2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) | 978-1-7281-7741-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICACITE51222.2021.9404712

Extractive Automatic Text Summarization using SpaCy in Python & NLP

SWARANJALI JUGRAN
B.Tech CSE(Business Analytics)
Galgotias University Greater Noida, U.P.
India, Asia, 226001

ASHISH KUMAR
B.Tech CSE
Galgotias University Greater Noida, U.P.
India, Asia, 226001

BHUPENDRA SINGH TYAGI
B.Tech CSE(CIS)
Galgotias University Greater
Noida, U.P.
India, Asia, 226001

Mr. VIVEK ANAND
SCSE Department
Galgotias University Greater
Noida, U.P.
India, Asia, 226001

swaranjali.jugran2_2017@galgotiasuniversity.edu.in, ashish.kumar05_2017@galgotiasuniversity.edu.in, atintyagirocks@gmail.com
vivek.anand@galgotiasuniversity.edu.in

Abstract— **Propulsion of the everchanging technological innovations, has led to consider the data generated in the present era very crucial with significant roles both in technical & non-technical fields. In the digital world, as the amount of data produced at every instance is very huge; there is an ultimate need to develop a machine that can reduce the length of the texts automatically. Moreover, applying text summarization gears up the procedure of researching, reduces reading time, and increases the amount of important information being generated in the specific field. The main agenda is to develop a meaningful and coherent summary to recapitulate highlights of the text. From the collection of fascinating problems, we have opted for the Automatic Text Summarization. The solution to this problem unlike doing manually has proved to be essential in accurately summarizing voluminous texts in a cost and time efficient manner.**

Keywords— Extractive Text Summarization, NLP, Word-Tokenization, Sentence-Tokenization.

I. INTRODUCTION

In the modern world, where tremendous amount of data is accessible on digital platforms, it is important to make an enhanced tool to get the desired data rapidly. It is a tough task for individuals to manually select the gist of elaborated text. There is an issue of scanning such large reports from the accessible archives/text. Also, the main concern is to recognize the most important data in the document, large text records or set of related text. With the revolutionary and rapidly growing amount of data, discovering the crisp amount of information is challenging. There should be some tool which compresses them into a shorter interpretation looking after its implications. Hence, it is essential to make a model that could condense data like us. Designing such a model is the real task. The purpose of this project is to produce such a model as the solution which is based on Extractive Approach for summarizing text, starting with the Natural Language Processing as the fundamental model. The extractive approach is actually successful in delivering the summary using the same set of words which are actually most important words present in the actual text/archive; hence, it delivers the relevant information. From here, we come across with the effectiveness of different methods for distinguishing them on the basis of size & accuracy of summary. Here, these methods try to first understand the text and then mark the words according to their importance and then selecting the sentences containing the most important words in it and using them or the words used in their place to form the actual summary shortening the length of actual text. The procedure in all the methods is same i.e., Text->Text-Processing->Summary, where text is the input, text-processing is the intermediatory step & summary is the final output. One of the approaches referred as the abstractive approach which is one of the two important methodologies involved in Automatic Text Summarization works by giving the synopsis that includes new set of words. It is used to deliver the required summary with the same meaning as the original text. As it is clear, the extractive approach basically first selects various and unique sentences/sections of the text/document then combines them to form a summary. These sentences are selected on the basis of accurate highlights/scores described as the importance of the sentences. The importance plus meaning of the first record is maintained & preserved in both the cases. Here, we have opted for the Extractive-Approach. Using this approach, we could produce the relevant summary with a ratio of text to a summary as 2:1 or even better.

II. FUNDAMENTALS

A. Text

A text is a written document or any object that can be 'read', 'written', 'displayed', 'visualized', 'typed', 'interpreted', 'scanned' or 'printed' whether this object is a work of literature, articles published in newspapers, magazines, a type of document. It is a coherent set of signs, symbols, semantics and syntax that transmits some kind of information. Text represents textual document which is a written or printed work and regarded in terms of its content [1].

B. Summarization

Summarization is the process of making a summary of any text. A summary is a crisp statement or restatement of major points, especially as a conclusion to a work, it is actually a comprehension and usually brief extract, abstract or recapitulation of previously stated facts or statements. To summarize means to sum up the main points of something — a summarization is the kind of summation of a large document or huge amount of text [1].

C. Text-Summarization

Text summarization is the process in which long piece of texts gets a crisp format with lesser number of words than the actual text still reflecting the same meaning as the original doc/text [1].

