

# Software Requirements Specification

**for**

# <AUTOMATIC TEXT SUMMERIZATION>

**Version 1.0**

**Supervised And Co-Supervised By:**

**Sir. Masood Hussain**

**Ms. Sonish Aslam(digital signature here)**

## Prepared by Group# 34

## Sir Syed University of Engineering & Technology

**Team**

## June 12th, 2023

|  |  |
| --- | --- |
| **Member Name** | **Primary Responsibility** |
| MUZAMMIL SARDAR ABBASI | ML MODEL CREATION AND LOGIN INTERFACE |
| IMAD KHAN | ML MODEL TESTING AND ADMIN PANEL |
| S.M ZEJAH ALI REHMANI | ML MODEL TRAINING AND REGISTRATION INTERFACE |
| M. AHSAN SIDDIQUE | ML MODEL VALIDATION AND ACCURACY CHECK |

***SRS for <AUTOMATIC TEXT SUMMARIZATION>***

**Table of Contents**

### Introduction

* 1. Purpose
  2. Scope
  3. Definitions, Acronyms and Abbreviations
  4. Acronyms and Abbreviations

### Project Planning and Management

* 1. SWOT Analysis
  2. Gantt Chart
  3. Work Breakdown Structure [WBS]

### Overall Description

* 1. Product Perspective
  2. Product Function
  3. Operating Environment
  4. Design and Implementation Constraints
  5. Assumptions and Dependencies

### Requirement Identifying Technique

* 1. Use Case Diagram
  2. Use Case Description

### Non- Functional Requirements

* 1. Performance Requirements
  2. Safety Requirements
  3. Security Requirements
  4. Software Quality Attributes
  5. Business Rules
  6. Interoperability
  7. Extensibility
  8. Maintainability
  9. Portability
  10. Reusability
  11. Installation

### Other Requirements

* 1. On-line User Documentation and Help System Requirements
  2. Purchased Requirements
  3. Licensing Requirements
  4. Legal, copyright, and Other Notices
  5. Applicable Standards
  6. Reports (Feedback, Invoice, User, Usage, Balance Sheet, Executive Summary, etc.)

1. **References**

### Introduction

* 1. **Purpose**:

The purpose of our automatic textual content summarization (ATS) application is to help produce the precis of any article containing any sort of crucial and applicable information from the unique documented textual content. Summarization is the venture of condensing a piece of a file into its shorter variant, which lessens the initial record’s size even as retaining the important piece of information intact.

* 1. **Scope:**

Our project is inspired by the huge problem people face when trying to read lengthy articles and blogs available on the internet. This is time-consuming and less effective in the long term Simply because there are fewer ways to summarize the work they want, with the resources that are suggested on the internet. As a result, people are often left confused that how are they going to achieve their goals if they don’t have the proper resources at hand.

Using AI algorithms, our app addresses that issue and provides the user with the solution and guidance to help them achieve their desired summary with resources that are easily available to them.

* 1. **Definitions, Acronyms, and Abbreviations:**
* **ATS:** Automatic Text summarization (ATS) is currently a famous exploration region among experimenters. Automatic text summarization is the method of producing the subset of the primary textbook. This subset of the main text represents the entire text and the article’s main idea. ATS is the crucial subject of Natural Language Processing (NLP) and Data Mining (DM). This consists of the abstractive and extractive summaries of the text.
* **CRUCIAL INFORMATION HANDLER:** As a way to create the precis of the applicable cloth, the summarizer extracts additional crucial information (either words or sentences) from the input content material. Consistent with the literature, there are five distinct varieties of textual content summarizing methods
* **SINGLE TEXT HANDLER:** In single-file text summarizing, a document is used to accomplish the summation, and a single output report is produced. Created a technique for textual content summarizing in a record the usage of automatic keyword extraction. A discourse-based summarizer turned into created by way of Marcu et al. To assess the suitability of texts for summarizing the use of discourse-based methodologies inside the context of single information.
* **RECORD OVERLAP:** Records overloading is one of the maximum pressing troubles added on by way of the net's explosive increase. Because there's so much data available on the net, condensing the pertinent facts right into a precis could be useful to many people. For humans, manually summarizing significant portions of textbook material is taxing. Researchers had been experimenting with methods to decorate precis creation such that the summaries produced with the aid of way of humans and machines are identical. This study gives an extensive chance assessment of the textbook summarizing generalities, in conjunction with strategies, datasets, evaluation requirements, and uncharted regions for research. The 2 methodologies included in-depth in this work which can be the maximum considerably stated are extractive and abstractive. Evaluating the synopsis. Moreover, the arrival of suitable coffers and structural help within the contrast and replication of findings, supply competitiveness to deal with the troubles. in these paintings, diverse evaluation strategies for produced summaries
  1. **Acronyms, and Abbreviations:**

Here are some common acronyms and abbreviations related to automatic text summarization:

* ***NLP: Natural Language Processing***
* ***AI: Artificial Intelligence***
* ***ML: Machine Learning***
* ***DL: Deep Learning***
* ***ASR: Automatic Speech Recognition***
* ***POS: Part-of-Speech***
* ***TF-IDF: Term Frequency-Inverse Document Frequency***
* ***RNN: Recurrent Neural Network***
* ***LSTM: Long Short-Term Memory***
* ***CNN: Convolutional Neural Network***
* ***BERT: Bidirectional Encoder Representations from Transformers***
* ***ROUGE: Recall-Oriented Understudy for Gisting Evaluation***
* ***BLEU: Bilingual Evaluation Understudy***
* ***METEOR: Metric for Evaluation of Translation with Explicit Ordering***
* ***LSA: Latent Semantic Analysis***
* ***LDA: Latent Dirichlet Allocation***
* ***API: Application Programming Interface***
* ***GUI: Graphical User Interface***
* ***POS: Part-of-Speech***
* ***TTS: Text-to-Speech***

These acronyms and abbreviations are commonly used in discussions, research papers, and literature related to automatic text summarization.

