Auto Text Summarization

By

Rutva K. Patel
16BIT129

NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Ahmedabad 382481

AUTO TEXT SUMMARIZATION

Mini Project – III

Submitted in fulfillment of the requirements

For the degree of

Bachelor of Technology in Information Technology

By

Rutva K. Patel
16BIT129

Guided By
PROF. MEENAXI TANK
[DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING]

NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Ahmedabad 382481

CERTIFICATE

This is to certify that the Mini Project -III entitled “Auto Text Summarization” submitted by Rutva K. Patel (16BIT129), towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Information Technology of Nirma University is the record of work carried out by him/her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Prof Meenaxi Tank
Ad-hoc Assistant Professor
Department of Computer Science & Engg.,
Institute of Technology,
Nirma University,
Ahmedabad

Dr. Madhuri Bhavsar
Dept. of Computer Science & Engineering,
Institute of Technology,
Nirma University,
Ahmedabad

ACKNOWLEDGEMENT

I would like to thank my project guide Prof. Meenaxi Tank for guiding me and sharing their knowledge on Machine Learning. I would also like to thank my fellow classmate Shikha Shah for helping me throughout the project and various contributor on Kaggle for sharing their approach on solving the same problem.

ABSTRACT

Internet has caused an information explosion. We are having lots of information which causes us to miss out on more important ones. Summary of a given text which contains major points can be useful for professionals as well as novice readers. Text summarization is a technique of extracting important information from a given text and presenting it in form of summary. I’ve performed both Extract and Abstract Auto text summarization in the paper. Extract Auto text summarization is done using PageRank Algorithm and Abstract Auto Text Summarization.

CONTENTS

CHAPTER 1: INTRODUCTION

Internet has caused an information explosion. Hence, Automatic Text Summarization is a technique to shorten a text document using Machine learning and data mining techniques to shorten the text document. There are various applications like Media monitoring, Newsletters, Search marketing and SEO, Internal document workflow, Financial research, Legal contract analysis, social media marketing, Question answering and bots, Video scripting, medical cases, Books and literature, email overload, E-Learning and class assignments, Science and R&D, Patent research, Meetings and video-conferencing, Help desk and customer support, helping disabled people, programming languages and Automated content creation. So basically, the definition- *the task of producing a concise and fluent summary while preserving key information content.*

There are generally two approaches to Auto Text Summarization:

1) Abstract Auto Text Summarization
2) Extract Auto Text Summarization

In Extractive Summarization, the summarized output is phrased by selecting the sentences from the input itself. While in Abstractive Summarization, it uses internal semantic representation and natural language processing techniques to create a summary that is closer to what humans would write.

CHAPTER 2: LITERATURE SURVEY

1. Structure Based Approaches:

This method identifies most important sentences through various features and structures such as tree, ontology, template etc. This method encodes most important text of Various Methods are
mentioned below:

Rule Based Method	
Description	→In this method, text to be summarized are represented in terms of list of features and categories. Features of a category is generated by information extraction rules. Then a content selection module selects the best feature from the ones generated. And then generation patterns are used for generation of summary sentences. In this method sentence is generated based on abstraction scheme. This scheme uses a rule-based information extraction module for sentence generation. →In abstractive summarization, we need to generate sentences. For this, it is essential to know semantics of a language. This method focusses on rules of grammar of language to check the grammatical correctness of the selected important sentences.
Advantages	It creates summaries with greater density of information than the original text
Limitations	All rules are written manually which is very time consuming and tiring.

Tree Based Approach	
Description	In this method similar sentences are pre-processed using shallow parser. Afterwards, those sentences are mapped to predicate argument structure. Then a phrase depicting the same meaning is generated. Use of language generator decreases grammatical mistakes and increases fluency.
Advantages	Use of language generator decreases grammatical mistakes and increases fluency.
Limitations	Context of sentence is not included while selecting phrases and even if it is not a part of common phrase, it is important part of sentence.
Template Based Approach	

Description	It uses template to represent a whole document. Linguistic patterns or extraction rules are matched to identify text snippets that will be mapped into template slots.
Advantages	It generates highly coherent summary because it relies on relevant information identified by IE system.
Limitations	Requires generalization of templates which is very difficult.

