Text Summarization and Questionnaire Generation using Deep Learning

Helee Rana
Computer Science Department
Pace University
New York, USA
hr28113n@pace.edu

Jasmeet Kaur Ubhi
Computer Science Department
Pace University
New York, USA
ju50752n@pace.edu

Karandeep Singh Multani
Computer Science Department
Pace University
New York, USA
km74230n@pace.edu

Rhea Rai
Computer Science Department
Pace University
New York, USA
rhea.p.rai@pace.edu

Shreyas Damle
Computer Science Department
Pace University
New York, USA
sd57132n@pace.edu

Henry Wong
Professor, Computer Science Department
Pace University
New York, USA
hwong@pace.edu

Abstract— **In this modern age, where tremendous information is accessible, it is most important to have the improved mechanism for quickly and efficiently extracting the information. Human beings find it very difficult to manually retrieve the description of a huge text documents and ask the relevant questions in order to test knowledge gain from the described text. There is thus a problem with the search and absorption of relevant information. In order to solve these problems, we proposed the automatic text summarization system along with the generation of questions with the hope that potential benefits of an automated Question Generation system can help people meet their useful enquiry needs. Generation question from the uploaded text document and getting the answers of generated questions has become the two main challenge for natural language communities. In order to generate question from the text, set of words and single sentences has been given to the system and its goal would be to create set of questions from the set of words provided and give its answers from the summarized text.**

Keywords— ***text summarization, questionnaire generation, tokenization, natural language processing, bag of words, latent Dirichlet allocation, stemming***

I. Introduction

Text summarization and questionnaire generation is a process that takes a source text and presents the most important content in a condensed form in a manner sensitive to the user or task needs. The need of having an automatic text summarization is increasing day by day due to the expansion of information available on-line. System has to understand the point of text and it becomes a big problem for text summarization. It cannot be done without having world knowledge, which requires grouping of the content and semantic analysis. There are so many methods used, and attempts has been made to make system run efficiently but nothing was successful so far. Fortunately, an approximation called abstraction is more feasible today. The system simply needs to identify the most important passages of the text to produce an abstract, but the problem is text summary not always in coherent form. Thus, at present, most automated systems produce abstracts only. The proposed application is based on machine learning where "Text summarization and questionnaire generation" is the technique for generating a concise and precise summary of bulky texts while keeping the stress on the sections which has meaningful information and without losing the useful data where different methods are used like paraphrasing and shortening of text, just like humans do. Furthermore, there are certain algorithms and methods has been performed on the remaining summary at the back-end part from "Tokenization" to "Model Evaluation". By going through these methods, we obtain a precise meaningful summary which lastly gives us "Generated questions" and "Summarization of whole Text". As in today's world everybody like or tend to read the books, so this application will be tailored to all demographics from children to grandparents.

One of the main purposes of the project is the automatic generation of questions that will help user to improve their understanding of the knowledge they found from the uploaded document. Questions can be used to develop users' interest in topic and help the user to self-recognize the knowledge they gain from the document. Generating questions helps to server the purpose of reading comprehension instruction. The potential benefits of this automatic systems are helping to reduce human dependence to get queries and various desires related to systems that interact with natural languages.

II. Literature review

As times moves forward, parallelly technology is moving ahead with an even faster speed. With upgrading technology, it automates tasks that would take ages to complete manually.

.

This automation has made complicated tasks easier and less time consuming. Similarly, businesses are opting for methodologies which require lesser resources and provide accurate output in short amount of time.

Earlier on, automatic text summarization systems were developed which followed techniques to generate summaries using position method, location method etc. Importance of words and sentences were derived using the frequency of words using auto-abstract by Luhn[1]. Later on, a basic summarization system was created by G. J. Rath, A. Resnick, and T. R. Savage [2]. Few years down the line, researches were based on linguistic approaches.

As we approached the 90s, machine learning techniques were introduced, which not only brought in easy techniques, but multiple paged documents were also being used. In [3], Radev, Hovy and Mckeown extracted logical data and modified this piece of information. Achieving summary which is coherent to its original text is crucial. Previously developed systems lacked the functionality of coherence.

Also, systems didn't provide the functionality of deriving questions out of the extracted summary wherein our system stands out in providing both, coherent extracted summary along with set of questions being generated.

III. Problem statement

Off lately, there's been an explosion of information from various sources available in assorted formats. This assortment of data has made it even more cumbersome to get a logical explanation out of it. This load of data holds a significant value when converted into knowledgeable information. Summarizing a huge document and getting a meaningful summary out of it is humanely impossible.

