

# Abstraction-Based Text Summarization using Python Libraries

**Byregowda B K<sup>1</sup>, Dr. Sheshappa S N<sup>2</sup>, Rahul Jairam<sup>3</sup>, Prajwal M D<sup>4</sup>**  
Assistant Professor, Department of Information Science and Engineering<sup>1</sup>  
Associate Professor, Department of Information Science and Engineering<sup>2</sup>  
Students, Department of Information Science and Engineering<sup>3,4</sup>  
Sir M Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India

**Abstract:** *As there is an increase in the usage of digital applications, the availability of data generated has increased to a tremendous scale. Data is an important component in almost every domain where research and analysis are required to solve the problems. It is available in a structured or unstructured format. Therefore, in order to get corresponding data as per the application's purpose, easily and quickly from different sources of data on the internet, an online content summarizer is desired. Summarizers makes it easier for users to understand the content without reading it completely. Abstractive Text Summarizer helps in defining the content by considering the important words and helps in creating summaries that are in a human-readable format. The main aim is to make summaries in such a way that it should not lose its context. Various Neural Network models are employed along with other machine translation models to bring about a concise summary generation.*

**Keywords:** Neural Network

## I. INTRODUCTION

With the rapid growth of the Internet, people are overwhelmed by the tremendous amount of online information and documents. As the information communication technologies are expanding at a great speed large number of electronic documents are easily available online and user facing a difficulty to find related information; As a result, users get so exhausted reading large amount of text that they may skip reading many important and relevant documents so these concerns have sparked interest in the development of Text summarization. Text summarization is the task of creating a document from one or more textual sources that is smaller in size but retains some or most of the information contained in the original sources. What information and which other characteristics of the source documents are kept depends on the intended use of the summary. It is a tedious task to gather all the data and give a summarized form. So, text summarizers came in hand that condense the document to a shorter version providing a clear and concise summary of the dialogue. A summary is beneficial as it helps in recouping long text files and thus saving time as well. This as a whole decodes the issues, challenges and how an abstractive method of summarization can be used for dialogue systems. Natural Language Processing (NLP) is all about interpreting human language from one structure to another. In NLP, Summarization is one of the research work which focuses on providing relevant summary using various Natural Language Processing .

## II. OBJECTIVE

The main objective of a text summarization system is to identify the most important information from the given text and present it to the end users. In this paper, Wikipedia articles are given as input to system and Abstractive text summarization is presented by identifying text features and scoring the sentences accordingly. The text is first pre-processed to tokenize the sentences and perform stemming operations. We then score the sentences using the different text features. Two novel approaches implemented are using the citations present in the text and identifying synonyms. These features along with the traditional methods are used to score the sentences. The scores are used to classify the sentence to be in the summary text not with the help of a neural network. The user can provide what percentage of the original text should be in the summary. It is found that scoring the sentences based on citations gives the best results.

### III. EXISTING SYSTEM

Taeho Jo et.al (2017) proposed a method that uses the feature vector of certain features and obtains the correlation between the vectors. We propose a version of KNN (K NearestNeighbor) where the similarity between feature vectors is computed considering the similarity among attributes or features as well as one among values. The task of text summarization is viewed as the binary classification task where each paragraph or sentence is classified into the essence or non- essence, and in previous works, improved results are obtained by the proposed version in the text classification and clustering. In this research, we define the similarity which considers both attributes and attribute values, modifies the KNN into the version based on the similarity, and use the modified version as the approach to the text summarization task. As the benefits of this research, we may expect a more compact representation of data items and better performance. Therefore, the goal of this research is to implement the text summarization algorithm which represents data items more compactly and provides more reliability. The proposed approach should be applied and validated in the specialized domains: engineering, medicine, science, and law, and it should be customized to the suitable version, to cut down the computation time. We develop and combine various schemes of computing the similarities among features. By adopting the proposed approach, we will develop the text summarization system as a real version. The total observed percentile is 65 percentage.

### IV. PROBLEM IDENTIFICATION

An Investigating Officer may sometimes be required to refer to online news articles to obtain further information about a case beyond what is already known through on-ground sources. Due to the proliferation of news websites on the internet, it is not uncommon for a simple search on a topic or suspect of interest to return thousands, and even lakhs, of relevant news articles. It would take an Investigating Officer hours and hours of manual effort to go through these news articles, understand them and assimilate key findings. Often information would be spread out and not available in a single article.