III. Text Summarization Methods

The various dimensions of automatic text summarization can be generally categorized as different approaches based on certain characteristics like single or multiple document(s) specific or general purpose, learning Algorithm output-based (extractive or abstractive) [10].

A. Abstractive Approach

Abstractive summarization, is all about trying to comprehend the content of the text, generating synonyms or completely new words and utilizing them to make the Summary. It perhaps not contains the same sentences as present in the original text. This approach incorporates learning methods to make its own sentences but reflect the same meaning as the original text was providing [10].

B. Extractive Approach

Extractive summarization is one of the methods which incorporates making a summary on the basis of scoring technique. It marks the sentences containing important words with higher value as compared to the sentences containing least valued words. A subset of these high-valued sentences is selected within the boundaries of the text. There are two important parts for accomplishing this approach: extraction and expectation both required for extracting & grouping words & sentences according to their score to display them as the appropriate summary [10].

IV. Natural Language Processing

C. NLP the abbreviation of Natural Language Processing, is the branch of artificial intelligence that is the intersection of Computational Learning & Linguistics (Natural Languages) or the communicating tool of humans. Natural Language Processing is the part of advance technology used to give insights of natural languages to the machine. The objective-list of NLP extends from simple interpretation to complex comprehension i.e., to read, comprehend, interpret, decipher and make sense of the human languages in a manner that is meaningful to the machines. There are two main techniques i.e., one to explore meaning & other to find a proper usage [1]. It also involves the Text-Mining Approach which is procedural in nature → 1. creation of corpus -- 2. text-cleaning --

D. 3. feature engineering -- 4. model building. Bringing NLP in use is a beneficial option as it provides machine the ability to learn the natural languages and overcome it's weaknesses, also enhances the quality of learning by incorporating various programming & grammatical rules. NLP supports both paradigms Procedural & Object- based and both are equally essential- first for step- by-step execution latter for being executed. It performs the task of real understanding & proper usage of linguistic data by solving tasks in Python.

E. By providing the interfaces to the dictionary resources; machine can be taught/trained about identifying parts-of-speech, tenses, kind of sentences, etc. simultaneously it can learn the art of marking respective tags & dealing with libraries, So, on the basis of grammatical parameters, it is easy & efficient to use. The Natural Language Tool Kit (NLTK) or other text processing tools would allow individuals to divide text into smaller segments/sections by using partitioning methodologies, visualizing their syntactical usage, tagging them and finally projecting their actual meaning. Accomplishment of all this leads a machine to comprehend the main source of knowledge and generate substantial glance or representation in concrete form.

V. Text-Processing

The automated process of analysis & manipulation of the text is known as text processing [1]. It takes the text as input, processes it & finally provides the required outcome; it could be widely used within different areas of an organization, such as product teams could get insights from customer feedbacks to automate customer services. Here, words/tokens of the text represent discrete, categorical features.

A. Tokenization

Splitting into tokens**.** Tokens refers to any individual unit in the program which is meaningful to either the machine or the human.

Word-Tokenization: When the entire text is divided into individual words and word-score is generated for every word according to it's count.

Sentence-Tokenization: When the entire text is divided into individual sentences and each sentence is provided it's sentence-score according to the occurrence of the high-scored words.

B. SpaCy

A free, open-source library accurate for advanced Natural Language Processing (NLP) via Python. It comprehends & delineates the text (either small or large) by processing the same. Moreover, spacy provides wide range of in-built features which makes it an efficient tool for Text Processing & Language Modelling [1].

Module used:

python -m spacy download en_core_web_sm for small text document,

python -m spacy download en_core_web_lg for large text document [12].

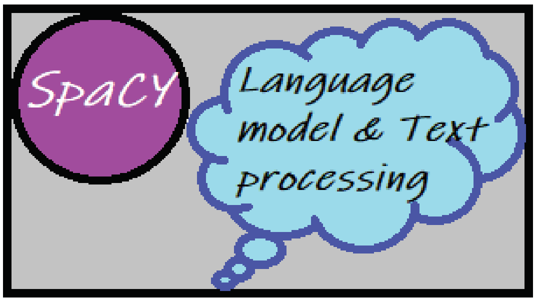

Fig.1
About
SpaCy

VI. Tables & Figures

A. *Relationship Between ML, DL & NLP*

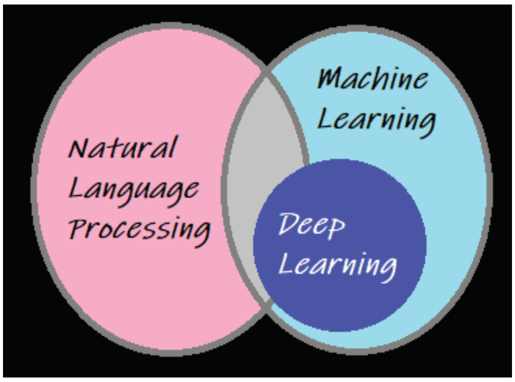

Fig.2 Venn Diagram

B. *Real Implementation Ratio*

Table 1: *Ratio Table*

INPUT	OUTPUT
Text	Summary
2(Two)	1(One)
Eg. Extract from IBM reports	Coherent Summary

