International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Automatic Text Summarization and Audio Generation

Tanooja K[1], Tejasri K[2], Akhilesh T[3], Prasanna Kavya M[4]

[1,2,3,4]Students, Department of CSE, Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, India

kosuru.tanooja[at]gmail.com, tejasri9550[at]gmail.com, akhilesh.tanneedi21[at]gmail.com, gracelouis0219[at]gmail.com

Abstract: ***Summarization is the task of reducing the size of the initial text while at the same time preserving key informational elements and the meaning of the content. Since manual text summarization is a time expensive and generally laborious task the automatization of the task is gaining increasing popularity. The framework first selects salient sentences and then independently condenses each of the selected sentences into a concise version. Finally an extractor utilizes the context information of the document to select candidates and assembles them into a summary. This summary is further converted into audio format.***

Keywords: Automatic Text Summarization, Sentence Ranking, Cosine Similarity, Text Rank, Audio Generation

1. Introduction

The information on world wide web is growing with an exponential rate. Therefore, it is necessary to provide the succinct form of the required information without losing its significance. Thus reducing the reading time desirable prospect as it can greatly decrease human effort as well help in finding the most important parts of any document or corpus of documents. Automatic text summarization (ATS) has been a successful solution for generating the shorter version of the input document without losing its main content. ATS can be categorized on the basis of its output as extractive and abstractive text summarization.

So here in our project we are using algorithms creating this text summarization and this is how useful it is nowadays.

The method of extracting these summaries from the original huge text without losing vital information is called Text Summarization. When you open news sites, do u just start reading every news article? Probably not.W e typically glance at the short news summary and then read more details if interested. Short ,informative summaries of the news is now everywhere like magazines ,news aggregators apps, research sites etc.

Well, it is possible to create the summaries automatically as the news comes in from various sources around the world.

Example: Banking system , E-learning, class ,assignments, and helping disabled persons.

1.1 Types of Summarization

In general, there are two different approaches for automatic summarization: extraction and abstraction.

1.1.1 Extractive Approach

Extractive summarization picks up sentences directly from the document based on a scoring function to form a coherent summary. This method works by identifying important sections of the text cropping out and stitching together portions of the content to produce a condensed version. Thus, they depend only on the extraction of sentences from the original text. Most of the summarization research today has focused on extractive summarization, once it is easier and yields naturally grammatical summaries requiring relatively little linguistic analysis. Moreover, extractive summaries contain the most important sentences of the input, which can be a single document or multiple document

1.1.2 The Abstractive Approach

Abstractive summarization methods aim at producing summary by interpreting the text using advanced natural language techniques in order to generate a new shorter text — parts of which may not appear as part of the original In this paper we address the automatic summarization task. Recent research works on extractive-summary generation employ some heuristics, but A few works indicate how to select the relevant features. We will present a summarization procedure based on the application of trainable Machine Learning algorithms which employs a set of features extracted directly from the original text. document, that conveys the most critical information from the original text, requiring rephrasing sentences and incorporating information from full text to generate summaries such as a human-written abstract usually does. In fact, an acceptable abstractive summary covers core information in the input and is linguistically fluent.Thus, they are not restricted to simply selecting and rearranging passages from the original text.

1.1.3 GTTS

There are several APIs available to convert text to speech in Python. One of such APIs is the Google Text to Speech API commonly known as the gTTS API. gTTS is a very easy to use tool which converts the text entered, into audio which can be saved as a mp3 file. The gTTS API supports several languages including English, Hindi, Tamil, French, German and many more. The speech can be delivered in any one of the two available audio speeds, fast or slow. However, as of the latest update, it is not possible to change the voice of the generated audio. The text-to-speech (TTS) is the process of converting words into a vocal audio form. The program, tool, or software takes an input text from the user, and using

methods of natural language processing understands the linguistics of the language being used, and performs logical inference on the text. This processed text is passed into the next block where digital signal processing is performed on the processed text. Using many algorithms and transformations this processed text is finally converted into a speech format. This entire process involves the synthesising of speech. Below is a simple block diagram to understand the same.