Ontology Based Approach	
Description	-Use ontology (knowledge base) to improve the process of summarization. It exploits fuzzy ontology to handle uncertain data that simple domain ontology cannot. -In this method, domain ontology for news event is defined by the domain experts. Next phase is document processing phase. Meaningful terms from corpus are produces in this phase. The meaningful terms are classified by the classifier on basis of events of news. Membership degree associated with various events of domain ontology. Membership degree is generated by fuzzy inference.
Advantages	Drawing relation or context is easy due to ontology. Handles uncertainty at reasonable amount.
Limitations	This approach is limited to Chinese news only. Creating rule-based system for handling uncertainty is a complex task.

2. Semantic Based Approaches:

Semantic representation of document is given to NLG system. This method processes linguistic data and identifies verb and noun phase. Various techniques are mentioned below:

Multimodal Semantic Analysis	
Description	In this method, a semantic model, which captures concepts and relationship among concepts, is built to represent the contents (text and images) of multimodal documents. The important concepts are rated based on some measure and finally the selected concepts are expressed as sentences to form summary. Multimodal document contains both text and images. The framework has three steps: In first step, a semantic model is constructed using knowledge representation based on objects (concepts) organized by ontology. In second step, informational content (concepts) is rated based on

	information density metric. The metric determines the relevance of concepts based on completeness of attributes, the number of relationships with other concepts and the number of expressions showing the occurrence of concept in the current document. In third step, the important concepts are expressed as sentences. The expressions observed by the parser are stored in a semantic model for expressing concepts and relationship.
Advantages	It produces abstract summary; whose coverage is excellent because it includes salient textual and graphical content from the entire document
Limitations	It is manually evaluated by humans. An automatic evaluation of the framework is desirable

Information Item Based method

Description	In this method, the contents of summary are generated from abstract representation of source documents, rather than from sentences of source documents. The abstract representation is Information Item, which is the smallest element of coherent information in a text. The framework consists of following modules: Information Item retrieval, sentence generation, sentence selection and summary generation. In Information Item (INIT) retrieval, first syntactic analysis of text is done with parser and the verb’s subject and object are extracted. So, an INIT is defined as a dated and located subject–verb–object triple. In sentence generation module, a sentence is directly generated from INIT using a language generator, the NLG. Sentence selection module ranks the sentences generated from INIT based on their average Document Frequency (DF) score. Finally, a summary generation step account for the planning stage and include dates and locations for the highly ranked generated sentences.
Advantages	It produces short, coherent, information rich and less redundant summary.
Limitations	-many candidate information items are rejected due to the difficulty of creating meaningful and grammatical sentences from them -linguistic quality of summaries is very low due to incorrect parses.

Semantic Graph Based Method

Description	This method aims to summarize a document by creating a semantic graph called Rich Semantic Graph (RSG) for the original document, reducing the generated semantic graph, and then generating the final abstractive summary from the reduced semantic graph. The abstractive approach

	proposed by consists of three phases. The first Phase represents the input document semantically using Rich Semantic Graph (RSG). In RSG, the verbs and nouns of the input document are represented as graph nodes along with edges corresponding to semantic and topological relations between them. The second phase reduces the generated rich semantic graph of the source document to more reduced graph using some heuristic rules. Finally, the third Phase generates the abstractive summary from the reduced rich semantic graph. This phase accepts a semantic representation in the form of RSG and generates the summarized text.
Advantages	It produces concise, coherent and less redundant and grammatically correct sentences
Limitations	This method is limited to single document abstractive summarization

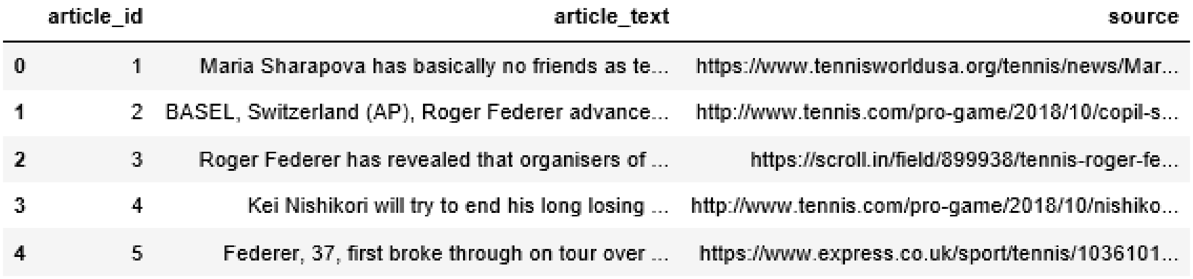
So, basically to conclude from the literature survey the technique used for the summarization majorly depends on criteria and its usage.

