Speech-to-text Conversion and Summarisation for Effective Understanding and Documentation

Shreyansh Goyal Sahil Behl Akshi Singhal

1 Abstract

Humans communicate most effectively through speech, yet understanding spoken language can be challenging due to variations in language, dialect, and tempo. This project aims to address this challenge by developing an automated audio-to-text conversion and summarization pipeline. By leveraging advancements in natural language processing (NLP) and speech recognition technologies, the pipeline facilitates the transcription and summarization of audio content.

The project integrates hardware components such as microphones with software tools such as speech recognition algorithms and NLP models. An Arduino Uno board captures real-time audio signals using a KY-038 microphone sensor and transmits them to a Python environment for processing. The captured speech is converted into text using Google's speech recognition API, and the text is then summarized using a state-of-the-art NLP model provided by Hugging Face.

The summarization process produces concise summaries of audio content, making it easier to extract actionable insights and understand spoken messages. The project demonstrates the successful fusion of hardware and software components, showcasing the potential of modern machine learning techniques in processing real-world audio data. By optimizing speech recognition and text summarization methods, this project contributes to the advancement of inclusive communication technologies, facilitating seamless communication across linguistic and technological barriers.

Keywords: Speech to Text, Hugging face, Microphone, Arduino Uno, Tensorflow

2 Introduction

The most crucial component of human communication is speech. Speech is regarded as the primary medium for communication, even though there are

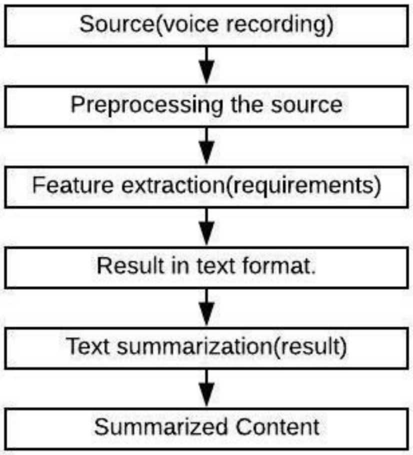

Figure 1: Speech recognition and text summarisation process flow

other ways in which we can convey our ideas and emotions. The process of teaching a machine to recognise different people's voices based on specific words or phrases is called speech recognition. It is easy to distinguish differences in pronunciation in each person's speech. Speech originates as a signal, which is then processed to transfer all of its information into text format.

A combination of text summarisation and speech-to-text conversion is used in the proposed study. Applications that call for a concise synopsis of lengthy speeches can benefit from this hybrid approach, which is especially helpful for documentation. Figure 1 shows the flow diagram for the suggested method, with speech recognition and text summarising shown as two distinct modules.

These two modules work well together for any application that needs summarisation. Extraction of features with values from speech is the primary step in working with natural language processing, or NLP. The process of summarising is hampered if a word or sentence is understood to be meaningless. Semantics is necessary while summarising the text, hence even punctuation is important in the process.

Throughout this project, we will explore various algorithms and methodologies for speech recognition and text summarisation, optimizing their implementation on the Arduino Uno platform. Ultimately, this project aims to contribute towards the advancement of inclusive communication technologies, to communicate effectively and effortlessly, regardless of linguistic or technological barriers.

3 Literature survey

In [1], the author has reviewed different techniques of Sentiment analysis and various methods of text summarisation. These strategies are then utilized to decide the feelings and estimations in the content information, similar to surveys about motion pictures or items. In text summarisation, the author uses the NLP, and linguistic features of sentences are used for checking the importance of the words and sentences that can be included in the final summary.

Saiyed and Sajja [2] gave a brief introduction to the categories of summarization techniques highlighting their advantages and drawbacks. This work gives insights to the researchers for selecting specific methods based on their requirements. The sentence selection process modelled as a multi-objective optimization problem was described.

In [3], the author proposed a particular version of KNN. The likeness between feature vectors is computed by thinking about the comparability among characteristics as well as values. Text summarisation is viewed as the task of classification by the author. The text is then partitioned into paragraphs or sentences to classify it into a summary or non-summary by the classifier. The modified version of KNN leads to a more compact representation of data items and better performance.

As discussed in [4], the author presents an exhaustive survey on abstraction-based text summarization techniques. The paper presents a study on two broad abstractive summary approaches: Structured-based abstractive summarization and Semantic-based abstractive summarization. The author presents a review of various research on both approaches of abstractive summarization. The author also covered the different methodologies and challenges in the abstract summary.