### V. METHODOLOGY

#### Steps involved to generate the abstraction based summary

**STEP 1:** Here the text is used collected from the user as input for summarizer.

**STEP 2:** In this step collected text is cleaned, means deleting the stop words, special characters, numbers which is irrelevant to text and punctuations

**STEP 3:** In this step word token and sentences token are created this process is called Tokenization

**STEP 4:** In this step by those tokens created in pervious step, frequency is found for every word in the users input text.

**STEP 5:** Here in this step weights are assigned to words.

**STEP 6:** Based on the weights, most top rated 20% weighted sentences are called final summary.

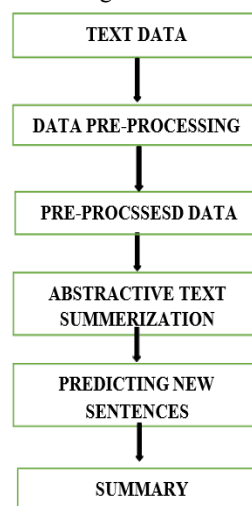


Figure 1: Summary Generation Process

**VI. IMPLEMENTATION**

The pseudocode of a NLTK TEXT- SUMMARIZER mechanism

Start

INPUT: Un-summarized Text

Output: Summary For given input Text

1. Import nltk , (import by typing “ PIP INSTALL NLTK” command)
2. import Stopwords
3. Def nltk\_summarizer(TEXT)
4. SW = set(stopword.word(“English”))
5. words = word\_tokenize(TEXT)
6. freqTable=dict( )
7. // Removing Stop Words
8. for word in words
9. word = word.lower( )
10. if word not in stopWords
11. if word in freqTable
12. freqTable[word] += 1
13. else
14. freqTable[word] = 1
15. end for
16. sentence\_list = sent\_tokenize(docx)
17. max\_freq = max(freqTable.values( ))
18. for word in freqTable.keys( )
19. freqTable[word] = (freqTable[word]/max\_freq)
20. sentence\_scores = { }
21. for sent in sentence\_list
22. for word in nltk.word\_tokenize(sent.lower( ))
23. if word in freqTable.keys( )
24. if len(sent.split( ‘ ’)) < 30
25. if sent not in sentence\_scores.keys( )
26. sentence\_scores[sent] = freqTable[word]
27. else
28. sentence\_scores[sent] += freqTable[word] //total number of length of words.
29. end for
30. end for
31. return summary
32. Stop