2. Literature Survey

2.1. Automatic Text Summarization Using a Machine Learning Approach

In this paper we address the automatic summarization task. Recent research works on extractive-summary generation employ some heuristics, but A few works indicate how to select the relevant features. These features are of two kinds: statistical – based on the frequency of some elements in the text; and linguistic – extracted from simplified argumentative structure of the text. We also present some computational results obtained with the application of our summarizer to some well known text databases, and we compare these results to some baseline summarization procedures.

Automatic text processing is a research field that is currently extremely active. One important task in this field is automatic summarization, which consists of reducing the size of a text while preserving its information content . A summarizer is a system that produces a condensed representation of its input for user consumption.

2.2. Implemented Text Rank Based Automatic Text Summarization Using Keyword Extraction

The Automatic text summarization is to transform lengthy documents into short versions, something which can be difficult and costly to undertake if done manually. For the summarization, the machine learning algorithm can be trained to comprehend documents and identify the sections that communicate important facts and information before producing the required summarised texts. For example, the image of the news article has been fed into a machine learning algorithm to produce a summary . In a specific way, there are two different approaches for text automatic summarization that are (a) extraction and (b) abstraction. The extractive summarization methods work by identifying important sections of the text and generating them verbatim; so they depend only on extraction of sentences from the original text while abstractive summarization methods target at producing important material in a new way. This can be explained as they interpret and examine the text using advanced natural language techniques in order to generate a new shorter text that conveys the most critical information from the original text.

2.2.1 Text Rank Based Algorithm

The algorithm Text Rank is a graph-based ranking algorithm for text processing which is used in order to find the most relevant sentences in text and also to find keywords . Some of the instructions for this algorithm are given below: In place of web pages, we use sentences. Similarity between any two sentences is used as an equivalent to the web page transition probability. The similarity scores are stored in a square matrix, similar to the matrix M used for Page Rank.

2.2.2 Page Rank Based Algorithm

The Text Rank is a kind of algorithm which is based on the Page Rank method that is often used in keyword extraction and text summarization. In this research study, we will help you to understand how Pagerank works with keyword extraction with examples and show the implementation using Python language. The PageRank is used to calculate the weight for the web pages. We take all web pages as a large directed graph. In this graph, a node is treated as a web page. If webpage 'A' has the link to web page 'B', it is represented as a directed edge from 'A' to 'B'.

3. Problem Definition

In the big data era, there has been an explosion in the amount of text data from a variety of sources. This volume of text is an inestimable source of information and knowledge which needs to be effectively summarised to be useful. There is an enormous amount of textual material, and it is only growing every single day. Think of the internet, consisting of web pages, news articles, status updates, blogs and so much more. The data is unstructured and the best that we can do to navigate it is to use search and skim the results.There is a great need to reduce much of this text data

to shorter, focused summaries that capture the salient details, both so we can navigate it more effectively as well as check whether the larger documents contain the information that we are looking for. So, In this project we use machine learning algorithms for generating the summary of a given huge file. We have used an extractive approach to achieve this. After generating the summary, Tha summary is converted into an audio clip that could be heard

4. Methodology

4.1 Proposed Methodology

Although some works in automated text summarization have been conducted using abstractive technique, the number of works is less than by using sentence extraction. The major issue of using sentence abstraction is the text processor has to understand the whole text and generate a summary as a human does. Although this technique can produce better summaries, however, this technique is very difficult to implement. On the other hand, sentence extraction is less difficult to implement, compared to sentence abstraction, but it is being conducted without understanding the context. In this paper, abstraction and sentence extraction techniques are combined. The steps involved in our proposed framework are as follows:

4.2 Sentence Segmentation

Sentence segmentation is the first step in automated text summarization. Normally, to parse a paragraph of text, a simple and limited way of dividing it into sentences would be to use '.' to obtain their ends. Extending this to '.', '!', and