C. *Comparision Table [1, 11]*

Table 2: *SpaCy, CoreNLP & NLTK (statistically)*

Package	Precision	Recall	F-Score
SpaCy	0.72	0.65	0.69
CoreNLP	0.79	0.73	0.76
NLTK	0.51	0.65	0.58

Table 3: *SpaCy, CoreNLP, NLTK (grammatically)*

Package	Tokenization	Tag
CoreNLP	2 milli-second(ms)	10ms
NLTK	4ms	443ms
SpaCy	0.2ms	1ms

D. *Schematic Diagram*

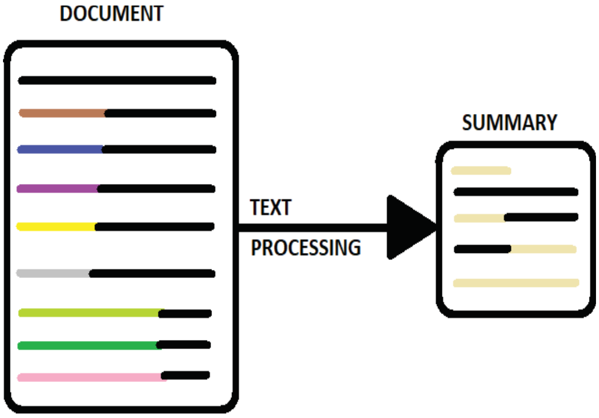

Fig.3 Text-Processing

E. *Architecture Diagram*

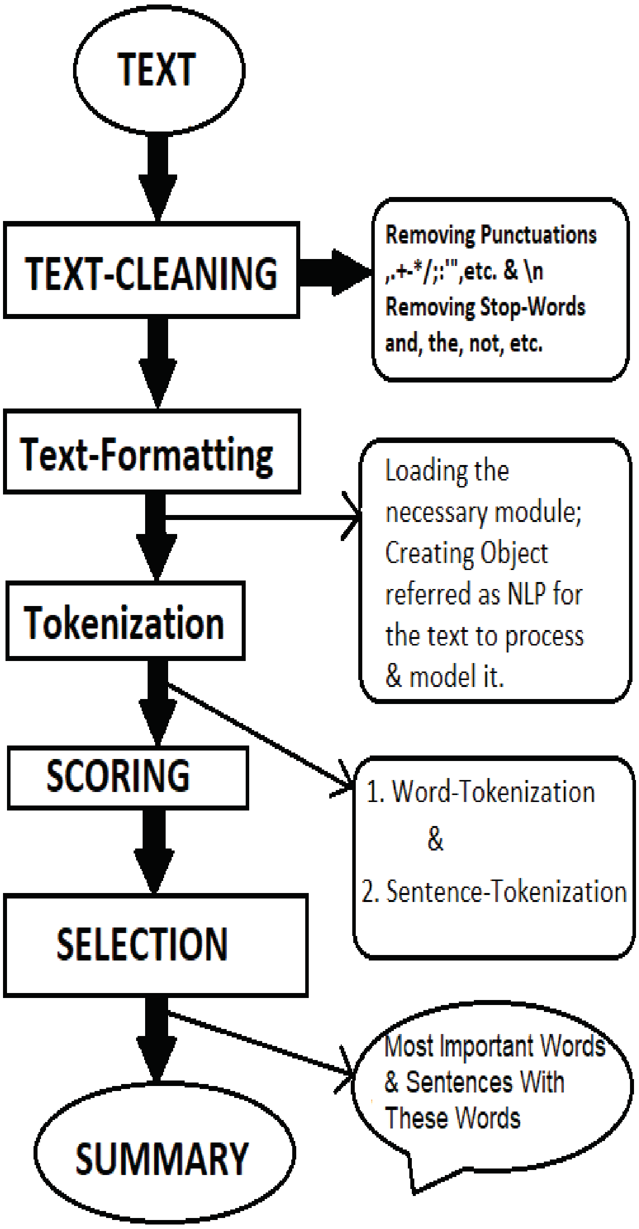
TEXT
TEXT-CLEANING
Removing Punctuations ,.+-*/;:'",etc. & \n
Removing Stop-Words and, the, not, etc.
Text-Formatting
Loading the necessary module; Creating Object referred as NLP for the text to process & model it.
Tokenization
1. Word-Tokenization & 2. Sentence-Tokenization
SCORING
SELECTION
Most Important Words & Sentences With These Words
SUMMARY