**VII. FLOW CHART**

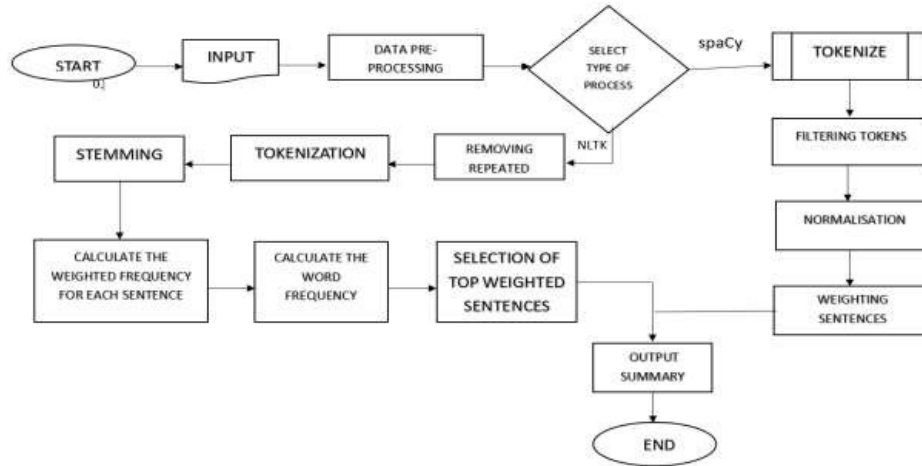


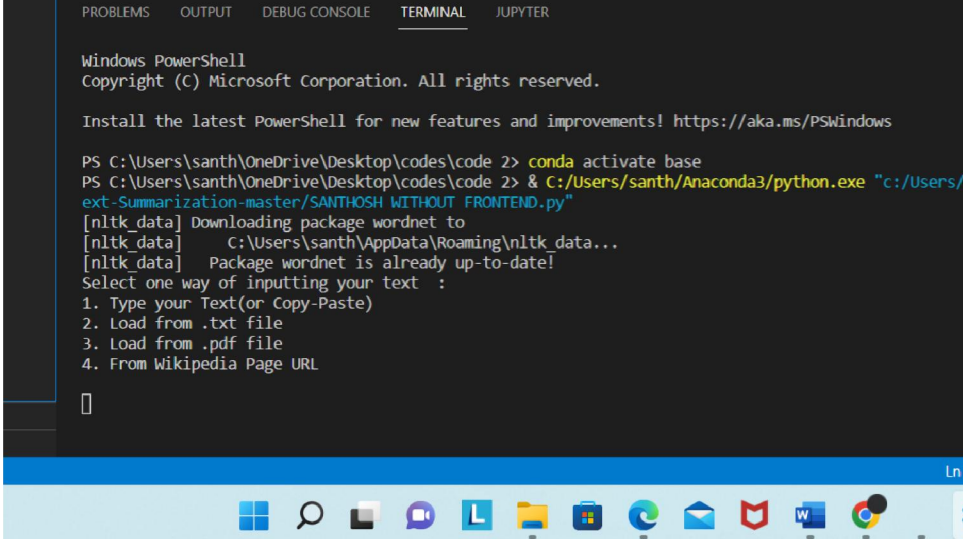
Figure 2: Flow Chart

**VIII. TESTING**

| TC No | TEST CASE  | EXPECTED OUTPUT   | OBTAINED OUTPUT   | RESULT |
|-------|--|---|---|--------|
| 1     | Passing the text input                                 | The text should be read and displayed in the input area | The text should be read and displayed in the input area | Pass   |
| 2     | Passing the text input as null                         | Show alert messages "Enter the text"                    | Show alert messages "Enter the text"                    | Pass   |
| 3     | Selecting the NLTK method                              | Ready to create summary                                 | Ready to create summary                                 | Pass   |
| 4     | Without selecting the NLTK method                      | An error should be thrown specifying "importing NLTK"   | An error is thrown                                      | Pass   |
| 5     | The model should return reduced and meaningful summary | Abstractive summary will be displayed                   | Abstractive summary will be displayed                   | Pass   |

**IX. RESULTS**

- The solution should take the desired length of summary from the user as an input should return summarized output.
- The most important output of these Abstraction based text summarizer is to reduce the reading time.
- Abstraction based text summarization produces meaningful sentences.
- It makes the user to read the summarized output easily.
- It gives short, exact and more content full summary without repetitive summary.



```

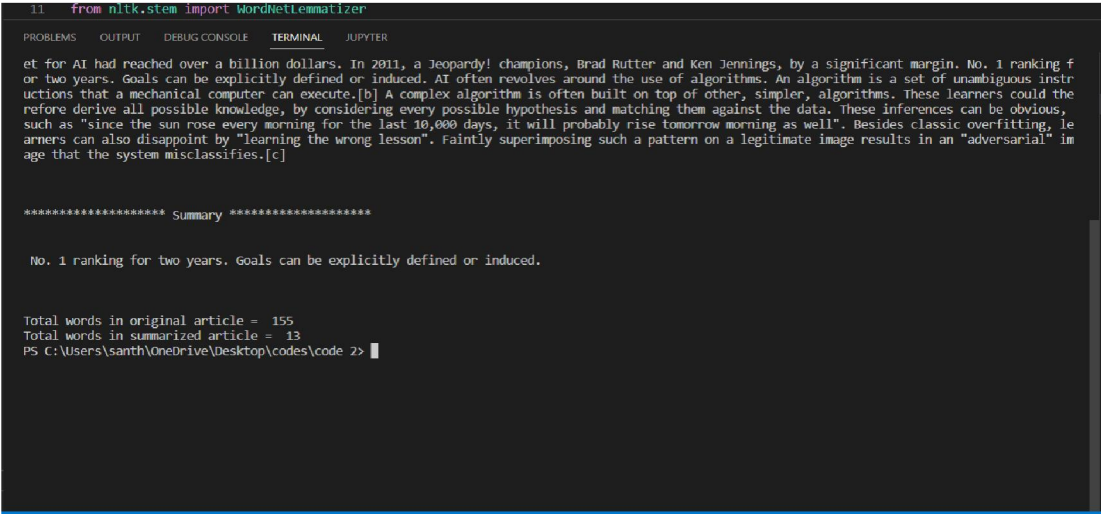
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\santh\OneDrive\Desktop\codes\code > conda activate base
PS C:\Users\santh\OneDrive\Desktop\codes\code > & C:/Users/santh/Anaconda3/python.exe "c:/Users/santh/OneDrive/Desktop/codes/ext-Summarization-master/SANTHOSH_WITHOUT_FRONTEND.py"
[nltk_data] Downloading package wordnet to
[nltk_data] C:\Users\santh\AppData\Roaming\nltk_data...
[nltk_data] Package wordnet is already up-to-date!
Select one way of inputting your text :
1. Type your Text(or Copy-Paste)
2. Load from .txt file
3. Load from .pdf file
4. From Wikipedia Page URL
  
