iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VI **Month of publication:** June 2022

DOI: https://doi.org/10.22214/ijraset.2022.44508

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

An Intelligent TLDR Software for Summarization

Krishna Vamsi Rokkam[1], Sharanya Akkenapally[2], Mahima Chowdary Maddineni[3]
[1, 2, 3]*Department of Information Technology, Sreenidhi Institute of Science and Technology*

Abstract: ***The amount of textual data available from diverse resources is increasing dramatically in the substantial data age. This textual volume has a wealth of information and expertise that must be skilfully summarised in order to be useful. Because billions of articles are published every day, it takes a long time to look through and keep up with all of the information available. Much of this text material has to be reduced to shorter, focused summaries that capture the most important aspects, both so we can explore it more efficiently and to ensure that the bigger papers include the information we need. Because manual text summarising is a time-consuming and typically difficult activity, automating it is expanding in popularity and thus provides an ideal impetus for academic study. The growing availability of documents has necessitated much study in the field of natural language processing (NLP) for automatic text summarization. "Is there any software that can assist us digest the facts more efficiently and in less time?" is the genuine question. As a result, the major goal of the summarization system is to extract the most important information from the data and deliver it to the consumers. In NLP, summarization is the act of condensing text information in huge texts to make it easier to understand and consume. We suggest a solution by developing a text summary programme that uses Natural Language Processing and accepts an input (plain text or text scrapped from a website). The output is the outlined text. Natural language processing, along with machine learning, makes it easier to condense large quantities of information into a coherent and fluent summary that only incorporates the article's most important points.***
Keywords: Text, Summarization, Natural Language processing, TLDR Software, Text pre-processing

I. INTRODUCTION

There is a huge quantity of written material, and each and every day, that quantity is simply going to increase further. Imagine the internet, which is made up of different web pages, items of news, status updates, blogs, and a great deal more. Because the data are not organised in any particular way, the only way to traverse them is to perform a search and then scan through the results. There is a significant requirement for reducing a significant portion of this text data to shorter, more focused summaries that capture the essential details. This is necessary not only to enable us to navigate it with greater efficiency but also to determine whether or not the more extensive documents contain the data that we are looking for.

.

II. RELATED WORK

Among various types of data generated, texts are the most general over the network. The information overload problem worsens as the quantity of data keeps increasing rapidly. Text summarization contracts input text into a concise and coherent summary conveying the critical information in the original text. Text summarization approaches can be of two types: extractive summarization and the other one is abstractive summarization. The extractive summarization generates the summary from the sentences in the input, and abstractive summarization creates the generalization of the input using different words. Textual data in digital documents quickly adds up to massive amounts of information. As in [3], the majority of this enormous amount of unstructured data is unconstrained and hasn't been arranged into typical databases. As a result, processing documents is a purely administrative task mostly as a result of a lack of norms. As a result, there is a requirement for a system to reduce the text's size, give it some structure, and process the document to make it more precise. It is understandable to the user. This project aims to address all the above aspects of the problem. With extractive text summarization, the world's expanding amount of textual information.

There are now automatic text summarizers available, each of which uses a unique methodology. The extraction of statistical information based on the frequency and distribution of words is a common strategy. Selecting phrases with the highest scores from those obtained is a common next step. Another method uses this information in conjunction with the positioning of sentences inside the paragraph, cue words, and title and heading terms that are included in the paragraph. The method used in this project uses a Text Rank technique, which analyses a passage based on a number of important characteristics to evaluate whether or not a certain sentence ought to be included in the summary. If one were to tokenize the text into sentences and paragraphs and then analyse every sentence individually, one could be able to construct a thorough summary of the content.