'?' would handle more cases correctly. However, while this is a reasonable list of punctuation characters that can end sentences, this technique does not recognize the punctuation characters that appear in the middle of sentences. For example, a sentence “The book cost Mr. Ali $30.65.” has `.’ in the two places in a sentence where it does not mean the end of a sentence. In this methodology Split method is proposed to be used. Using the Split method on this input will result in an array with three elements, when we really want an array with only two. We can do this by treating each of the characters '.', '!', '?' as potential rather than definite end-of-sentence markers. Scan through the input text, and each time it comes to one of these characters, it needs a way of deciding whether or not it marks the end of a sentence. A set of predicates related to the possible end-of-sentence positions is generated. Various features, relating to the characters before and after the possible end-of-sentence markers, are used to generate this set of predicates.

4.3 Tokenization

After a paragraph has been segmented into sentences, each sentence will be tokenized. Tokenization is a process of breaking down a sentence into a list of words. In this work, a lexicon that consists of a dynamic knowledge that helps in parsing by providing phrases and words information to the parse engine is used.

4.4 Parts of Speech Tagging

Each token will be attached with its part of speech. For example, a noun will be attached to the word “city”. The challenging issues in assigning a part of speech to a word is, one word may have more than one part of speech. For example, the word “place” can belong to a verb or noun. To resolve the problem, disambiguation rules are created and used. Examples of the rules are; if a word (w) at i position is a preposition, then w at i+1 position is a noun and if a word (w) at i+1 position is a preposition then w at i position is belong to the verb.

4.5 Keyword Identification

Keywords will be extracted from the token list by using a dynamic lexicon. The dynamic lexicon is a lexicon that updates its contents automatically by adding new keywords from the paragraph that do not exist before. The words that can be categorised as keywords include title’ words, thematic words and emphasise words.

4.6 Relevant Term Identification

After a keyword has been identified, the remaining words in a sentence are defined as candidate words. Each candidate word will be ranked with a degree of relevance to the identified keyword. In ranking the candidate words, a fuzzy approach is applied, where each word will be ranked based on human common sense. For example if a word “university” is taken as a keyword, the word “lecturer” is possible to be very much relevant to the university, where we can say 0.9. However, the word department might be ranked less than the “lecturer” word, let say 0.4. The assumption is made that not all an entity university has an entity department. Thus the candidate words that have high degree ranks are considered as relevant words and the sentences that contain these words are possible to be extracted.

4.7 Calculate Probability Of Keywords and Relevant Term Occurrences In a Sentence

The probability theory is used to calculate the probability of keywords and its relevant words to occur in a sentence. Assume that “university” is a term to be considered as a keyword, and the relevant terms are “programs”, “lecturer”

and “students”. The frequent terms can be obtained by counting term frequencies. The probability of keyword P(k) term in a sentence is calculated as in the following equation P (k) = ¥k(s)/ ¥k(d) (1) where ¥ks represents the total frequencies of a keyword in a sentence and ¥k(d) is a total keyword in a document. The probability of relevant term P(R) is calculated as P(R) = ¥R(s) /¥R(d) (2) where ¥Rs represents the total frequencies of a relevant term in a sentence and ¥Rd is the total of frequencies in a document. The probability of a sentence P(s) can be formalised as P(s)= ℨ(K(i-n)) P(R(i-n)) (3) where is i=1, and n= a finite number. Ps of each sentence will be calculated and stored for further text processing usage

4.8 Sentence Extraction

The sentence extraction is conducted on a sentence that has a high probability value.

4.9 Sentence Refinement

Sentence refinement is conducted on the extracted sentences. In this stage, the understanding of the whole sentence will be conducted and the refinement is made by considering the context of the sentence. This step is conducted to ensure the selected sentences for a summary are precise and concise and unnecessary words are removed. 3.10 SUMMARY GENERATION The final step in automated text summarization is a summary generation. At this stage refined sentences are combined into a paragraph.