### Project Planning and Management

### 2.1 SWOT Analysis:

### 2.1.1 INTRODUCTION:

Mobile internet use has increased greatly in popularity due to technology's quick development as a means of entertainment. The market for mobile applications is constantly growing, and their field of influence is getting bigger. The general population has begun to notice a new product. Information overloading is one of the maximum pressing troubles delivered on with the aid of the net's explosive expansion. Because there is a lot of information to be had on the internet, condensing the pertinent information into a precis might be useful to many people. For humans, manually summarizing sizeable quantities of textbook fabric is taxing. Researchers have been experimenting with techniques to improve precis advent such that the summaries produced by way of people and machines are the same. This takes look at offers an intensive threat evaluation of textbook summarizing generalities, consisting of strategies, datasets, assessment standards, and uncharted regions for investigation. The two methodologies protected in-depth in these paintings which might be the most broadly stated are extractive and abstractive. Comparing the synopsis. Moreover, the introduction of appropriate coffers and structural assistance in the comparison and replication of findings offer competitiveness to deal with the issues. In this work, numerous assessment techniques for produced summaries also are discussed. Sooner or later, towards the conclusion of this examination, some of the difficulties and areas for further investigation into textbook summary are cited that can be useful for implicit experimenters running in this field.

SWOT analysis is a strategic planning tool used to evaluate the strengths, weaknesses, opportunities, and threats of a particular entity or concept. Here's a SWOT analysis specifically focused on automatic text summarization:

**STRENGTHS:**

**Efficiency:** Automatic text summarization can quickly process and summarize large volumes of text, saving time and effort for users.

**Scalability:** The technology can be applied to diverse domains and accommodate varying document lengths, making it suitable for a wide range of applications.

**Information Filtering:** Automatic summarization helps users extract relevant information from a text, allowing for efficient decision-making and information retrieval.

**Language Independence:** The technology can be applied to multiple languages, enabling cross-lingual summarization and making it accessible to a global audience.

**Adaptability:** Automatic summarization algorithms can be trained and fine-tuned on specific datasets or domains, allowing for customization and improved performance in specific contexts.

**WEAKNESSES:**

**Semantic Understanding:** Automatic summarization algorithms may struggle with fully understanding the context, nuances, and underlying meaning of the text, leading to potential loss of information or misinterpretation.

**Subjectivity:** Different users may have varying preferences for what constitutes an effective summary, and automatic summarization may not always align with individual expectations.

**Evaluation Challenges:** Measuring the quality of summaries can be subjective, and evaluation metrics may not fully capture the nuances of a well-written summary.

**Language Complexity:** Automatic summarization faces challenges when dealing with complex sentence structures, figurative language, or domain-specific jargon, potentially leading to inaccuracies or inadequate summaries.

**OPPORTUNITIES:**

**Enhanced User Experience:** Advances in automatic summarization can improve user experiences by providing more accurate, relevant, and concise summaries tailored to individual needs.

**Personalization:** Incorporating user preferences and feedback can lead to personalized summarization systems that better align with individual requirements and improve user satisfaction.

**Multimodal Summarization:** Automatic summarization can be extended to include other modalities, such as images, audio, or video, providing users with a more comprehensive and holistic understanding of the content.

**Real-time Summarization:** Developing techniques for on-the-fly summarization can enable real-time applications, such as live event summarization or summarization in conversational agents.

**THREATS:**

**Ethical Considerations**: Automatic summarization may raise concerns about information bias, privacy, and the potential for manipulation or distortion of information.

**Legal Implications:** Summarization algorithms should adhere to copyright laws and intellectual property rights, as they involve extracting and reproducing content from the original text.

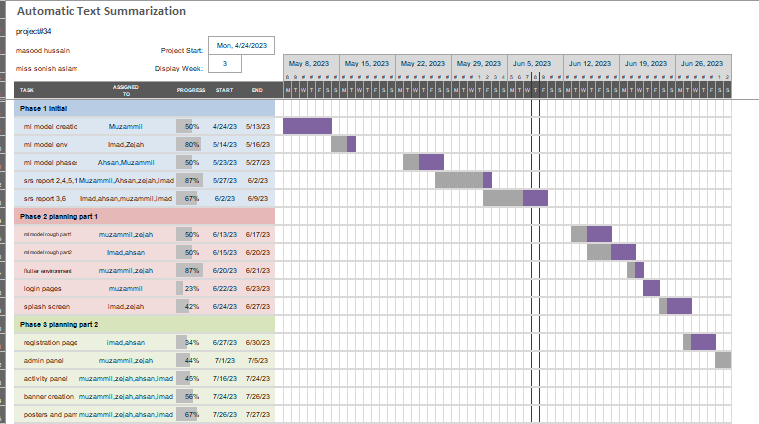
**Quality Assurance:** Ensuring the accuracy, reliability, and consistency of automatic summarization systems poses a challenge, as errors or biases in summaries can have significant consequences in critical applications.

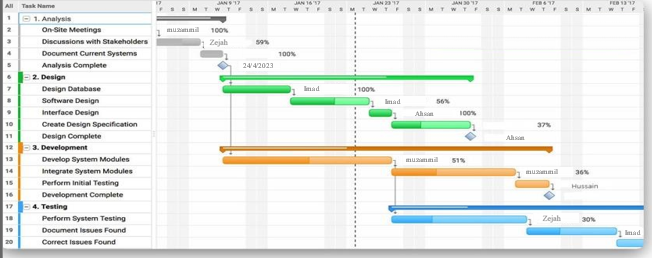
**Competition:** As automatic summarization gains popularity, increased competition may arise among different approaches and technologies, requiring continuous innovation to stay relevant.

This SWOT analysis provides an overview of the strengths, weaknesses, opportunities, and threats associated with automatic text summarization. It can help in understanding the current landscape of automatic summarization and guide decision-making and future development in the field.