```

### 9.1 Output Options



```

11 from nltk.stem import WordNetLemmatizer

et for AI had reached over a billion dollars. In 2011, a Jeopardy! champions, Brad Rutter and Ken Jennings, by a significant margin. No. 1 ranking f
or two years. Goals can be explicitly defined or induced. AI often revolves around the use of algorithms. An algorithm is a set of unambiguous instr
uctions that a mechanical computer can execute.[b] A complex algorithm is often built on top of other, simpler, algorithms. These learners could the
refore derive all possible knowledge, by considering every possible hypothesis and matching them against the data. These inferences can be obvious,
such as "since the sun rose every morning for the last 10,000 days, it will probably rise tomorrow morning as well". Besides classic overfitting, le
arners can also disappoint by "learning the wrong lesson". Faintly superimposing such a pattern on a legitimate image results in an "adversarial" im
age that the system misclassifies.[c]

***** Summary *****

No. 1 ranking for two years. Goals can be explicitly defined or induced.

Total words in original article = 155
Total words in summarized article = 13
PS C:\Users\santh\OneDrive\Desktop\codes\code >
  
```

#### Summary Output for Given Input Text

#### REFERENCES

- [1]. L. Abualigah, M.Q. Bashabsheh, H. Alabool, M. Shehab Text summarization: A brief review Stud. Comput. Intell., 874 (2020), pp. 1-15, 10.1007/978-3-030-34614-0\_1
- [2]. K. Al-sabahi, Z. Zuping, M. Nadher A hierarchical structured self-attentive model for extractive document summarization (HSSAS) IEEE Access XX (2018), 10.1109/ACCESS.2018.2829199
- [3]. A. Alzuhair, M. Al-dhelaan An approach for combining multiple weighting schemes and ranking methods in graph-based multi-document summarization IEEE Access, 7 (2019), pp. 120375-120386, 10.1109/ACCESS.2019.2936832
- [4]. D. Anand, R. Wagh Effective deep learning approaches for summarization of legal texts J. King Saud Univ. - Comput. Inf. Sci. (2019), 10.1016/j.jksuci.2019.11.015
- [5]. A.M. Azmi, N.I. Altmami An abstractive Arabic text summarizer with user controlled granularity Inf. Process. Manag., 54 (2018), pp. 903-921, 10.1016/j.ipm.2018.06.002
- [6]. S.A. Babar, P.D. Patil Improving performance of text summarization Procedia Comput. Sci., 46 (2015), pp. 354-363, 10.1016/j.procs.2015.02.031

- [7]. K. Chen, S.-H. Liu, H.-M. Wang An Information distillation framework for extractive summarization IEEE/ACM Trans. Audio Speech Lang. Process., 26 (2018), pp. 161-170
- [8]. Chen, Student Member, S. Liu, Student Member, B. Chen, H. Wang, Senior Member E. Jan, W. Hsu, H. Chen Extractive broadcast news summarization leveraging recurrent neural network language modeling techniques IEEE/ACM Trans. Audio Speech Lang. Process., 23 (2015), pp. 1322-1334
- [9]. H. Christian, M.P. Agus, D. Suhartono Summarization using term frequency-inverse document frequency (TF-IDF) Com. Tech., 7 (2016), pp. 285-294 <https://doi.org/10.21512/comtech.v7i4.3746>
- [10]. F.C.T. Chua, S. Asur Automatic summarization of events from social media Proc. Seventh Int. AAAI Conf. Weblogs Soc. Media (2009), pp. 81-90