The system that is being devised not only handles single paged documents but will also handle multiple paged documents. This system will produce an extractive/abstractive summary out of it therefore giving the user an easier alternative to further derive questions based on the summary produced. Questions will be generated from the summary derived.

The existing system address the following functionalities:

- Develop an abstractive content-aware summary of single paged documents of text.
- Ensure retrieval of content relevant to aspects of each category of documents being considered.
- Develop a method of forming different phrases to present the information being extracted.
- Produce simple, lucid and cohesive text, that would depict important aspects of the original text document.

IV. Tools and technologies

Our project is a web-based application in which user can login using its credential and will be to upload its Text document. The system will apply the NLTK algorithm to the uploaded document and gives the summary of the text document and generate the Questions from it. The frontend is managed by CSS and Bootstrap and the framework has been written in Django for Python3 using PyCharm IDE.

V. Proposed system

Initially, user uploads a text document from which questions need to be generated, into the application of any length and then pre-processing is implemented on the uploaded text document. In the process of pre-processing word tokenization, removal of stop words, stemming and removing punctuation is performed to get the main and useful content from the text document.

Word tokenization separates large text document into single words, every sentence is split into a single word. This is necessary for natural language processing as we need to process every single word to further classify and analyze it while generating questions from the text document.
We remove stop words in order to filter out the data. Every paragraph or a large text document has many words such as a, an, the, in, is etc. these words are being removed and the important text is filtered in the second step of stop words removal.
In the next step, stemming is performed on the text document. The process of stemming removes affixes, suffixes and prefixes of a word. Stemming is an important step to be performed while generating questions from a text document.

Further, we remove punctuations from the text such as ',', '!', '/' etc. to filter the text more. Every step is performed using the Natural Language Toolkit (NLTK) which contains libraries and programs for natural language processing. Hence, we now have the important text which we process further.
As we processed the word document and only useful text is now filtered, we store this useful text in a dictionary as every single separate word. In the dictionary with every word stored its frequency is also stored corresponding to the word. Calculating the frequency of each word helps to identify its occurrence and hence is efficient to further classify.

After performing all the above steps, we also store the sentences from text document in another variable and here every sentence is a token and this sentence is compared with the frequency we earlier calculated of every word. Therefore, now every sentence has a value or frequency according to the appearance of word and from its frequency. Once frequency or score of every sentence is calculated, average frequency of all sentences is computed, further summary is generated.
Further, summary is then computed from the threshold value (average computed in earlier step), each sentence score and the tokenized sentences. The sentences consisting greater value than the average value is considered important to generate summary. Summary is now generated.
The next crucial step is generating question from the processed document. From keywords that we separated according to their frequency, the most frequently occurred word is important to form a question. Here we have implemented 'Bag of Words (BOW)' and 'Latent Dirichlet Allocation (LDA)'. BoW is algorithm for natural language processing which is very simple to implement. This stores the important words from sentences

that occurred most frequently or according to their computed word score irrespective of their semantics. LDA is used to identify parts of similar data and their relationship of being similar.

Using bottom up approach and implementing the combination of both algorithms we frame a question. Bottom up approach is used as we have our solution generated i.e. summary and using the summary, we incline to frame a question from previously stored template of questions. To elaborate more, in a paragraph we identified 'Operating System' as an important key word, hence using previously stored questions template we frame a question "What is Operating System?".
Hence, summary and its relevant questions are generated.

VI. System Architecture

A user can login by giving the required credentials and creates an account. A login page is seen every time user opens the web page. User can further upload a document which he/she wishes to get the summary of or generate questions.
The document can be further broken down into chapters and important keywords.
This step initializes the backend functions and the first step of backend functions that is tokenization, comes into picture.

A. Tokenization: Tokenization is splitting the paragraph of text into single words or tokens. The whole document is tokenized and converted to separate tokens so that they could be handled further. These tokens are stored in the database and ready for pre-processing.