There are many criteria which should be considered while deciding which method to choose. Efficiency of method depends on content of text. In the same way, there are many criteria on which we can make summary. User should be aware that he/she wants to summarize the text based on which criteria. Some are mentioned below:

1. Summarization based on output
2. Summarization based on details
3. Summarization based on content
4. Summarization based on limitation
5. Summarization based on number of input texts
6. Summarization based on language acceptance

CHAPTER 3: DATASET USED

For the Extract Auto Text Summarization, I've used Articles related to Tennis dataset for the Extractive Auto Text Summarization. The dataset looks as shown below



	article_id	article_text	source
0	1	Maria Sharapova has basically no friends as te...	https://www.tennisworldusa.org/tennis/news/Mar...
1	2	BASEL, Switzerland (AP), Roger Federer advance...	http://www.tennis.com/pro-game/2018/10/copil-s...
2	3	Roger Federer has revealed that organisers of ...	https://scroll.in/field/899938/tennis-roger-fe...
3	4	Kei Nishikori will try to end his long losing ...	http://www.tennis.com/pro-game/2018/10/nishiko...
4	5	Federer, 37, first broke through on tour over ...	https://www.express.co.uk/sport/tennis/1036101...

Figure

In this dataset there is are fields like article_id, article_text and its source. We have used the article id and the article text and Extracted the top 3 rank sentences for the summarization.

For the Abstract Auto Text Summarization, I've used the Food Reviews dataset from Amazon to build a model that summarizes each review. In this dataset there are various fields like Id, ProductId, UserId, ProfileName, HelpfulnessNumerator, HelpfulnessDenominator, Score, Time, Summary and Text. Specifically, I will be using the description of a review as our input data, and the title of a review as our target data.

For instance,

Description (1): The coffee tasted great and was at such a good price! I highly recommend this to everyone!

Summary (1): great coffee

The code is written in Python and TensorFlow 1.1 will be our deep learning library.

CHAPTER 4: PREPROCESSING

4.1 Preprocessing for Extract Auto text Summarization

We have used numpy, pandas, nltk and re libraries for this execution.

After inspecting the data, the first step will be splitting the text into individual sentences. We will use the sent_tokenize( ) function of the nltk library to do this.

We have further used Glove word Embeddings. GloVe word embeddings are vector representation of words. These word embeddings will be used to create vectors for our sentences. We could have also used the Bag-of-Words or TF-IDF approaches to create features for our sentences, but these methods ignore the order of the words (and the number of features is usually pretty large). We will be using the pre-trained Wikipedia 2014 + Gigaword 5 GloVe vectors.

After extracting word embedding or word vectors, we now have word vectors for 400,000 different terms stored in the dictionary – 'word_embeddings'.

Text Preprocessing:

Basic text Preprocessing is done by removing punctuations, numbers, special characters and making the alphabets lowercase. We also got rid of the stopwords with regular words. Stopwords are basically the words that are used in regular language like is, am, the, of, in, etc. This is done by downloading the stopwords from nltk and making a function to remove such words from our dataset.

The next step is to find similarities between the sentences, and we will use the cosine similarity approach for this challenge.

Other features used for preprocessing are word frequency information, length of sentences (avoiding too long and too short sentences), using cue words (positively correlated and negatively correlated), document structure (by giving scores to each position) (i.e. information based on

Structure- title words sections, position-genre, first or last sentence of paragraph), heading similarity, title similarity, sentence to sentence cohesion (ratio of the sum of the similarity value(SSS) of sentence i to the Max(SSS)).

