Auto Text Summarization

AKHIL PAWAR[1], SUHAS TAMBE[2], ADITYA KIRTANE[3], M.R. GORBAL[4]
[1, 2, 3, 4] *Department of Information Technology, Shivajirao S. Jondhale College of Engineering, Dombivli.*

Abstract- ***Automatic text summarization is basically summarizing of the given paragraph using natural language processing and machine learning. There has been an explosion in the amount of text data from a variety of sources. This volume of text is an invaluable source of information and knowledge which needs to be effectively y summarized to be useful. In this review, the main approaches to automatic text hello. We review the different processes for summarization and describe the effectiveness and shortcomings of the different methods. Two types will be used i.e.-extractive approach and abstractive approach. The basic idea behind summarization is finding the subset of the data which contains the information of all the set. There is a great need to reduce unnecessary data. It is very difficult to summarize the document manually so there is the great need of automatic method. The extractive approach basically chooses the various and unique sentences, sections and so forth make a shorter type of the first report. The sentences are estimated and chosen based on accurate highlights of the sentences. In the Extractive technique, we have to choose the subset from the given expression or sentences in given frame of the synopsis.***

Indexed Terms- Auto Text; Extractive; Summarization

I. INTRODUCTION

With the developing measure of data, it has turned out to be hard to discover brief data. In this way, it is critical to making a framework that could condense like a human. Programmed content rundown with the assistance of Normal Dialect Handling is an instrument that gives synopses of a given archive. Content Outline strategies is divided in two ways i.e. - extractive and abstractive approach. The extractive approach basically chooses the various and unique sentences, sections and so forth make a shorter type of the first report. The sentences are estimated and chosen based on accurate highlights of the sentences. In the Extractive technique, we have to choose the subset from the given expression or sentences in given frame of the synopsis. The extractive outline frameworks depend on two methods i.e. - extraction and expectation which includes the arrangement of the particular sentences that are essential in the general comprehension the archive. What's more, the other methodology i.e., abstractive content synopsis includes producing completely y new articulations to catch the importance of the first record. This methodology is all the more difficult but on the other hand is the methodology utilized by people.

New methodologies like Machine taking in procedures from firmly related fields, for example, content mining and data recovery have been utilized to help programmed content synopsis.

Automatic text summarization is the process of shortening a text documentation using a system for prioritizing information. Technologies that generate summaries take into account variables such as length, style, and syntax. Text summarization from the perspective of humans is taking a chunk of information and extracting what one deems most important. Automatic text summarization is based on the logical quantification of features of the text including, weighting keywords, and sentence ranking.

II. PROBLEM DEFINITION

In the new period, where tremendous measure of data is accessible on the Web, it is most vital to give the enhanced gadget to get data rapidly. It is extremely intense for individuals to physically pick the synopsis of expansive archives of content. So, there is an issue of scanning for vital reports from the accessible archives and discovering essential data. Along these lines programmed content rundown is the need of great importance. Content rundown is the way toward recognizing the most vital important data in a record or set of related archives. What's more, compact them into a shorter rendition looking after its implications.

III. LITERATURE SURVEY

Title	Automatic Text Summarization Approaches
Authors	Ahmad T. Al-Taani (Ph.D., MSc, BSc) Professor of Computer Science (Artificial Intelligence) Faculty of Information Technology and Computer Sciences Yarmouk University, Jordan.
Year of publication	August 2017
Summary	Automated Text summarization systems are important in Many aspects in a language like natural language processing. ATS creates the summary of given document which save time and resources. There are single and multi-document text summary. Only y one document is extracted in case of single document summarization whereas group of documents is selected in multi document summarization. On other hand, mathematics techniques make the extract summarization language independent to theoretical ways. In analysis, we tend to the thought of the utilization of extract summarization methodology. There are two content-based summaries i.e. - generic and query-based summaries. In the generic summarization system if the user doesn't have knowledge about text, then informant measure equal level in information. Whereas in query-based summarization, before starting of the summarization technique, to is verified of the initial text. In statistical approaches, researchers are based upon sentence ranking and the important sentences are selected from the give document, regarded as the important summary compression ratio Graph-based approaches concentrate on the semantic analysis and relationship among sentences. The graph-based approach is used in the representation for text inside documents.