### 

* 1. **Gantt Chart:**



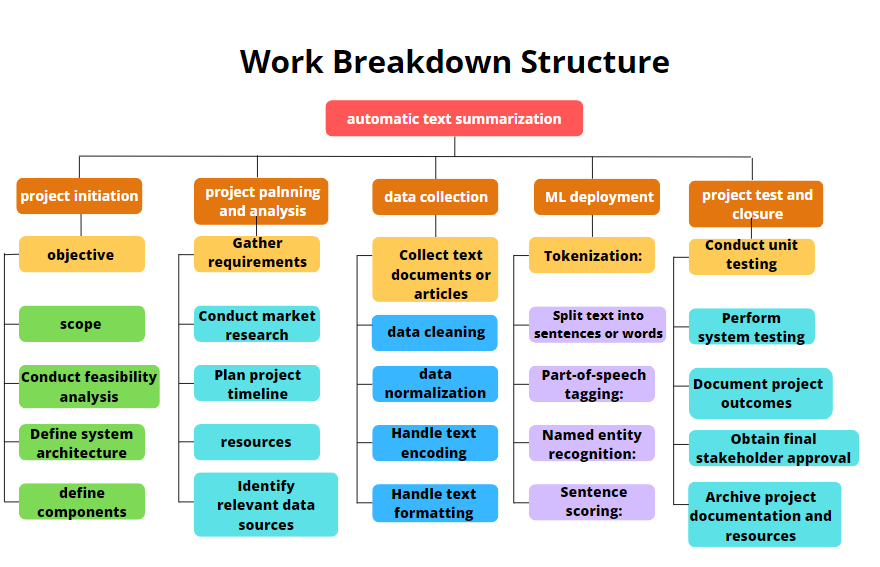


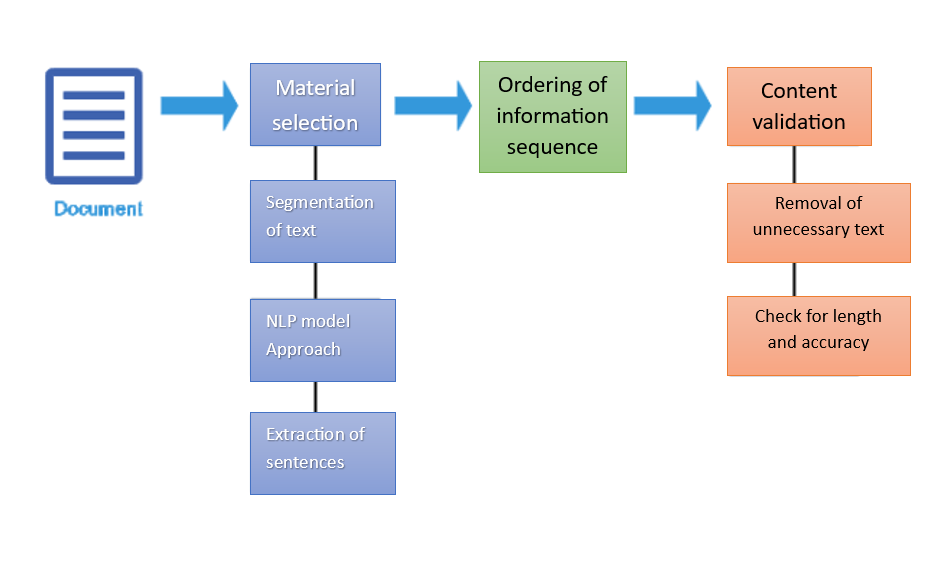
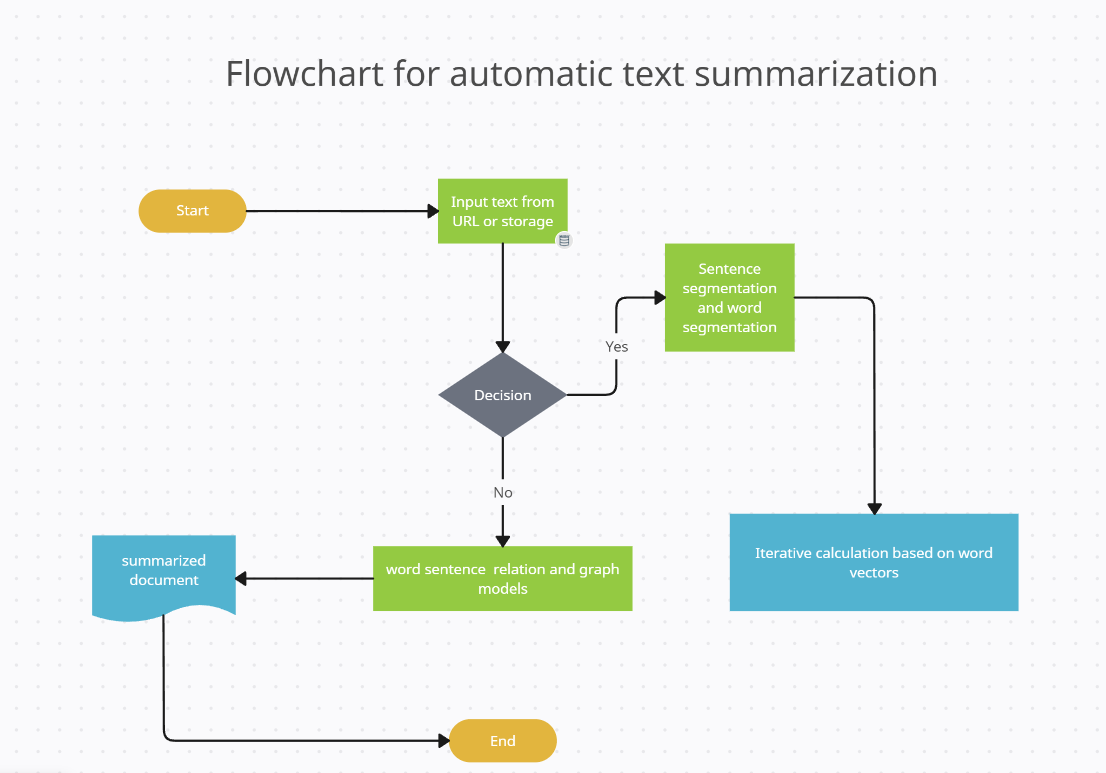
**Member 1:** S.M Zejah Ali Rehmani (Roll no: 2020-SE-154) will be responsible for creating datasets, incorporating datasets, developing logos and textures, as well as documentation.

**Member 2:** Muzammil Sardar Abbasi (Roll no: 2020-SE-165) will be responsible for the application's UI/UX design, development, and unit testing, as well as its API development, integration, artificial intelligence (AI), and database creation.

**Member 3:** Muhammad Ahsan Siddique (Roll no: 2020-SE-218) will be responsible for the documentation and system testing of the application on various devices.

**Member 4:** Imad Khan (Roll no: 2020-SE-176) will be responsible for developing some application functions as well as documentation.