After applying all the above-mentioned methods, we will find similarity between sentence by preparing a similarity matrix and we will use the cosine similarity approach for this challenge. We will use Cosine Similarity to compute the similarity between a pair of sentences and for that we use sklearn library

4.2 Preprocessing for Abstract Auto text Summarization

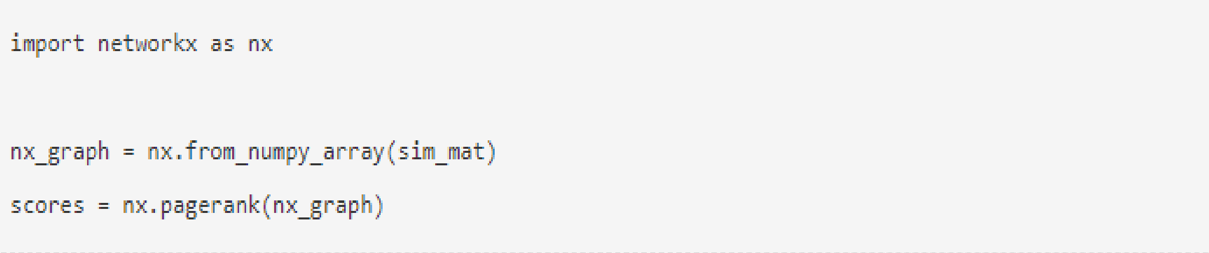
From the dataset we will first remove all the null values and the unneeded features using drop. Similar to the preprocessing done in Extract Auto Text Summarization, we will Remove unwanted characters, stopwords, and format the text to create fewer nulls word embeddings. We will remove the stopwords from the texts because they do not provide much use for training our model. However, we will keep them for our summaries so that they sound more like natural phrases. For word to vector conversion we have used Conceptnet Numberbatch's (CN) embeddings, similar to GloVe but gives better results. We find a) missing words b) high frequency words that are more than a threshold we have set (i.e. 20)

We then sort the summaries and text by length of texts, shortest to longest. We limit the length of summaries and texts based on the min and max ranges. We then remove reviews that include too many UNKs.

CHAPTER 5: TECHNIQUES USED

5.1 TECHNIQUES USED IN AUTO TEXT SUMMARIZATION

After creating a similarity matrix using cosine similarity scores, we convert the similarity matrix into a graph. The nodes of this graph will represent the sentences and the edges will represent the similarity scores between the sentences. On this graph, we will apply the PageRank algorithm to arrive at the sentence rankings.



```
import networkx as nx


nx_graph = nx.from_numpy_array(sim_mat)

scores = nx.pagerank(nx_graph)
```

The above shown code snippet is inbuilt code for applying pageRank Algorithm.

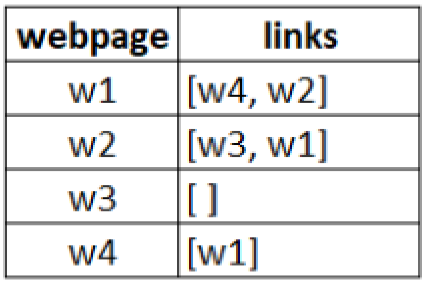
PageRank Algorithm:

PageRank Algorithm is inspired from Text Rank algorithm.

What is PageRank Algorithm exactly?

Suppose we have 4 web pages — w1, w2, w3, and w4. These pages contain links pointing to one another. Some pages might have no link – these are called dangling pages. Let us consider the instance shown below.

- Web page w1 has links directing to w2 and w4
- w2 has links for w3 and w1
- w4 has links only for the web page w1
- w3 has no links and hence it will be called a dangling page.

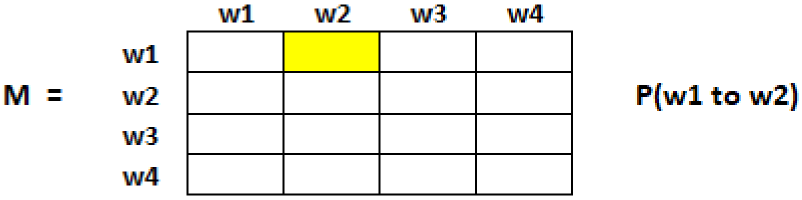


webpage	links
w1	[w4, w2]
w2	[w3, w1]
w3	[ ]
w4	[w1]

In order to rank these pages, we would have to compute a score called the PageRank score. This score is the probability of a user visiting that page.