Title	A Survey of Text Summarization Extractive Techniques
Authors	Vishal Gupta University Institute of Engineering & Technology, Computer Science & Engineering, Punjab University Chandigarh, India Gurpreet Singh Lehal

	Department of Computer Science, Punjabi University Patiala, Punjab, India
Year of publicati-ons	August 2010
Summary	Text Summarization flattens the document into summary which maintain its important information. It becomes very difficult for human beings to summarize the paragraph. There are basically two approaches i.e.- extractive and abstractive summarization. Extractive approach - The sentences which are important are selected from the provided document and converts into summary. Based on its statistical and semantic features the importance is decided of the particular sentence.

Title	Framework of automatic text summarization using Reinforcement learning
Authors	Seonggi Ryang, Graduate school of Information science and technology, University of Tokyo Takeshi Abekawa, National institute of informatics
Year of Publication	August 2012
Summary	Well organized summary is generated of single and multiple documents. Multi-document summarization has become very important part of our daily lives as there is lot of information about one particular topic so it becomes very difficult to read. Summary of document helps to easily understand about the topic and important information is generated. The extractive approach is used which is popular for document summary. Summary is generated by selecting words and sentences from the provided document because it is difficult to guarantee the linguistic quality. Marginal relevance (MMR) is used which is used to score every textual unit and take out the highest score. Greedy MMR algorithm is also used but due to its greediness they don't take into account the whole quality. Global inference algorithm is also used for summary. However, these algorithms create lot of problem in formulation of integer linear programming for scoring and the time complexity is very hard. So, there is great need of efficient algorithms. In this paper the new approach in generated called Automatic

	Summarization using Reinforcement Learning (ASRL), where the summary is generated within framework and scores the function of summary. The method is used and adapts to problem with automatic summarization in natural way. Sentence compression is also adapted as action of framework. ASRL is evaluated which is comparable with the state of ILP-style taking rouge score into consideration. Evaluation is done on basis of execution time. State space is searched efficiently for sub optimal solution underscore functions and the score function, and produce a summary whose score denotes the expectation of the score of the same features' states. The quality of summary only depends on score function.

IV. CONCEPTUAL OVERVIEW OF THE PROJECT

Automatic text summarization is the process of shortening a text documentation using a system for prioritizing information. Technologies that generate summaries take into account variables such as length, style, and syntax. Text summarization from the perspective of humans is taking a chunk of information and extracting what one deems most important. Automatic text summarization is based on the logical quantification of features of the text including, weighting keywords, and sentence ranking.

- Extractive Text Summarization

Extractive text summarization does not use words aside from the ones already in the text, and selects some combination of the existing words most relevant to the meaning of the source. Techniques of extractive summarization include ranking sentences and phrases in order of importance and selecting the most important components of the document to construct the summary. These methods tend to more robust because they use existing phrases, but lack flexibility since they cannot use new words or paraphrase.