III.MATERIAL AND METHOD

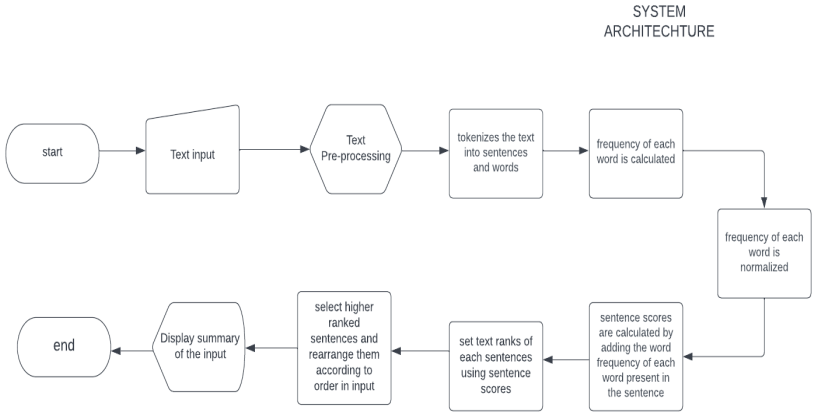


Fig.1 System flow diagram

Natural language processing (NLP) targets on making algorithms to understand natural human language. Natural Language Toolkit, or NLTK, is a Python package that is used for NLP. [2] Python has numerous packages that can be used to reuse code. It has transparent semantics and syntax, making it an excellent choice for natural language processing. It's also straightforward to use and has excellent support for integrating with other tools and languages. The most significant advantage of Python for natural language processing is that it provides developers with a large number of libraries that handle a variety of NLP-related tasks such as topic modelling, document categorization, sentiment analysis, and so on. It is undeniably tough to create software that can deal with natural language. Python's broad toolset, on the other hand, allows developers to create fantastic tools.

TextRank is an algorithm that is frequently used in keyword extraction and text summarising. It is based on the PageRank algorithm. In order to determine which sentences in a body of text are the most pertinent, a graph is constructed. [4] The graph's vertices represent each sentence in the document, and the transitions between sentences are determined by the amount of content overlap between the sentences; specifically, this is done by counting the number of words that both sentences share.

The PageRank algorithm as in [7], which determines which sentences are the most essential based on their weight in the network of sentences, is fed the sentences that make up this network. Now, when extracting a summary of the text, we can take only the most important sentences.

The textrank algorithm creates a word network in order to locate meaningful keywords in the content. This network is constructed by observing which words are connected to one another. A connection is made between two words if they come one after the other in the text; this connection is given more significance if the two words in question appear in close proximity to one another more frequently. The PageRank algorithm is implemented on top of the generated network in order to determine the significance of each individual word. Only the first one third of all of these words are retained since they are regarded to be significant. Following this step, a keywords table is created by grouping together the pertinent terms in the text in the event that they are found immediately following one another.

IV. RESULT ANALYSIS

The acronym ROUGE[6] stands for "recall-oriented understudy for pointing evaluation." In its most basic form, it is a set of measurements that can be used to evaluate automatic text summarization and machine translation. It accomplishes this by contrasting a translation or summary that was generated automatically with a collection of reference summaries (typically human-produced). We can really compute the precision and recall by using the overlap, which will allow us to obtain a good quantitative result. To put it another way, recall (in the context of ROUGE) refers to the portion of the reference summary that is being recovered or captured by the system summary. If we ignore everything but the words themselves, we can arrive at the following conclusion:

Recall = number of overlapping words / Total words in reference summary

This seems like it would make an excellent text summarising system. However, it does not present the opposing viewpoint to the argument.[1] A system summary, which is generated by a machine, can be very lengthy because it includes all of the words that are in the reference summary. However, it's possible that many of the words in the system overview aren't necessary, which makes the summary wordier than it needs to be.
Here is where the need for precision becomes apparent. In terms of accuracy, what you are ultimately measuring is the proportion of the system summary that was actually necessary or relevant to the purpose of the analysis. Precision can be measured as:

Precision = Number of overlapping word / Total words in system summary

When attempting to develop summaries that are condensed in nature, accuracy becomes an extremely important factor to take into consideration. Because of this, it is always recommended to first compute the precision and then the recall before reporting the F-measure. [5] If your summaries are in some manner compelled to be succinct as a result of some constraints, then you might want to think about using just the recall instead of the precision because accuracy is less of an issue in this circumstance.

Three different granularity levels [7] can be used to compare the system summaries to reference summaries: ROUGE-S, ROUGE-L, or ROUGE-N

- ROUGE-N — measures unigrams, bigrams, trigrams, and orders of magnitude higher n-grams that share a same root
- Rouge-L measures the longest matching sequence of words using LCS. Using LCS is advantageous because it does not require consecutive matches, but rather in-sequence matches that reflect sentence-level word order. " You don't need to specify a specific n-gram length because it automatically includes the longest in-sequence common n-gram.

Doesn't matter in which order the words are in, as long as there are no gaps between them. Skip-gram concurrence is another term for this. Word pairs with a maximum of two spaces in between them can have their overlap measured using skip-bigram, for example. Let's say you want to skip-bigram "cat in the hat," the skip-bigrams for that sentence would be something like "cat hat" for example.
Using ROUGE-1 as an example, it refers to the number of unigrams that overlap between the system summary and the reference summary. Bigram overlap between the system and reference summaries is referred to as ROUGE-2.