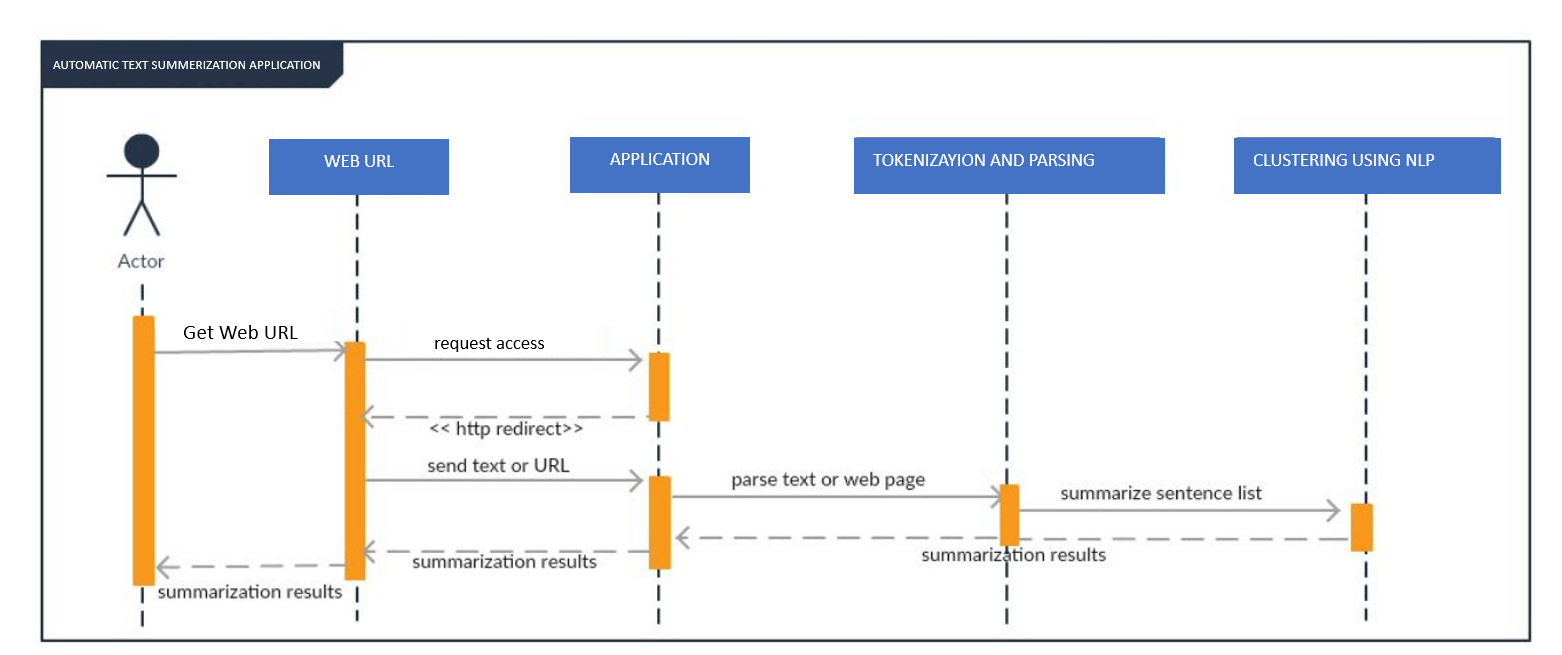
* 1. **Work Breakdown Structure [WBS]**
     + **Project Initiation**
     + project objective and scope
     + Stakeholders and requirements
     + Conduct analysis
     + Research and Planning
     + **Gather requirements from stakeholders**
     + Conduct market research and analysis
     + Define system architecture and components
     + Plan project timeline and resources
     + **Data Collection and Preprocessing**
     + Identify relevant data sources
     + Collect text documents or articles
     + Perform data cleaning and normalization
     + Handle text encoding and formatting
     + **Text Processing and Analysis**
     + Tokenization: Split text into sentences or words
     + Part-of-speech tagging: Identify grammatical components
     + Named entity recognition: Identify entities (people, places, etc.)
     + Keyword extraction: Identify important terms
     + Sentence scoring: Determine sentence importance
     + **Summarization Algorithm Development**
     + Develop extractive summarization algorithms
     + Evaluate and refine algorithms based on performance
     + Experiment with different techniques (e.g., graph-based, statistical)
     + Fine-tune parameters for optimal results
     + **System Integration**
     + Develop an interface for text input and output
     + Integrate the summarization algorithm with the system
     + Implement error handling and exception management
     + Test the system's functionality and usability
     + **User Interface and Experience**
     + Design a user-friendly interface for input and output
     + Implement features for customization and personalization
     + Ensure responsive and intuitive user interactions
     + Incorporate user feedback and iterate on improvements
     + **Testing and Evaluation**
     + Conduct unit testing for individual components
     + Perform system testing for end-to-end functionality
     + Evaluate the quality of generated summaries
     + Collect user feedback and make necessary adjustments
     + **Deployment and Maintenance**
     + Prepare the system for deployment in a production environment
     + Ensure scalability and performance optimization
     + Create documentation and user guides
     + Provide ongoing maintenance and support
     + **Project Closure**
     + Conduct a project review and lessons learned session
     + Document project outcomes and deliverables
     + Obtain final stakeholder approval
     + Archive project documentation and resources

****

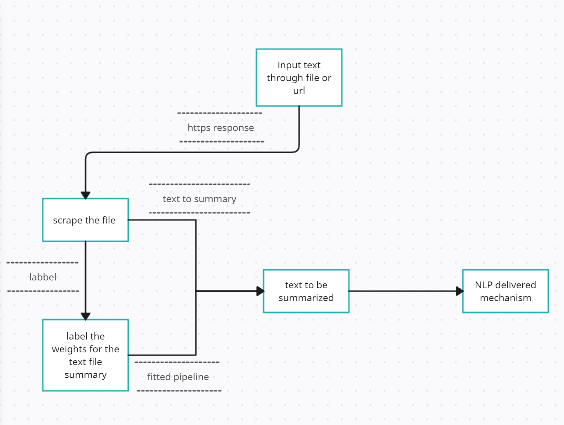
FLOWCHART DIAGRAM

NLP ALGORITHM DIAGRAM

UML SEQUENCE DIAGRAM

****

DATA FLOW DIAGRAM

****

### Overall Description

* 1. **Product Perspective:**

It is established that there are many websites on the internet that suggest you diets and articles that help you with your effectiveness, but most of them fail to provide useful solutions to the reading time problem, simply because people don’t have mentioned plans and resources available.

So, our app will be specially designed for the people which will provide them with all one solution using an AI recommendation system that will be based on AI model NLP.

* 1. **Product Function:**

**3.2.1 Input and Analyze:**

Our app will be able to analyze the provided URL and check the current state of the user-provided text data to be summarized.