Algorithm 1 Text Rank Algorithm

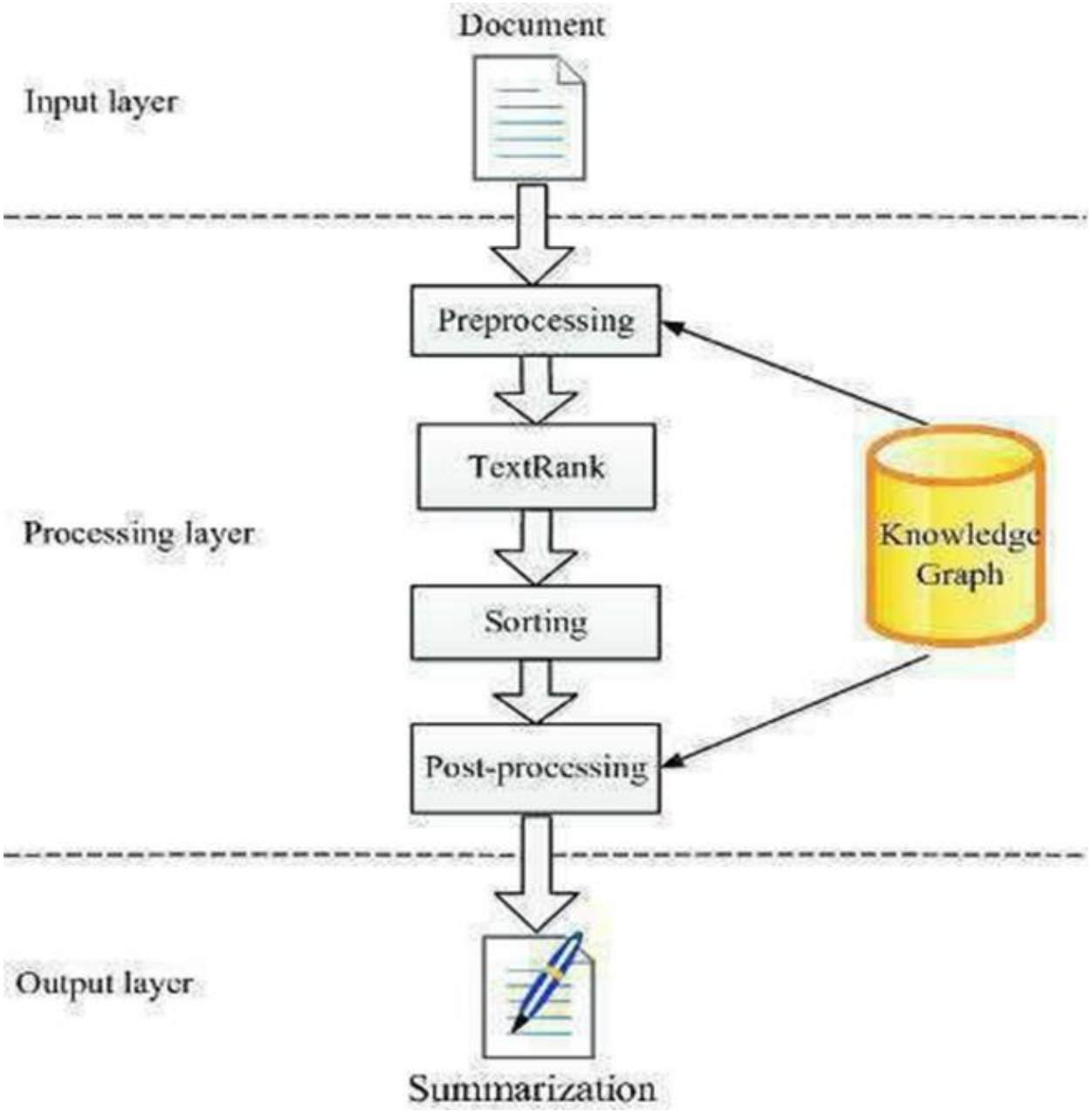
1: procedure TEXTRANK ALGORITHM
2: Identify filtered text units most representative of the text and add them as vertices to the graph.
3: Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph.
4: Iterate the graph-based ranking algorithm until convergence.
5: Sort vertices based on their final score. Use the values attached to each vertex for ranking/selection decisions.

- Text Rank for Sentence Extraction

To apply TextRank, firstly graph associated with the text is build, where the graph vertices are representative for the units to be ranked. The main goal is to rank entire sentences; therefore, the vertex is added to the graph for each sentence of the text. Next, connection between two sentences is determined by similarity relations between them, and similarity is measured by content overlap. A link is drawn between two sentence nodes if they share mostly common content. The measure of content overlap is determined by semantic similarity algorithm discussed in previous steps. To avoid long sentences, a normalization factor is used, and divides the content overlap by it. The resulting graph which is produced of sentences as vertex and edges representing the similarities with a weight associated with each edge. The text is therefore represented as a weighted graph, and consequently the weighted graph-based ranking formula is used as discussed in Page Rank algorithm section. After the ranking algorithm run on the graph, top ranked sentences are selected for the summary on the basis of their scores.

First, we take the input text and split the entire text down to individual words. Using a list of stop words, words are filtered so that only nouns and adjectives are considered. Then a graph of words is created where the words are the nodes/vertices. Each vertex's edges are defined by connections of a word to other words that are close to it in the text. The TextRank algorithm is then run on the graph. Each node is given a weight of 1. Then, we go through the list of nodes and collect the number of edges and connections the word has, which is essentially the influence of the connected vertex. The scores are computed and normalized for every node, and the algorithm takes the top-scoring words that have been identified as important keywords. The algorithm sums up the scores for each of the keywords in all of the sentences, and ranks the sentences in order of score and significance. Finally, the top K sentences are returned to become the TextRank generated summary.