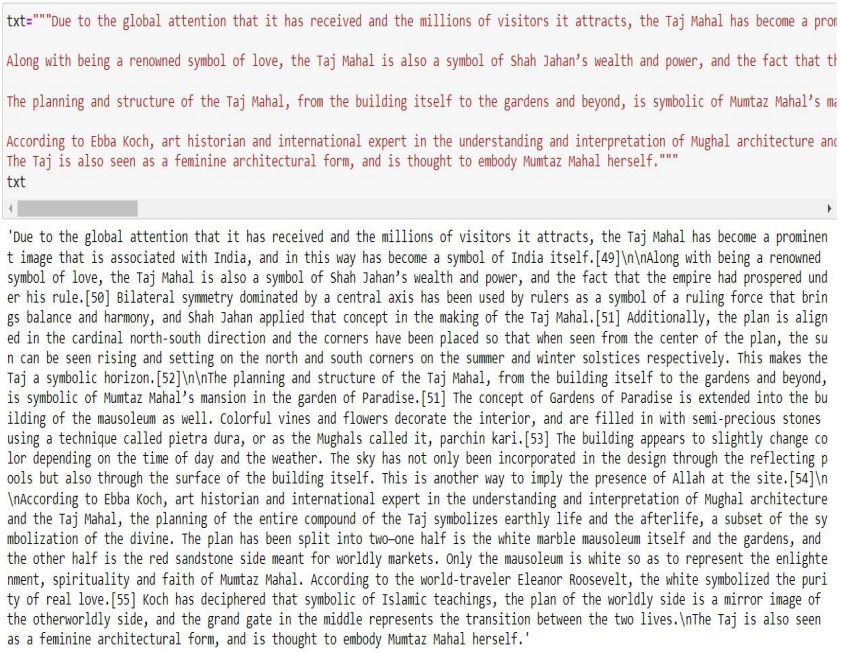


Fig.2 Taking an example for result analysis

```
res=summarizer(txt)
res
```

```
'Due to the global attention that it has received and the millions of visitors it attracts, the Taj Mahal has become a prominen
t image that is associated with India, and in this way has become a symbol of India itself.[49]\n\nAlong with being a renowned
symbol of love, the Taj Mahal is also a symbol of Shah Jahan's wealth and power, and the fact that the empire had prospered und
er his rule.[50] Bilateral symmetry dominated by a central axis has been used by rulers as a symbol of a ruling force that brin
gs balance and harmony, and Shah Jahan applied that concept in the making of the Taj Mahal.[51] This makes the Taj a symbolic h
orizon.[52]\n\nThe planning and structure of the Taj Mahal, from the building itself to the gardens and beyond, is symbolic of
Mumtaz Mahal's mansion in the garden of Paradise.[51]'
```

```
from rouge import Rouge
r = Rouge()
r.get_scores(res, txt)
```

```
[{'rouge-1': {'r': 0.3739130434782609, 'p': 1.0, 'f': 0.5443037935066496},
  'rouge-2': {'r': 0.33163265306122447,
   'p': 0.8923664122137404,
   'f': 0.49713192741157247},
  'rouge-l': {'r': 0.3739130434782609, 'p': 1.0, 'f': 0.5443037935066496}}]
```

Fig.3 Finding out the performance score of model

V. CONCLUSIONS

Summarizing methods are all extractive, although the community is rapidly moving toward abstract summarization. An abstractive summarising can be created by using phrase compression and textual entailment techniques, even though a complete abstractive summarization would require a better grasp of natural language. Textual entailment aids in the detection of condensed versions of longer texts that convey the same message. We can construct shorter and more concise summaries using textual entailment.