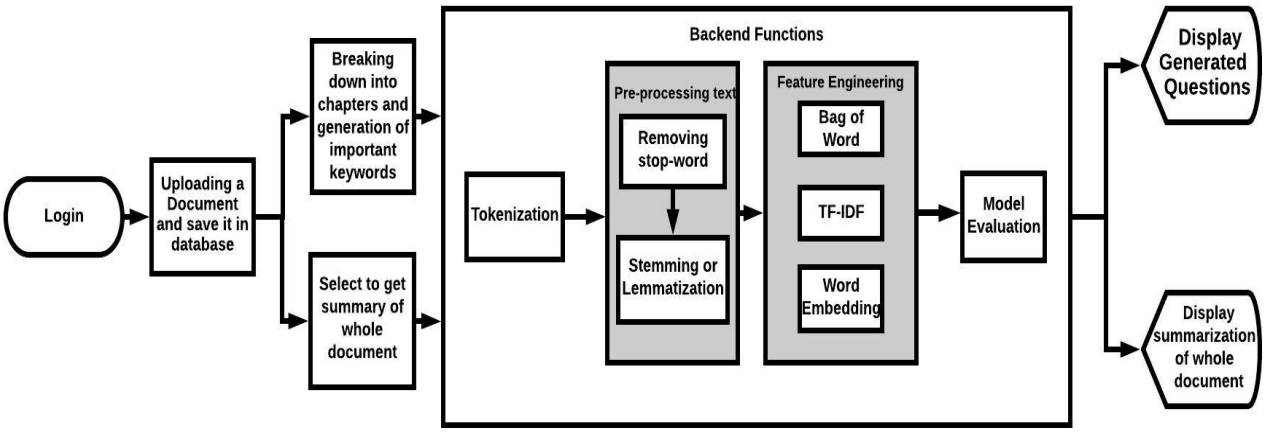

Fig 1. Project flow diagram

B. Text Pre-processing:

1. **Removing stop-words**: Stop-words are commonly used words that include 'the', 'a', 'an', 'that' and so on. We need to ensure that the stop-words are removed to not want these words taking up space in our database or taking up valuable processing time. They can be removed by storing a list of words that we want to remove, in the database.
2. **Stemming or Lemmatization**: In natural language processing, stemming is an approach where words in our document will get reduced to their stems or smaller form of that word. For example, if we have certain words in the document like 'worked ', 'working' and 'work, they belong to the same stem 'work'. Lemmatization involves finding a meaning and parts of speech of a word and grouping similar type of words. This is a crucial step in text pre-processing.

Our system will detect these words and form groups of similar words.

3. **Feature engineering:** After text preprocessing it is time to extract features out of the words.

A. Bag of words and LDA:
Bag-of-words approach calculates the count of the repeated words in the preprocessed text and also returns the highest count. This helps the system to know the important words in the document. While generating questions, these words are given the highest priority and based upon these words, the questions are framed. **Latent Dirichlet Allocation (LDA)** is used to identify parts of similar data and their relationship of being similar.

B. TF: TF stands for **"Term Frequency".** This is a technique will assign weights in the form of vectors to

C. the words of our document according to the importance.

D. Model Evaluation: Evaluation measures are to taken after the document is ready or the questions are generated in order to check whether it is valid in terms of factors such as structure, grammaticality, precision, information retrieval and many more.

Finally, the output is displayed in the form of a brief summary or questions that are generated.

VII. User use case diagram

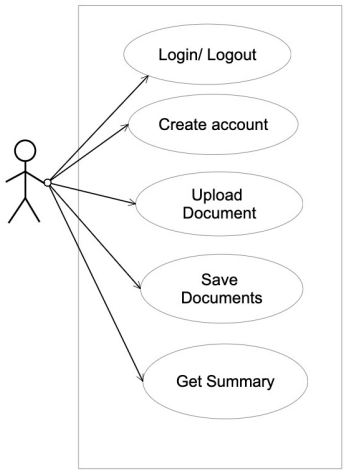

Fig 2. Use case diagram

Functional Requirements:

1. User will be able to create its account by sign up.
2. User have to give details like Username and meet the Password requirement and confirm its password.
3. Once account is created user will be able to login.
4. After login, user will be able to upload document by using the upload button and submit it.
5. Summary of uploaded document can be generating by the click of Summary button and summarized text will appear in text area.
6. User can use the service of Question generation by access the Question link from sidebar.

VIII. Key findings and observation

From the above defined procedure, we have successfully summarized text and generated questions. This section describes the key findings and comparisons.

The originally uploaded document consists of 1730 characters i.e. input text, is reduced to 1013 characters. This reduced text is important further. The document "summary of text" is the trimmed document result.