**3.2.2 Data rearrange:**

Our app will rearrange the data provided to be taken into bits for the AI model to be scraped and used to label it.

* 1. **Operating Environment:**

The operating environment for text summarization can vary depending on the specific implementation and deployment scenario. However, here are some key elements typically involved in the operating environment for text summarization systems:

**Hardware Infrastructure:**

Servers or cloud-based infrastructure to host the text summarization system.

Sufficient computational resources to handle the processing requirements, especially for large-scale summarization tasks.

**Software Dependencies:**

Programming languages and frameworks for developing the summarization system.

Text processing libraries and tools for tasks such as tokenization, part-of-speech tagging, and named entity recognition.

Machine learning or natural language processing libraries for implementing the summarization algorithms.

**Text Data Sources:**

Access to a wide range of text data sources, such as news articles, research papers, or online content, depending on the application domain.

APIs or data connectors to retrieve or access text data from various sources.

Preprocessing Tools:

Text cleaning and normalization tools to remove noise, irrelevant characters, or formatting inconsistencies from the input text.

Stop word removal techniques to filter out common words that do not contribute significantly to the summary.

**Summarization Algorithms:**

Extractive or abstractive summarization algorithms, depending on the chosen approach.

Sentence scoring mechanisms to determine the importance or relevance of sentences.

Statistical or graph-based algorithms for identifying key phrases or concepts.

Evaluation Metrics:

Metrics and evaluation methodologies to assess the quality and effectiveness of the generated summaries.

Common evaluation metrics include ROUGE (Recall-Oriented Understudy for Gusting Evaluation), BLEU (Bilingual Evaluation Understudy), or other domain-specific metrics.

**Integration Interfaces:**

Input interfaces to accept text documents or articles for summarization.

Output interfaces to present the generated summaries in the desired format, such as plain text, HTML, or JSON.

**Customization and Configuration Options:**

Configuration settings to customize the summarization process based on specific requirements, such as summary length constraints or domain-specific terminology.

Scalability and Performance Optimization:

Techniques to handle large volumes of text data efficiently, such as parallel processing or distributed computing.

Caching mechanisms or indexing structures to improve performance when processing similar or recurring text data.

**Monitoring and Error Handling:**

Logging and monitoring mechanisms to track the system's performance, usage statistics, and potential errors.

Error handling and exception management to gracefully handle unexpected scenarios during the summarization process.

**Security and Privacy Considerations:**

Data security measures to protect sensitive or confidential text data.

Compliance with privacy regulations, especially if the text data contains personal or sensitive information.

* 1. **Design and Implementation Constraints:**

To understand design and implementation constraints while building an app using Flutter we need to understand this:

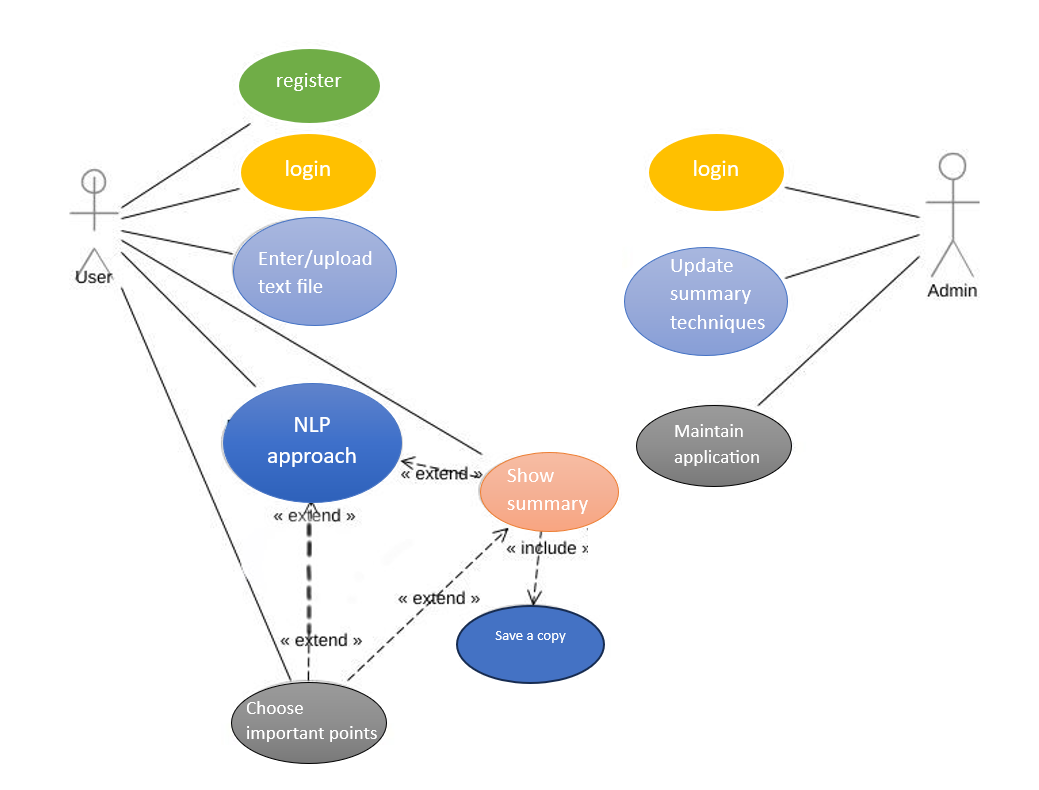
* A widget gets its own constraints from its parent. A constraint is just a set of 4 doubles: a minimum and maximum width, and a minimum and maximum height.
* Then the widget goes through its own list of children. One by one, the widget tells its children what their constraints are (which can be different for each child), and then asks each child what size it wants to be.
* Then, the widget positions its children (horizontally on the x-axis, and vertically on the y-axis), one by one.
* And, finally, the widget tells its parent about its own size (within the original constraints, of course).

As a result of the layout rule mentioned above, Flutter’s layout engine has a few important limitations:

* + A widget can decide its own size only within the constraints given to it by its parent. This means a widget usually can’t have any size it wants**.**
  + A widget can’t know and doesn’t decide its own position on the screen**,** since it’s the widget’s parent who decides the position of the widget.
  + Since the parent’s size and position, in its turn, also depends on its own parent, it’s impossible to precisely define the size and position of any widget without taking into consideration the tree as a whole.
  + If a child wants a different size from its parent and the parent doesn’t have enough information to align it, then the child’s size might be ignored. We need to be specific when defining alignment.
  1. **Assumptions and Dependencies:**
* If no users create any exercise routines, there won't be any listed for users in the application. Users won't be able to log in or access any application features if there is no internet connection.
* Unregistered users won't be able to utilize the software application.
* After accepting the terms of the service agreement, users will only be able to register and access the software.

### Requirement Identifying Technique

* 1. **Use Case Diagram**

****

* 1. **Use Case Description**

In this diagram, we have two main actors: The User and the Automatic Text Summarization System.

The User interacts with the system and can perform the following actions:

**Enter Text**: The User provides the input text that needs to be summarized.

**View Summary:** The User requests to see the generated summary of the input text.

The Automatic Text Summarization System, represented as a single entity, performs the core functionalities of the system. It includes the following use cases:

**Process Text:** The system processes the input text by performing various tasks such as tokenization, part-of-speech tagging, named entity recognition, and other preprocessing

**Generate Summary:** Based on the processed text, the system applies automatic summarization algorithms to generate a concise summary of the input text. Return Summary: The system returns the generated summary to the User, who can then view it.

### Non- Functional Requirements

* 1. **Performance Requirements:**

Our application performance is fast and responsive which provides fast access to data from Firebase which takes 1-2 seconds to fetch data, and which is dependent upon user connection speed. When considering the performance requirements for an automatic text summarization Android application, several factors should be considered to ensure optimal user experience and efficient processing. Here are some key performance requirements to consider:

**Responsiveness:** The application should provide a responsive user interface, ensuring that users experience minimal delays or lag when interacting with the app. The summarization process should be executed efficiently, allowing users to receive summaries in a timely manner.

**Processing Speed:** Efficient text summarization algorithms and techniques should be employed to achieve fast processing speeds. Users expect the app to generate summaries quickly, even when dealing with large volumes of text.

**Scalability:** The application should be designed to handle varying workloads and accommodate increasing amounts of text data. It should be capable of processing long documents or multiple documents concurrently without significant performance degradation.

**Resource Usage:** Optimize the app's resource utilization, such as CPU, memory, and battery consumption, to minimize the impact on device performance and battery life. Avoid excessive resource usage that could slow down the device or drain the battery quickly.

**Offline Capability:** Consider providing offline text summarization capabilities to allow users to generate summaries without requiring a constant internet connection. This feature can improve performance and usability, particularly in areas with limited or unreliable network connectivity.

**Caching and Preprocessing:** Implement intelligent caching mechanisms to store previously generated summaries or intermediate processing results. This can help reduce redundant computations and improve overall performance, especially when summarizing similar or frequently accessed texts.

**Multithreading and Asynchronous Processing:** Leverage multithreading and asynchronous processing techniques to improve performance. This allows the app to perform summarization tasks in the background while keeping the user interface responsive and interactive.

**Memory Management:** Efficiently manage memory usage, especially when dealing with large text inputs. Implement strategies such as chunking or streaming to process texts in manageable portions, minimizing memory requirements and improving performance.

**Error Handling and Recovery:** Handle errors and exceptions gracefully to ensure the app's stability and prevent crashes or unexpected behavior. Implement appropriate error-handling mechanisms and recovery strategies to maintain a smooth user experience.

**Performance Monitoring and Optimization:** Continuously monitor the app's performance metrics, such as response time, processing speed, and resource usage. Analyze performance data to identify bottlenecks, optimize algorithms, and improve overall efficiency.

By addressing these performance requirements, we can ensure that our automatic text summarization Android application delivers fast, responsive, and efficient summarization capabilities, providing users with a smooth and satisfactory experience.

* 1. **Safety Requirements:**

Our application uses secure Firebase authentication, and Firebase verification, and has (SUMIT) summary secure API which only responds to requests with secure access tokens, no third party can access our API to secure from unauthorized access from other users or internet data stealing. When developing an automatic text summarization Android app, it's important to consider safety requirements to ensure the privacy, security, and ethical use of the app. Here are some key safety requirements to consider:

**Data Privacy:** Protect the privacy of users' text data by implementing strong data protection measures. Use encryption techniques to secure sensitive data, both during transit and storage. Clearly communicate the app's data privacy policy and obtain user consent for data collection and usage.

**User Authentication and Authorization:** Implement robust user authentication mechanisms to prevent unauthorized access to the app and user data. Consider integrating secure login methods such as two-factor authentication for enhanced security.

**Secure Communication:** Ensure that all communications between the app and external servers or APIs are encrypted using secure protocols such as HTTPS. This prevents eavesdropping and protects user data during transmission.

**Secure Storage**: Store user data, including summaries and user preferences, in a secure manner. Use encryption and access controls to protect data at rest and prevent unauthorized access or data breaches.

**Ethical Use:** Design and implement the app to adhere to ethical guidelines and best practices. Avoid using the app for malicious purposes, such as generating misleading or harmful summaries. Clearly communicate the app's intended use and limitations to users.

**Error Handling:** Implement appropriate error handling mechanisms to ensure the app remains stable and secure. Gracefully handle exceptions, validate user input, and provide clear error messages to enhance user experience and prevent potential security vulnerabilities.

**User Consent and Transparency:** Obtain explicit consent from users for any data collection, processing, or sharing activities. Clearly communicate how the app uses and stores user data, including summaries, and provide users with control over their data through privacy settings or options to delete data.

**Regular Updates and Security Patches:** Maintain the app by regularly releasing updates and security patches to address any identified vulnerabilities or issues. Promptly address security concerns to ensure the app remains safe and secure for users.

**Third-Party Libraries and APIs**: When using third-party libraries or APIs for text processing or summarization, ensure they are reputable, regularly updated, and have a strong track record of security. Keep dependencies up to date to benefit from security fixes and improvements.

**User Support and Reporting:** Provide users with a mechanism to report any issues, vulnerabilities, or concerns related to the app's safety and security. Establish a support system to address user queries, aid, and promptly respond to any security incidents.

By incorporating these safety requirements into the development process, we can enhance the security, privacy, and ethical use of our automatic text summarization Android app, providing users with a safe and trustworthy experience.

* 1. **Security Requirements:**

To prevent unauthorized access from other users or the theft of internet data, our software uses secure Firebase authentication and Firebase verification. It also has a (SUMIT) owned secure API that only replies to requests with secure access tokens.