4.10 Converting Summary Into Audio Clip

After generating the summary, We produce an audio format of the summary. This audio can be heard if one doesn't want to read the summary. Inorder to convert the summary into audio we use a library that is available in python. There are several APIs available to convert text to speech in Python. One of such APIs is the Google Text to Speech API commonly known as the gTTS API. gTTS is a very easy to use tool which converts the text entered, into audio which can be saved as a mp3 file. The gTTS API supports several languages including English, Hindi, Tamil, French, German and many more. The speech can be delivered in any one of the two available audio speeds, fast or slow. However, as of the latest update, it is not possible to change the voice of the generated audio.

5. Result

5.1 Input File

Cricket is a bat-and-ball game played between two teams of eleven players each on a field at the centre of which is a 22-yard (20-metre) pitch with a wicket at each end, each comprising two bails balanced on three stumps. The game proceeds when a player on the fielding team, called the bowler, "bowls" (propels) the ball from one end of the pitch towards the wicket at the other end, with an "over" being completed once they have legally done so six times. The batting side has one player at each end of the pitch, with the player at the opposite end of the pitch from the bowler aiming to strike the ball with a bat. The batting side scores runs either when the ball reaches the boundary of the field, or when the two batters swap ends of the pitch, which results in one run. The fielding side's aim is to prevent run-scoring and dismiss each batter (so they are "out", and are said to have "lost their wicket"). Means of dismissal include being bowled, when the bowled ball hits the stumps and dislodges the bails, and by the fielding side either catching a hit ball before it touches the ground, or hitting a wicket with the ball before a batter can cross the crease line in front of the wicket to complete a run. 39 When ten batters have been dismissed, the innings ends and the teams swap roles. The game is adjudicated by two umpires, aided by a third umpire and match referee in international matches. Forms of cricket range from Twenty20, with each team batting for a single innings of 20 over and the game generally lasting three hours, to Test matches played over five days. Traditionally cricketers play in all-white kit, but in limited overs cricket they wear club or team colours. In addition to the basic kit, some players wear protective gear to prevent injury caused by the ball, which is a hard, solid spheroid made of compressed leather with a slightly raised sewn seam enclosing a cork core layered with tightly wound string. The earliest reference to cricket is in South East England in the mid-16th century. It spread globally with the expansion of the British Empire, with the first international matches in the second half of the 19th century. The game's governing body is the International Cricket Council (ICC), which has over 100 members, twelve of which are full members who play Test matches. The game's rules, the Laws of Cricket, are maintained by Marylebone Cricket Club (MCC) in London. The sport is followed primarily in South Asia, Australasia, the United Kingdom, southern Africa and the West Indies.

5.2 Output

The batting side has one player at each end of the pitch, with the player at the opposite end of the pitch from the bowler aiming to strike the ball with a bat. The batting side scores runs either when the ball reaches the boundary of the field, or when the two batters swap ends of the pitch, which results in one run. The game proceeds when a player on the fielding team, called the bowler, "bowls" (propels) the ball from one end of the pitch towards the wicket at the other end, with an "over" being completed once they have legally done so six times. Means of dismissal include being bowled, when the bowled ball hits the stumps and dislodges the bails, and by the fielding side either catching a hit ball before it touches the ground, or hitting a wicket with the ball before a batter can cross the crease line in front of the wicket to complete a run. It spread globally with the expansion of the British Empire, with the first international matches in the second half of the 19th century.

6. Conclusion

The increasing progression of the Internet has made a huge amount of information available. It is very difficult for humans to summarise huge amounts of text. So, there is an immense need for automatic summarization systems in this age of information excess. Due to the rapid growth of knowledge and use of the Internet, there is information overload. This problem can be solved, if there are robust text summarizers which produce a summary of documents to help users. Hence, there is a necessity to develop a system where a user can efficiently retrieve and get a summarised document. One potential solution is to summarise a document using either extractive or abstractive methods. The text summarization by extract is easier to build. In this project work, we emphasized various extractive approaches for single and document summarization. We have discussed the text rank algorithm most extensively used methods and machine learning approach. It delivers a good insight into recent trends and progresses in automatic summarization methods and describes the up-to-the-minute in this research area.