Fig.4 Step-by-Step Implementation

VII. Acknowledgement

A sincere thanks to my project guide Mr. Vivek Anand who guided me through all the endeavours of the project titled as "Extractive Automatic Text Summarization using SpaCy in Python & NLP". I learned a lot of new things and terminologies throughout this project. The accomplishment of this project has a big support of my team-mates Ashish Kumar And Bhupendra Singh Tyagi. We are extremely thankful to our family especially our parents for their moral support during the entire processing of our project. Thank you all for your kind support and guidance.

VIII. Conclusion

Existing Models were made on the basis of NLTK which is a library used for processing text string by string. The input and output using NLTK is the sequence of characters i.e., string. Providing several options of various algorithms for a particular problem is one of the specialities of this tool but it sometimes tends to be tedious & time- consuming to select and work accordingly. On the other hand, the proposed model, utilizes the library SpaCy which selects the best option itself becoming more time-efficient; it is based on the principles of object-oriented approach which is a key approach in programming nowadays. According to this it converts the text into one object as a whole. It includes word-vectors and this is lagging in previous tool, creating word vectors helps in proper assignment of real numbers to represent the meaning/efficacy of a word & clustering them accordingly, this makes Mathematical operation easy to use on these vectors. Spacy provides two modules marked with sm & lg incorporating small & large text respectively. Following table reveals the difference in both the packages:

Table 4: *SpaCy vs NLTK*

Models →	PREVIOUS	PROPOSED
Feature ↓	NLTK	SpaCy
Classifier	Yes	Yes
Topic Modelling	No	Yes
Vectorization	No	Yes
Tokenization	Yes	Yes
Parsing	Yes	Yes
TF-IDF	No	Yes

IX. Future Scope

In this project the extractive approach is explained, utilized and implemented; the next approach named as abstractive approach of Automatic Text Summarization could be the upcoming challenge, it is a technique wherein task of summarization becomes very complex as the whole text is required to be understood by the machine to generate a summary with entirely new words delivering the same meaning as the original text. Here, the model has to be trained with a lot of words & their synonyms, one word replacing many words & the correct usage of each word; RNN & LSTM are two future methodologies which would be incorporated to learn the words and to store their appropriate meanings, encoders-decoders & sequence-to- sequence model have to be utilised to produce efficient summary using this methodology. We would be extending the project in future to create the automatic text summarizer having the combination of both Extractive & Abstractive approach and would name it Hybrid text summariser.

X. References

[1] https://www.google.com for certain terms & their proper reference.

[2] Ahmad T. Al-Taani. "Automatic Text Summarization Approaches" International Conference on Infocom Technologies and Unmanned Systems (ICTUS'2017).

[3] Neelima Bhatia, Arunima Jaiswal, "Automatic Text Summarization: Single and Multiple Summarizations", International Journal of Computer Applications.

[4] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, Krys Kochut, "Text Summarization Techniques: A Brief Survey", (IJACSA) International Journal of Advanced Computer Science and Applications.

[5] Pankaj Gupta, Ritu Tiwari and Nirmal Robert, "Sentiment Analysis and Text Summarization of Online Reviews: A Survey" International Conzatiference on Communication and Signal Processing, August 2013.

[6] Vishal Gupta, Gurpreet Singh Lehal, "A Survey of Text Summarization Extractive Techniques." JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL. 2, NO. 3, AUGUST 2010.

[7] Jiwei Tan, Xiaojun Wan, Jianguo Xiao Institute of Computer Science and Technology, Peking University "Abstractive document summarization with a Graph- Based attentional neural model."

[8] Seonggi Ryang, Graduate school of Information science and technology, University of Tokyo "Framework of automatic text summarization using Reinforcement learning".

[9] Luhn, Hans Peter. "The automatic creation of literature abstracts." IBM Journal of research and development 2.2 (1958): 159–165.

[10] https://machinelearningmastery.com/gentle- introduction-text-summarization/

[11] www.analyticsvidhya.com

[12] www.SpaCy.io