* 1. **Software Quality Attributes:**

Software quality attributes refer to the desirable characteristics or qualities of a software application. When it comes to an automatic text summarization application, several key quality attributes should be considered to ensure a high-quality and reliable system. Here are some important software quality attributes for an automatic text summarization application:

**Accuracy:** The application should generate accurate summaries that capture the key information and meaning of the input text. The summaries should be relevant, coherent, and free from significant errors or distortions.

**Reliability:** The application should consistently produce reliable and trustworthy summaries. It should perform consistently across different inputs and demonstrate a high level of stability and robustness.

**Performance:** The application should be efficient and provide timely responses. It should generate summaries within acceptable time frames, even when dealing with large volumes of text. Performance requirements, such as response time and throughput, should be met to ensure a satisfactory user experience.

**Usability:** The application should be user-friendly and intuitive, with a well-designed user interface. Users should be able to interact with the app easily, input text, and receive summaries without confusion or unnecessary complexity. Consider incorporating features like clear instructions, error handling, and intuitive navigation.

**Maintainability:** The application should be designed and implemented in a way that facilitates easy maintenance and future enhancements. This includes writing clean, modular, and well-documented code, using best practices, and employing software engineering principles. Clear separation of concerns, code readability, and appropriate documentation contributes to maintainability.

**Scalability:** The application should be designed to handle varying workloads and accommodate increasing demands as the user base or text input size grows. It should be able to scale horizontally or vertically, ensuring that it can handle additional users, concurrent requests, or larger text inputs without significant performance degradation.

**Security:** The application should incorporate security measures to protect user data, prevent unauthorized access, and ensure secure communication. This includes employing encryption techniques, implementing secure authentication and authorization mechanisms, and following best practices for data privacy and protection.

**Testability**: The application should be designed with testability in mind, allowing for effective testing and quality assurance. This includes writing testable code, providing appropriate testing interfaces or APIs, and employing automated testing methodologies to ensure the correctness and reliability of the application.

**Adaptability:** The application should be adaptable to different domains, languages, or specific requirements. It should be flexible enough to handle various types of text inputs and produce meaningful summaries irrespective of the content or context.

**Accessibility:** Consider incorporating accessibility features to make the application usable by a wide range of users, including those with disabilities. This involves adhering to accessibility guidelines, providing appropriate text alternatives, and supporting assistive technologies.

By considering these software quality attributes, you can develop an automatic text summarization application that delivers accurate, reliable, efficient, and user-friendly summarization capabilities, meeting the expectations and requirements of your users.

* 1. **Business Rules:**

It's important to note that these rules can be adjusted and customized based on the specific needs and constraints of each business, as well as the capabilities of the automatic text summarization system being used.

**Length Constraint**: Set a maximum limit for the length of the generated summary. This can be based on the desired output format or platform where the summary will be displayed. For example, if the summary is meant to be displayed on a mobile device, it may need to be shorter to fit the limited screen space.

**Content Relevance:** Ensure that the summary focuses on the most important and relevant information from the original text. This can be achieved by using algorithms that analyze the importance of sentences or keywords in the text and prioritize them for inclusion in the summary. This can involve techniques such as natural language processing and syntactic analysis to ensure that the summary reads well and conveys the intended meaning accurately.

**Context Preservation**: Preserve the context of the original text as much as possible. While summaries aim to condense information, it is important to ensure that the summary retains the core message and context of the original text. This can be achieved by including key details, references, or connections that provide a holistic understanding of the subject matter. This can help users quickly identify key elements in the summarized content and provide additional context or reference points.

**Plagiarism and Copyright:** Implement mechanisms to avoid plagiarism and respect copyright laws. Automatic text summarization should not infringe upon the intellectual property rights of the original content creators. This can be accomplished by using techniques such as paraphrasing, proper citation, or obtaining appropriate licenses for the source content. Businesses may have different requirements or preferences for summarization based on their specific domains or target audiences. Allowing users to customize the summary generation process can enhance the usefulness and relevance of generated summaries.

**Performance and Efficiency:** Optimize the summarization system for speed and scalability. Depending on the scale of text data to be processed, the system should be designed to handle large volumes of data efficiently and provide summaries within acceptable timeframes.

**Evaluation and Feedback:** Establish mechanisms to evaluate the quality and effectiveness of the generated summaries. Feedback from users or automated evaluation metrics can be used to continuously improve the summarization algorithms and ensure that the summaries meet the desired standards.

* 1. **Interoperability**

After passing through a few security checks, our app interacts with the database and API in roughly a second. This allows several users to interact simultaneously.

* 1. **Extensibility**

The extensibility of automatic text summarization refers to the ability to adapt and extend the existing methods and systems to cater to different domains, languages, or specific requirements. Extensibility is an important aspect of automatic text summarization as it allows the technology to be applied effectively in diverse contexts. Here are a few key points regarding the extensibility of automatic text summarization:

**1. Domain Adaptation:** Automatic text summarization systems need to be adaptable to different domains, such as news articles, scientific papers, legal documents, or social media posts. The system should be able to handle the specific language, jargon, and characteristics of the given domain to produce accurate and relevant summaries.

**2.** **Multilingual Summarization**: Extensibility involves the capability of summarizing texts in multiple languages. Language-specific challenges, such as word order, grammar, and semantic nuances, need to be considered to generate high-quality summaries in different languages.

**3. Customization for Specific Needs:** Different users or applications may have specific requirements for summaries. The extensibility of text summarization systems enables customization based on factors such as length restrictions, domain-specific keywords, or preferred summarization styles (extractive vs. abstractive).

**4. Incorporating New Data Sources:** Automatic text summarization should be flexible enough to integrate and summarize content from various sources, including online articles, blogs, social media feeds, or real-time streams. The system should handle different formats and adapt to the evolving nature of data sources.

**5. Fine-tuning and Transfer Learning:** Extensibility involves leveraging pre-trained models and transfer learning techniques. Fine-tuning existing models with domain-specific data or adapting models from related tasks can enhance the performance and applicability of automatic text summarization to new contexts.

**6. Open APIs and Toolkits**: Extensible summarization frameworks provide APIs and toolkits that allow developers to integrate and customize the summarization capabilities within their applications or workflows. These interfaces enable the extension of summarization functionalities to meet specific application requirements.

**7. Research and Development:** The extensibility of automatic text summarization relies on continuous research and development efforts. Advancements in machine learning, natural language processing, and text summarization techniques contribute to the extensibility of the technology.

By focusing on extensibility, automatic text summarization systems can be adapted and extended to cater to a wide range of applications, languages, and domains, making them more versatile and useful in various contexts.

* 1. **Maintainability**

our application should be accessible 99.99% of the time. It is recommended to carry out any software updates, patches, and fixes without terminating the application. To address disasters, a disaster recovery ecosystem should be in place.

* 1. **Portability**

Our application is very portable because it can be deployed on any Android device with version 7 or higher and any IOS device with version 9 or higher. It can be deployed via the application's APK or from the Google Play Store or App Store at any time. The application is versatile and compatible with a variety of Android and IOS devices.

* 1. **Reusability:**

Our app has a lot of reusable widgets and components that are simple to integrate into many other applications. For example, the authentication process can be used in another Flutter app, just like in SUMIT, which offers full security through Firebase. Components that read APIS and convert them into lists can also be reused. Over 80% of our app is reusable and can benefit many other applications, both with and without some modifications.

* 1. **Installation:**

### Android:

Installation of Android is possible through the Play Store or the app of our app.

### IOS:

Installation can only be done through App Store.

### Other Requirements

* 1. **On-line User Documentation and Help System Requirements**

In an online user documentation and help system for a text summarization app, several requirements should be considered to ensure effective support for users. Here are some important requirements:

**1: Clear and comprehensive documentation**: The user documentation should provide clear instructions on how to use the text summarization app. It should cover all features and functionalities, explaining them in a user-friendly manner. The documentation should be well-structured and easy to navigate, allowing users to quickly find the information they need.

**2: Search functionality**: A search feature within the documentation is crucial for users to find specific information or answers to their questions. It should be able to search for keywords and provide relevant results, saving users time and effort.

**3: Interactive tutorials**: Including interactive tutorials or walkthroughs can be beneficial for new users. These tutorials should guide users through the app's features step-by-step, allowing them to practice and understand the summarization process effectively.

**4: Frequently Asked Questions (FAQs):** A dedicated section for frequently asked questions can help address common user queries and issues. Compile a list of relevant FAQs and provide clear and concise answers to assist users in troubleshooting or understanding the app better.

**5: Contextual help and tooltips:** Implementing contextual help and tooltips within the app can provide users with on-the-spot guidance. When users hover over or click on specific elements or options, relevant information, and explanations should be displayed, aiding users in understanding the app's functionality in real-time.

**6: Visual aids and examples**: Incorporate visual aids, such as screenshots or videos, to demonstrate specific tasks or processes within the app. Examples of summarized texts and before-and-after comparisons can help users grasp the app's capabilities and potential output.

**7: User feedback and support channels:** Provide a feedback mechanism within the documentation or app interface, allowing users to report issues or suggest improvements. Additionally, include contact information or links to customer support channels, such as email, live chat, or community forums, where users can seek assistance or engage with other users.

**8: Mobile-friendly and responsive design:** Ensure that the online documentation and help system is mobile-friendly and responsive, adapting to different screen sizes and devices. This is important as users may access the documentation from various platforms, including smartphones and tablets.

**9: By considering these requirements:** we can create an effective online user documentation and help system for a text summarization app, supporting users in understanding and utilizing the app's features to their fullest potential.

* 1. **Purchased Requirements**

There might be some charges if the user wants to make his own customized plan.

* 1. **Licensing Requirements**

It shall be taken as per requirement.

* 1. **Legal, copyright, and Other Notices**

It shall be taken as per requirement.

* 1. **Applicable Standards**

It shall be taken as per requirement.

* 1. **Reports (Feedback, Invoice, User, Usage, Balance Sheet, Executive Summary, etc.)**

Automatic text summarization is a useful technique for condensing large volumes of information into concise summaries. While it is commonly used for news articles, research papers, and other textual content, it can also be applied to various types of reports. Here are some examples of reports that can benefit from automatic text summarization:

**Feedback Reports:** These reports contain customer, user, or employee feedback and reviews. Automatic text summarization can extract key insights and sentiments from these reports, concisely summarizing the overall feedback received.

**User Reports:** User reports typically include data and analytics related to user behavior, engagement, and interactions with a product or service. Automatic text summarization can extract crucial trends, patterns, and user preferences, offering a concise overview of user activities and insights.

**Usage Reports:** These reports focus on tracking the usage or consumption of a particular resource or service. Summarizing usage reports can summarize resource utilization, peak usage periods, and other significant usage patterns.

**Executive Summary:** An executive summary provides an overview of a longer report, highlighting the main points, conclusions, and recommendations. Automatic summarization can help generate a concise summary of the key findings, enabling busy executives to quickly grasp the main insights without going through the entire report.

When applying automatic text summarization to these reports, it's essential to consider the specific requirements and context of each report type. Different algorithms and techniques can be utilized, such as extractive summarization, which selects and condenses important sentences, or abstractive summarization, which generates a summary using natural language generation techniques. The choice depends on the available data, desired level of detail, and the specific needs of the end-users.

1. **References:**
2. "Text Summarization Techniques: A Brief Survey" by Inderjeet Kaur and Preeti Saini - This research paper provides an overview of different text summarization techniques, including extractive and abstractive approaches. You can search for it using the title and authors. [link] <https://query.data.world/s/lpim7d7ewx33ykadthda623eminani>
3. "A Neural Attention Model for Abstractive Sentence Summarization" by Alexander M. Rush, Sumit Chopra, and Jason Weston - This paper introduces an abstractive text summarization model based on neural attention mechanisms. Searching for the title and authors will lead you to the paper. [link] <https://www.researchgate.net/publication/281487270_A_Neural_Attention_Model_for_Abstractive_Sentence_Summarization>
4. "TextRank: Bringing Order into Texts" by Rada Mihalcea and Paul Tarau - This paper presents the TextRank algorithm, which is a graph-based approach for extractive text summarization. Search for the title and authors to find the paper. [link] <https://www.researchgate.net/publication/286694529_Application_of_TextRank_Algorithm_for_Credibility_Assessment>
5. "Deep Reinforcement Learning for Sequence-to-Sequence Models of Text Summarization" by Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky - This research paper explores the application of deep reinforcement learning for abstractive text summarization. Search for the title and authors to access the paper. [link] <https://arxiv.org/pdf/1812.02303>
6. "Attention Is All You Need" by Vaswani et al. - This influential paper introduces the Transformer model, which has been widely used in natural language processing tasks, including text summarization. Searching for the title and authors will lead you to the paper. [link] <https://arxiv.org/abs/1706.03762>