VI. FUTURE SCOPE

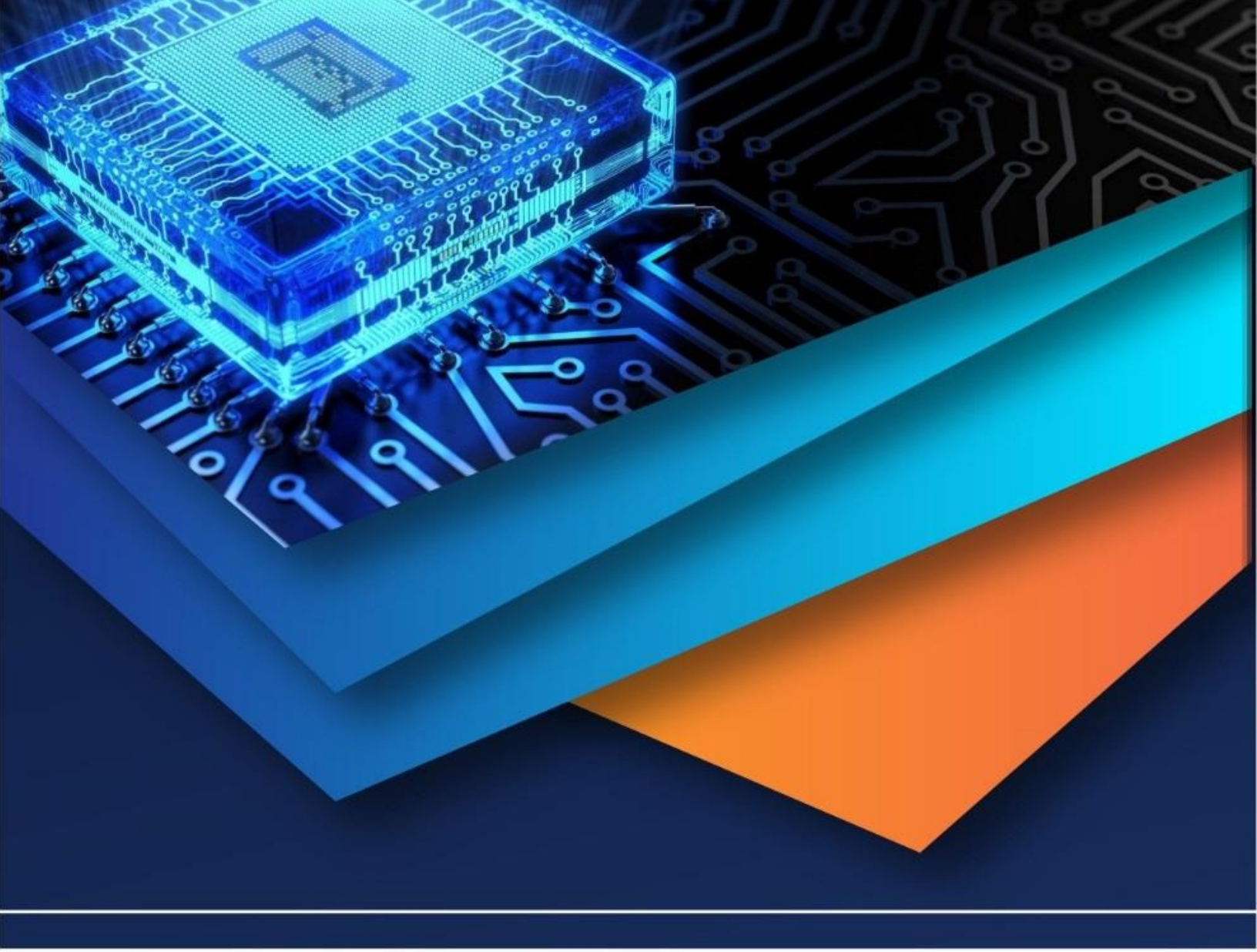
The difficulty occurs in producing accurate summaries with suitable semantics, which include all aspects of text summarising, image summarization, article summaries, and multiple document summaries. We can also implement additional translation and text-to-speech capabilities to the model, making it more robust, adaptable and user-friendly. Summarization procedures must create a persuasive summary in a short amount of time, with little repetition and grammatically sound phrases.

VII. ACKNOWLEDGMENT

We would like to express our gratitude to our college management and faculty for guiding us in our research, especially the department of Information Technology. Our sincere thanks are also extended to our family who have supported throughout the process. Finally, the information provided in this paper is genuine and true to my knowledge.

REFERENCES

[1] https://medium.com/analytics-vidhya/simple-text-summarization-using-nltk eedc36ebaaf8#:~:text=Text%20summarization%20is%20the%20process,relevant%20information%20within%20the%20Text.
[2] https://machinelearningmastery.com/gentle-introduction-text-summarization/
[3] https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25
[4] Saranyamol C S, Sindhu L, "A Survey on Automatic Text Summarization", International Journal of Computer Science and Information Technologies, 2014, Vol. 5 Issue 6.
[5] 2. Reeve Lawrence H., Han Hyoil, Nagori Saya V., Yang Jonathan C., Schwimmer Tamara A., Brooks Ari D., "Concept Frequency Distribution in Biomedical Text Summarization", ACM 15th Conference on Information and Knowledge Management (CIKM), Arlington, VA, USA,2006.
[6] Blog.mashape.com/list-of-30-summarizer-apis-libraries-and-software.
[7] 4. Khan Atif, Salim Naomie, "A review on abstractive summarization Methods", Journal of Theoretical and Applied Information Technology, 2014, Vol. 59 No. 1.
[8] 5. Suneetha Manne, Zaheer Parvez Shaik Mohd. , Dr. S. Sameen Fatima, "Extraction Based Automatic Text Summarization System with HMM Tagger", Proceedings of the International Conference on Information Systems Design and Intelligent Applications, 2012, Vol. 132, P.P 421-428.
[9] 6. Gupta Vishal, "A Survey of Text Summarizers for Indian Languages and Comparison of their Performance", Journal of Emerging Technologies In Web Intelligence, 2013, Vol. 5, No. 4.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY