**SUMMARIZE AN ARTICLE USING ADAPTIVE TEXT**

**SUMMARIZATION API WITH IBM CLOUD**

**Index**

1. Introduction

a. Overview

b. Purpose

2. Literature Survey

a. Existing problem

b. Proposed solution

3. Theoretical Analysis

a. Block diagram

b. Hardware / Software designing

4. Experimental Investigations

5. Flowchart

6. Result

7. Advantages & Disadvantages

8. Applications

9. Conclusion

10. Future Scope

11. Bibliography

12. Appendix

a. Source code

b. UI output Screenshot

**1.Introduction**

**a.Overview:**

Text summarization is the process of shortening long pieces of text while preserving key information content and overall meaning, to create a subset (a summary) that represents the most important or relevant information within the Text.

**b.Purpose:**

Text summarization methods are increasingly needed in different fields of knowledge. In the scientific literature, generic extractive multi-document text summarization can be formulated as an optimization problem which involves several criteria. Only two criteria have been considered simultaneously, i.e., content coverage and redundancy reduction, whereas the other ones, relevance and coherence have been considered separately. Therefore, there is a lack of studies comparing the performance of different criteria. For this reason, a comparative study of the different criteria suitable for generic extractive text summarization is performed using IBM Cloud.

**2. Literature Survey**

**a.Existing Problem:**

Summarizing the content,just by reading a document is hard and timing consuming process. As technology is improving nobody wants to dirty their hands.There exists a need for a concise and meaningful summary of text from multiple text resources such as books, news articles, blog posts, research papers, emails, and tweets.

**b.Proposed Solution:**

To overcome the drawback that exists , SUMMARIZATION OF ARTICLE is proposed.As the amount of public content increases, so does the difficulty of reading it in a limited span of time. Moreover, digesting all of this content in a meaningful fashion requires time and effort. An efficient way of skimming through the content is to read summaries, if available. This application helps you in getting the summary of long articles which can save your time. The article can be directly taken from the URL/website or give your article and get a summary of it.

* **Extraction-based summarization:** Here, content is extracted

from the original data, but the extracted content is not modified in any way.

**Example:**

**Before Summarization**

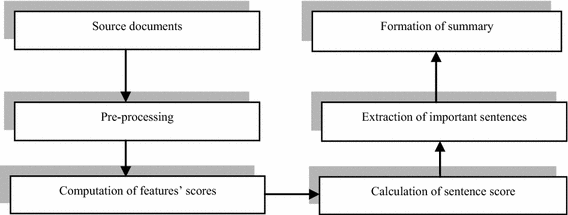
John and Joseph took a taxi to attend the night party in the city. While in the party, John collapsed and was rushed to the hospital.

**After Summarization**

John and Joseph attend party. John rushed hospital.

**3.Theoretical Analysis**

**a.Block Diagram:**



**b.Software Designing:**

**. Skills:**Python,Flask Integration,IBM Cloud,Python-Flask,API

To accomplish the system,we need to go through the following

1. Installation of Pre-requisites

* Installation of Anaconda IDE / Anaconda Navigator.
* Installation of Python packages.

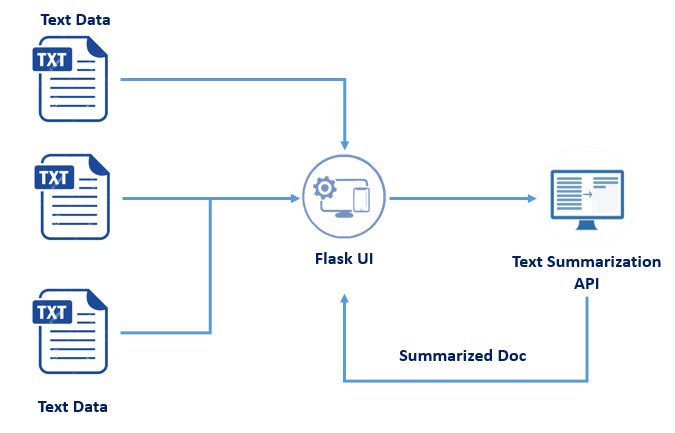
1. RapidAPI Account Creation.
2. Subscription of Application Oriented API.
3. Building a Flask Application.

* Importing of Libraries and routing the HTML pages
* Running of flask Application.

5. Deploy the flask application into IBM Cloud .

6. Accepting input from the user and get the results back in UI.

**Software design Architecture:**



**4.Experimental Investigations:**

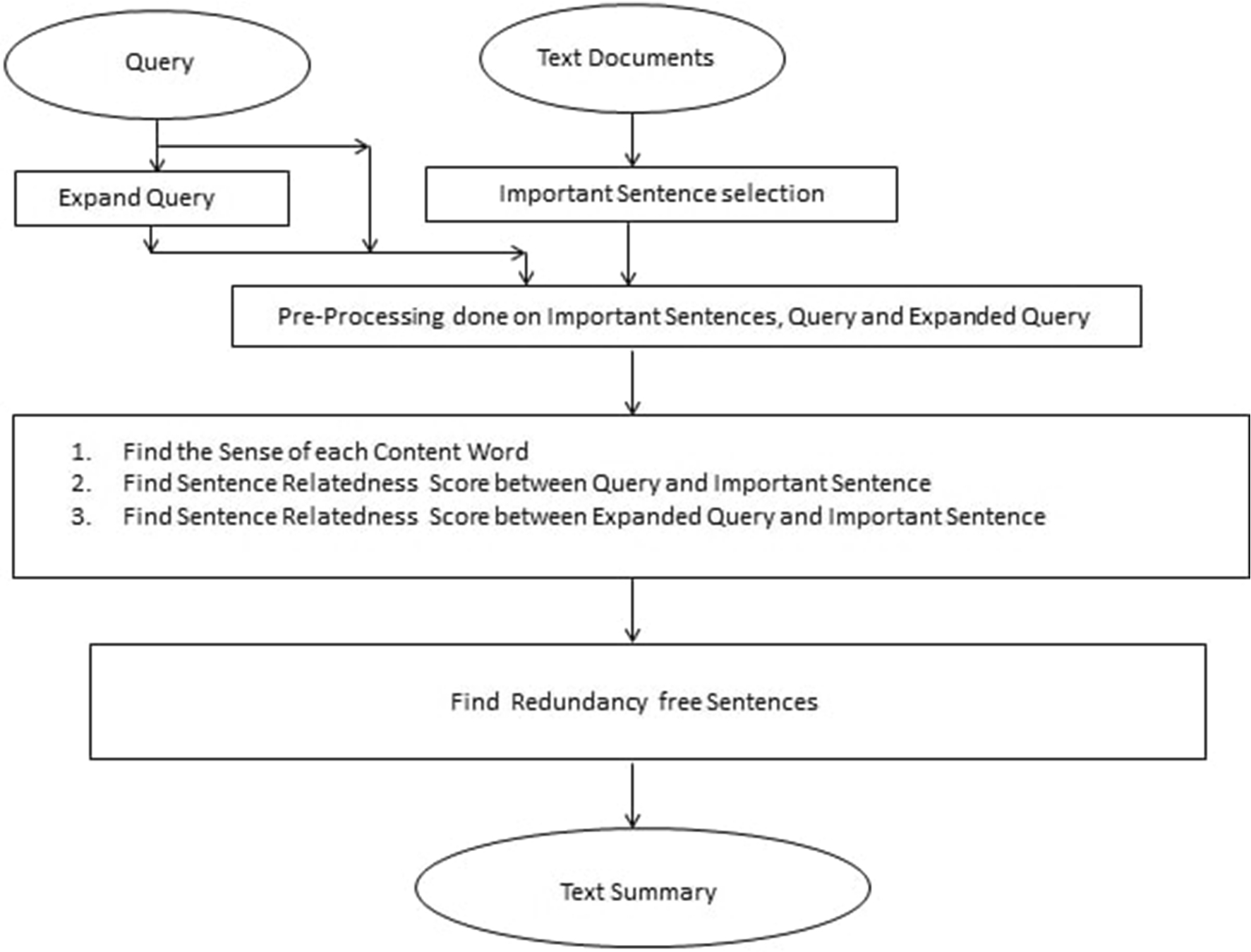
Many articles stated the need and use of text summariser.It can be developed even with tensorflow,NLP algorithm and so on.Here in this,flask API and IBM cloud is used.

**Reference :**

https://en.wikipedia.org/wiki/Automatic\_summarization#Applications

**5.Flowchart**

%3CmxGraphModel%3E%3Croot%3E%3CmxCell%20id%3D%220%22%2F%3E%3CmxCell%20id%3D%221%22%20parent%3D%220%22%2F%3E%3CmxCell%20id%3D%222%22%20value%3D%22Source%20Text%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22280%22%20y%3D%2220%22%20width%3D%22160%22%20height%3D%2240%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%223%22%20value%3D%22Data%20processing%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22280%22%20y%3D%22100%22%20width%3D%22160%22%20height%3D%2240%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%224%22%20value%3D%22Computaion%20of%20feature%20scores%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22280%22%20y%3D%22180%22%20width%3D%22160%22%20height%3D%2240%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%225%22%20value%3D%22Formation%20of%20Summary%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22560%22%20y%3D%2220%22%20width%3D%22160%22%20height%3D%2240%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%226%22%20value%3D%22Extraction%20of%20Important%20Sectences%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22560%22%20y%3D%22100%22%20width%3D%22160%22%20height%3D%2240%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%227%22%20value%3D%22Calculation%20of%20sentence%20score%22%20style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%3Bhtml%3D1%3B%22%20vertex%3D%221%22%20parent%3D%221%22%3E%3CmxGeometry%20x%3D%22560%22%20y%3D%22180%22%20width%3D%22160%22%20height%3D%2240%22%20as%3D%22geometry%22%2F%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%228%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D1%3BexitDx%3D0%3BexitDy%3D0%3B%22%20edge%3D%221%22%20source%3D%222%22%20target%3D%223%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22280%22%20y%3D%22240%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22330%22%20y%3D%22190%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%229%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D1%3BexitDx%3D0%3BexitDy%3D0%3BentryX%3D0.5%3BentryY%3D0%3BentryDx%3D0%3BentryDy%3D0%3B%22%20edge%3D%221%22%20source%3D%223%22%20target%3D%224%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22280%22%20y%3D%22280%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22330%22%20y%3D%22230%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2210%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3BexitX%3D1%3BexitY%3D0.5%3BexitDx%3D0%3BexitDy%3D0%3B%22%20edge%3D%221%22%20source%3D%224%22%20target%3D%227%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22280%22%20y%3D%22280%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22330%22%20y%3D%22230%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2211%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3BentryX%3D0.5%3BentryY%3D1%3BentryDx%3D0%3BentryDy%3D0%3BexitX%3D0.5%3BexitY%3D0%3BexitDx%3D0%3BexitDy%3D0%3B%22%20edge%3D%221%22%20source%3D%227%22%20target%3D%226%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22280%22%20y%3D%22280%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22330%22%20y%3D%22230%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3CmxCell%20id%3D%2212%22%20value%3D%22%22%20style%3D%22endArrow%3Dclassic%3Bhtml%3D1%3BexitX%3D0.5%3BexitY%3D0%3BexitDx%3D0%3BexitDy%3D0%3B%22%20edge%3D%221%22%20source%3D%226%22%20target%3D%225%22%20parent%3D%221%22%3E%3CmxGeometry%20width%3D%2250%22%20height%3D%2250%22%20relative%3D%221%22%20as%3D%22geometry%22%3E%3CmxPoint%20x%3D%22280%22%20y%3D%22280%22%20as%3D%22sourcePoint%22%2F%3E%3CmxPoint%20x%3D%22330%22%20y%3D%22230%22%20as%3D%22targetPoint%22%2F%3E%3C%2FmxGeometry%3E%3C%2FmxCell%3E%3C%2Froot%3E%3C%2FmxGraphModel%3444



**6.Result**

Using flask framework and IBM Cloud,the article summarizer was developed.User has to give input text and the summarized data will be processed in the backend result and produce the simplified data.