To capture the probabilities of users navigating from one page to another, we will create a square matrix M, having n rows and n columns, where n is the number of web pages.

Each element of this matrix denotes the probability of a user transitioning from one web page to another. For example, the highlighted cell below contains the probability of transition from w1 to w2.



M =

	w1	w2	w3	w4
w1		▨		
w2				
w3				
w4				

P(w1 to w2)

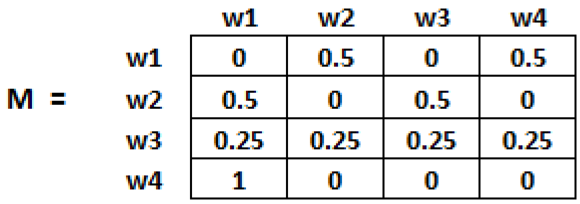
The initialization of the probabilities is explained in the steps below:

Probability of going from page i to j, i.e., M[i][j], is initialized with 1/(number of unique links in web page wi)

If there is no link between the page i and j, then the probability will be initialized with 0

If a user has landed on a dangling page, then it is assumed that he is equally likely to transition to any page. Hence, M[i][j] will be initialized with 1/(number of web pages)

Hence, in our case, the matrix M will be initialized as follows:



$$M = \begin{array}{c|c|c|c|c|}
 & w1 & w2 & w3 & w4 \\
\hline
w1 & 0 & 0.5 & 0 & 0.5 \\
\hline
w2 & 0.5 & 0 & 0.5 & 0 \\
\hline
w3 & 0.25 & 0.25 & 0.25 & 0.25 \\
\hline
w4 & 1 & 0 & 0 & 0 \\
\hline
\end{array}$$

Figure

CHAPTER 6: SUMMARY AND CONCLUSIONS

6.1 Summary

This project gave summarized version of the text in this case we have used 2 datasets the tennis player articles and amazon reviews. This project uses page rank algorithm for extractive auto text summarization and RNN (Recurrent Neural Network) for the abstract Auto text summarization.

This project further aims to summarize the articles related to politics which shall be retrieved through web Crawling using Beautiful Soup and implemented in a Voting Application through the use of API.

6.2 Conclusion

Using Extract Auto text summarization the top 3 sentences are printed according to the sentence score and in Abstract auto text summarization we have used RNN and obtained an accuracy score of 0.801. Thus here the RNN is more effective than other methods implemented.

REFERENCES:

[1] Mitrat, Mandar, Amit Singhal, and Chris Buckleytt. "Automatic text summarization by paragraph extraction." *Intelligent Scalable Text Summarization* (1997).

[2] Mitrat, M., Singhal, A. and Buckleytt, C., 1997. Automatic text summarization by paragraph extraction. *Intelligent Scalable Text Summarization*.

[3] Alguliev, Rasim, and Ramiz Aliguliyev. "Evolutionary algorithm for extractive text summarization." *Intelligent Information Management* 1.02 (2009): 128.

[4] Yousefi-Azar, Mahmood, and Len Hamey. "Text summarization using unsupervised deep learning." *Expert Systems with Applications* 68 (2017): 93-105.

[5] Chuang, Wesley T., and Jihoon Yang. "Text summarization by sentence segment extraction using machine learning algorithms." *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, Berlin, Heidelberg, 2000.

[6] Chuang, W.T. and Yang, J., 2000, April. Text summarization by sentence segment extraction using machine learning algorithms. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 454-457). Springer, Berlin, Heidelberg.

[7] Hu, M., & Liu, B. (2004, August). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 168-177). ACM.

[8] Hu, Minqing, and Bing Liu. "Mining opinion features in customer reviews." *AAAI*. Vol. 4. No. 4. 2004.

APPENDIX:

(1) https://towardsdatascience.com/text-summarization-with-amazon-reviews-41801c2210b

(2) https://www.kaggle.com/currie32/summarizing-text-with-amazon-reviews

(3) https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/

(4) https://github.com/