The following sample of code shows the text reduction:

```
len(text)
>>1730

len(summary_of_text)
>>1013
```

The top four high frequency important words extracted from the "summary_of_text" using Bag of Words approach are:
1. Word: - "System"; Occurred: 13 Times.
2. Word: - "Operating"; Occurred: 5 Times.
3. Word: - "user"; Occurred: 5 Times.
4. Word: - "computer"; Occurred: 4 Times.

The important observation noted here is that Bag of Words (BoW) can catch the technical important word "Operating System" from the whole text and concatenate it with question template.
The formula for calculating term frequency and weightage of every sentence is given below:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

Hence, the above listed are been considered as the most crucial tasks and the observations are also systematically defined.

IX. Result and analysis

There have been tremendous researches in every field to automate functions and reduce human effort by using Artificial Intelligence and Machine Learning techniques. We have proposed a system to extract information quickly and efficiently, involving less human effort. Summarization provides a wide-ranging overview of the text entered and lately many researches are being conducted for this abstractive summarization. Our implemented procedure contributes in reducing the manual work, and this is demanded in market a result of reduction in manual efforts. The procedures implemented is a crucial topic of research in Natural Language Processing (NLP). Our main motive lies in extracting symptomatic sentences or paragraphs from the original document. The concept of text summarization has number of approaches defined in Neural Networks, Fuzzy logic, NLP to a certain limit in extracting summary from any document or text. We impulse on implementing extractive method to define the algorithm of text summarization.

The main advantage lies for educational purpose as the application benefits by generating questions to test their skills. Self-learning is also encouraged with the use of this application as research have stated that deploying questions helps one to learn more.

X. Conclusion

Text summarization and questionnaire generation is an old and difficult challenge but right now research direction moved towards increasing industries like biomedicine, product review, education domains, emails and blogs. This is due to the overload information available on World Wide Web

Text summarization with the help of python library NLTK (Natural language toolkit) was successfully implemented. Additionally, automatic text summarization approaches based on tokenization techniques, Pre-processing text, Model evaluation, Feature engineering, successful in making an effective and efficient summary of the original document.

In this paper, we utilized various extractive approaches for document summarization and executed to obtain a brief summary. The most extensively used methods such as machine learning techniques, frequency driven methods were also presented in order to give an idea of backend functioning of our

model. Our paper provides a good insight on the working of algorithms and the output form after applying these algorithms.

XI. Future work

The future scope of this application has a lot of different horizons and a lot of features can be integrated within this application to make it more robust.

We plan on integrating:

- Different subjective solution generation approaches like abstractive summarization techniques using neural networks.
- Conversion of a pdf document to understandable and usable text format.
- Distinguishing Chapters and identifying various important topics from the chapters.
- Storing the corpora of uploaded documents, which can be used as a database for solution generation.
- Ethically Extracting more textual information from World Wide Web for a specific topic to help more efficient solution generation.
- Converting text from various documents such as PDF, Word, Images etc.

References

[1] H. P. Luhn, “The automatic creation of literature abstracts,” IBM Journal of Research and Development, vol. 2, pp.159-65, 1958.

[2] G. J. Rath, A. Resnick, and T. R. Savage, “The formation of abstracts by the selection of sentences,” American Documentation, Vol. 12, pp.139- 143, 1961.

[3] H. P. Edmundson., “New methods in automatic extracting,” Journal of the Association for Computing Machinery, Vol. 16, No. 2, pp.264-285, 1969.

[4] Dragomir R. Radev, Eduard Hovy and Kathleen McKeown,” Introduction to the special issue on summarization, Association for Computational Linguistics, Volume 28, Number 4, 2002.

[5] Ani Nenkova and Kathleen McKeown, ”Automatic summarization,“ Foundations and Trends in Information Retrieval, Vol. 5, Nos. 2–3, pp.103–233, 2011.

[6] R. Ferreira et al., “Assessing sentence scoring techniques for extractive text summarization,” Expert Systems with Applications, vol. 40, No. 14, pp.5755 – 5764, 2013.

[7]Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslami, Pooya Khosravyan Dehkordy, ”Optimizing text summarization based on fuzzy logic,” Seventh IEEE/ACIS International Conference on Computer and Information Science, 2009.

[8] M.A. Fattah and Fuji Ren, “Automatic text summarization,” International Journal of Computer Science, Volume 3, Number 1, 2009

[9] Vishal Gupta and Gurpreet Singh Lehal, “A survey of text summarization extractive techniques," Journal of Emerging Technologies in Web Intelligence, Vol. 2, No. 3, August 2010.