- Design:



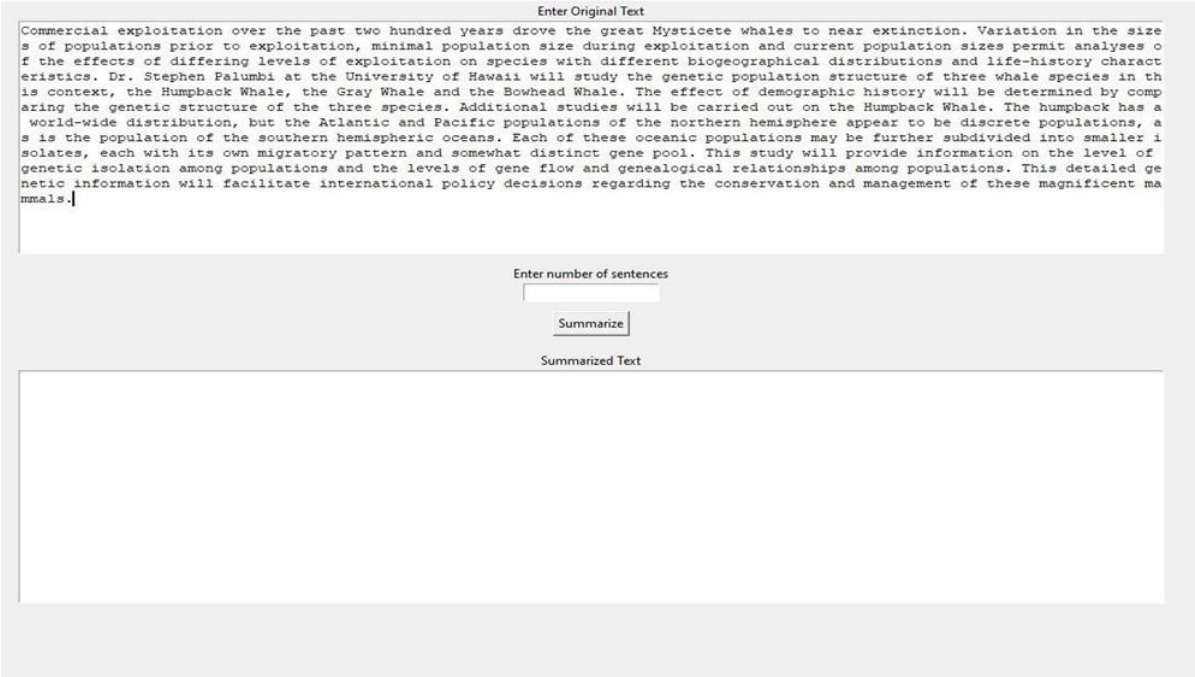
ats.py - I:\project\ats.py (3.9.0)

File Edit Format Run Options Window Help

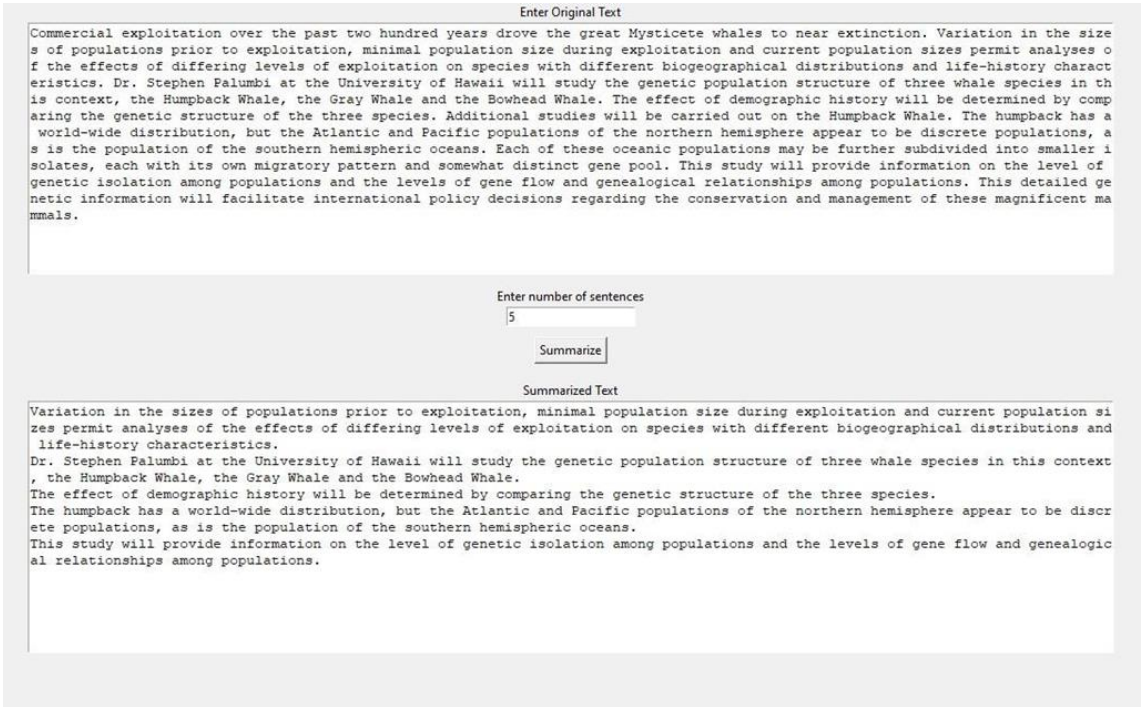
```
from tkinter import *
import summarizer

root = Tk()

root.geometry("1280x720")

ogTextLabel = Label(root, text="Enter Original Text")
ogTextLabel.pack()
ogTextBox = Text(root, height=15, width=130)
ogTextBox.pack()

numSentenceText = Label(root, text="Enter number of sentences")
numSentenceText.pack(pady=(10, 0))
numSentenceEntry = Entry(root)
numSentenceEntry.pack()

summarizedText = Text(root, height=15, width=130)

def summarizeText():
    ogText = str(ogTextBox.get("1.0", "end-1c"))
    numSentence = numSentenceEntry.get()

    ogText = ogText.replace('\n', ' ').replace('\r', '')

    summaryList = summarizer.summarize(str(ogText), int(numSentence))
    summary = "\n".join(summaryList)

    summarizedText.delete('1.0', END)
    summarizedText.insert(END, summary)


summarizeBtn = Button(root, text="Summarize", command=summarizeText)
summarizeBtn.pack(pady=(10, 0))

summarizedTextLabel = Label(root, text="Summarized Text")
summarizedTextLabel.pack(pady=(15, 0))
summarizedText.pack()

root.mainloop()
```

