation of web crawler based on 'internet +'data automatic extraction. In *2023 3rd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 594–598, 2023.

[11] A. Kitanovski, M. Toshevska, and G. Mirceva. Distilbert and roberta models for identification of fake news. In *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, pages 1102–1106, 2023.

[12] Deepak Kumar, Chaman Verma, and Nitika Nitika. Revolutionizing text summarization: A comprehensive comparative analysis of nlp-based models. In *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)*, pages 1–5, 2024.

[13] Ahcene Haddouche, Ikram Rabia, and Aicha Aid. Transformer-based question answering model for the biomedical domain. In *2023 5th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pages 1–6, 2023.

[14] Gaurav Sharma. Web crawling and scraping: A survey. In *2024 International Conference on Healthcare Innovations, Software and Engineering Technologies (HISET)*, pages 190–192, 2024.

[15] Ayat Abodayeh, Reem Hejazi, Ward Najjar, Leena Shihadeh, and Rabia Latif. Web scraping for data analytics: A beautifulsoup implementation. In *2023 Sixth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, pages 65–69, 2023.