**Link:**[GitHub - smartinternz02/SI-GuidedProject-3316-1624017310](https://github.com/smartinternz02/SI-GuidedProject-3316-1624017310)

**7.Advantages**

* Summaries reduce reading time.
* When researching documents, summaries make the selection process easier.
* Automatic summarization improves the effectiveness of indexing.
* Automatic summarization algorithms are less biased than human summarizers.

**Disadvantages**

* May not provide periodic dates after summarizing.

### **8.Applications**

Text Summarizations finds a wide variety of applications in creation of headlines, synopses, reviews, book, movie and pla summaries, resumes, and so on.

**9.Conclusion**

This article explains simple solution to summarizing data.Thus this system helps to overcome the drawback of manpower to summarize and time consumption.

**10.Future Scope**

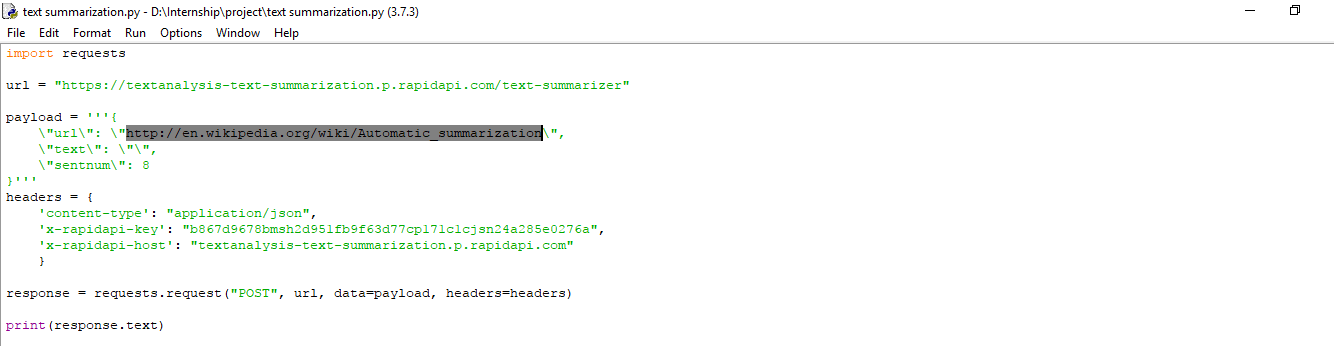
SUMMARIZATION OF ARTICLE provides efficient solution for summarizing data with no time .It is one of the necessity for removing redundancy and increase the efficient data.So,it will definitely be a useful work for summarize data.

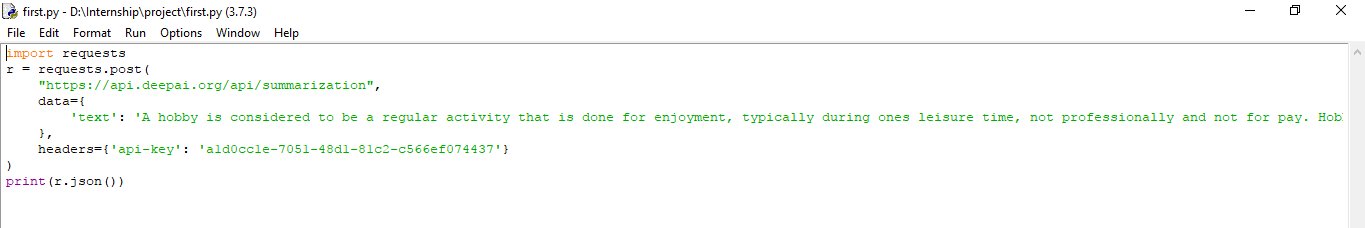
**11.Bibliography**

* IJARCCE ISSN International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 3, March 2016 Copyright to IJARCCE DOI 10.17148/IJARCCE.2016.5340 154 A Review Paper on Text Summarization[IJARCCE 40.pdf](https://www.ijarcce.com/upload/2016/march-16/IJARCCE%2040.pdf)
* Article on Automatic Summarization.[Automatic summarization - Wikipedia](https://en.wikipedia.org/wiki/Automatic_summarization)

**12.Appendix**

**a.Source code**





**b.UI output Screenshot**