V. RESULT

- Input:



- Output:



CONCLUSION

As with time internet is growing at a very fast rate and with-it data and information is also increasing. it will be going to be difficult for human to summarize large amount of data. Thus, there is a need of automatic text summarization because of this huge amount of data.

Until now, we have read multiple papers regarding text summarization, natural language processing and lesk algorithms. There are multiple automatic text summarizers with great capabilities and giving good results. We have learned all the basics of Extractive and Abstractive Method of automatic text summarization and tried to implement extractive one. We have made a basic automatic text summarizer using NLTK library using python and it is working on small documents. We have used extractive approach to do text summarization.

We have successfully implemented state-of-the-art model for abstractive sentence summarization to recurrent neural network architecture. The model is a simplified version of the encoder-decoder framework for machine translation. The model is trained on the Amazon-fine-food-review.

ACKNOWLEDGEMENT

With due respect and gratitude, we take the opportunity to thank all those who helped us directly and indirectly. We feel pleasure in expressing our Heartfelt gratitude and vote of thanks to our guide Prof. M.R. Gorbal, who guided us in difficult situations. We would also like to thank our respected Head of Department Dr. Savita Sangam for providing unlimited access to all possible resources and encouragement.

REFERENCES

[1] Jing, Hongyan. “Sentence Reduction for Automatic Text Summarization.” Proceedings of the Sixth Conference on Applied Natural Language Processing -, 2000,

[2] Garg, Sneh, and Sunil Chhillar. “Review of Text Reduction Algorithms and Text Reduction Using Sentence Vectorization.” International Journal of Computer Applications, vol. 107, no. 12, 2014, pp. 39–42.,

[3] JRC1995. “JRC1995/Abstractive-Summarization.” GitHub, github.com/JRC1995/Abstractive-Summarization/blob/master/Summarization_model.ipynb.

[4] “A Gentle Introduction to Text Summarization.” Machine Learning Mastery, 21 Nov. 2017, machinelearningmastery.com/gentle-introduction-text-summarization/.

[5] “A Survey of Relevant Text Content Summarization Techniques.” International Journal of Science and Research (IJSR), vol. 5, no. 1, 2016, pp. 129–132.,

[6] “Text Summarization in Python: Extractive vs. Abstractive Techniques Revisited.” Pragmatic Machine Learning, rare-technologies.com/text-summarization-in-python-extractive-vs-abstractive-techniques-revisited/.

[7] “Text Summarization with TensorFlow.” Google AI Blog, 24 Aug. 2016, ai.googleblog.com/2016/08/text-summarization-with-tensorflow.html.

[8] “Encoder-Decoder Long Short-Term Memory Networks.” Machine Learning Mastery, 20 July 2017, machinelearningmastery.com/encoder-decoder-long-short-term-memory-networks/.