7. Future Scope

We focused on summarization of news articles belonging to sports and technical domain. One of the future scope may be to apply the topic-focused summarization framework to news articles or blogs. At present, the need of information is essential to the people from the different parts of the world. Hence the summarization should be done in multilingual manner. Also mentioning that there is availability of online lexical databases in other languages. And converting the summarization of particular language to its Audio is essential part. The Implemented system in this thesis can work as framework for the research community to understand and extend the applicability of cognitive and symbolic approach in various domains of business needs. Research in summarization continues to enhance the diversity and information richness, and strive to produce coherent and focused answers to users information need.

References

[1] Implementation of Automatic Text Summarization with TextRank Method in the Development of Al-Qur'an Vocabulary Encyclopaedia.
[2] Automatic text summarization of news articles https://ieeexplore.ieee.org/document/8336568.
[3] A Survey on NLP based Text Summarization for Summarising Product Reviews https://ieeexplore.ieee.org/document/9183355.
[4] Generating Extractive Document Summaries Using Weighted Undirected Graph and Page Rank Algorithm.
[5] Automatic text summarization: A comprehensive survey.

https://www.sciencedirect.com/science/article/abs/pii/S0957417420305030.
[6] Review of automatic text summarization techniques & methods. https://www.sciencedirect.com/science/article/pii/S1319157820303712
[7] https://www-sciencedirect-com-anits.knimbus.com/science/article/abs/pii/S09507051210 04974
[8] https://www.iieta.org/journals/isi/paper/10.18280/isi.260112
[9] https://search.yahoo.com/search?fr=mcafee&type=E211US826G0&p=pdf+summarization+in+python+github 49
[10] https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarizati on-using-deep-learning-python/
[11] https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an -overview-68ded5717a25
[12] Implemented Text Rank based Automatic Text Summarization using Keyword Extraction . https://www.irjiet.com/common_src/article_file/1605534448_c05213d47b_4_irjiet.pdf
[13] https://iq.opengenus.org/textrank-for-text-summarization/
[14] Study of automatic text summarization approaches in different languages. https://link.springer.com/article/10.1007/s10462-021-09964-4
[15] Automatic Text Summarization Using a Machine Learning Approach. https://www.researchgate.net/publication/220974615_Automatic_Text_Summarization_Using_a_Machine_Learning_Approach
16.https://en.wikipedia.org/wiki/Automatic_summarization

Author Profile

Tanooja Kosuru, currently pursuing my bachelors degree from Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam. In Department of CSE. Having interest in python and Machine learning, worked on this, project Automatic Text and Machine learning. This project has helped me in enhancing my knowledge. Looking forward to work on more such projects in this domain.

Tejasri Karri, pursuing my bachelors in Department of CSE at Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam. I have learnt Machine learning basics using Python and found it interesting and utilized this to develop my team project Automatic Text summarization and audio Generatin. I am looking forward to research in the Machine learning domains to gain more knowledge and expand my career opportunities.

Akhilesh Tannedi, is currently pursuing final year of B. Tech in Computer Science stream in Anil Neerukonda Institute of Technology and Sciences. Author is good at programming. Analytical problem solving skills. Author has learnt C, C++, Python 3, Java, Html5, CSS, JavaScript with React and Node libraries. Learnt Machine Learning basics using Python. Author has solved over 300 coding problems using C++, Python as main preference. And currently working to develop the tem project on Automatic Text Summerization and Audio Generation.

Prasanna Kavya Matham is the member of project in Automatic Text Summerization and and Audio Generation. Author has graduated from Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam in Computer Science. Author has good enthusiasm which makes project/ Work wealthy on project doing in Machine learning and project is Automatic Text Summerization and Audio Generation