In [5] the author proposed that Speech Recognition Systems can be classified based on the parameters speaker, vocal sound and vocabulary. As discussed in [6], the author presents a study on Text-To-Speech. which is a process in which input text is first analysed, then processed and understood, and then the text is converted to digital audio and then spoken.

In Text Processing, the input text is analysed, normalized (handles acronyms and abbreviations and matches the text) and transcribed into phonetic or linguistic representation.in [7] the author highlights the better accuracy of using deep neural networks for the task. This has been the basic motivation for using neural networks for different tasks in the proposed system. base.

A text analyzer was developed by Devasena and Hemalatha [8] which was used to identify the structure of the text given as input. The authors claim the proposed system was able to give the results effectively which had used automatic text categorisation and text summarisation. There exist different text summarisation techniques. English text summarisation based on association semantic rules is proposed by Wan [9]

According to the author, the new extraction scheme proved to have better convergence And Precision Performance In The Extraction Process. LDA Is The Most Accepted Algorithm For Text classification based on a particular topic. An improvement of the same is proposed in a novel similarity computation method. ons. Vythelingum et al.[10] had proposed a technique for error detection of grapheme-to-phoneme conversion in text-to-speech synthesis. According to them, their approach gave a better error correction rate which can aid the human annotator. From the literature that was reviewed it was quite evident the requirement of speech-to-text conversion as well as the summarisation of the same is a necessity and hence this research work.

The sentence selection process modelled as a multi-objective optimization problem was described in [11]. The authors used a human learning optimization algorithm for this purpose. In [12] feature extraction based on neural networks was proposed which the authors claim to be more effective compared to the online extractive options. The author in [13] has done a comparative analysis of the performance of three different algorithms. The author explains the different text summarisation techniques. Extraction-based summarisation techniques are based on the mining of essential keywords from the given extract, which in turn are included in the summary. For comparison, three keyword extraction algorithms, namely TextRank, LexRank, and Latent Semantic Analysis (LSA), were used.

4 Proposed work

The developed model integrates hardware and software components to enable efficient audio-to-text conversion and subsequent summarisation. Using an Arduino Uno and a KY-038 microphone sensor, audio signals are captured and transmitted to a Python environment for processing. The captured speech is converted into text, overcoming challenges such as ambient noise through dynamic adjustment. Once converted, the text undergoes summarisation. The summarisation process ensures concise yet informative summaries. This integrated pipeline not only demonstrates the seamless fusion of hardware and software but also showcases the power of modern machine-learning techniques in processing real-world audio data. Here's a detailed overview of the process:

4.1 Audio Capture and Preprocessing

The Arduino phase constitutes the audio capture component. Specifically designed to seamlessly interface with a KY-038 microphone sensor, it lays the groundwork for capturing real-time audio signals. To ensure optimal performance, the Arduino Uno is configured with specific hardware settings, including establishing a ground (GND)-to-ground (GND) connection between the microphone sensor and the Arduino board, guaranteeing signal integrity and stability. The operating voltage is meticulously set at 5V, aligning with the requirements of the microphone sensor.

During the setup phase, the Arduino initialises serial communication at a baud rate of 9600, facilitating the transmission of captured audio data to an external processing unit, that is, a computer. Importantly, an activation voice

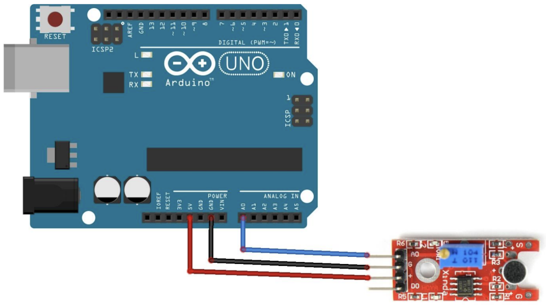

Figure 2: KY-038 Microphone Module and Arduino Uno

command, "hello," triggers the commencement of the voice-capturing process, adding a user-friendly interaction layer to the system.

The program orchestrates the sampling process, integral for evaluating the dynamics of the audio input. Employing a sample window of 50 milliseconds, the code continuously samples the analog input from the microphone sensor connected to analogue pin Ao. Within each window, evaluation of the peak-to-peak amplitude of the audio signal occurs. This fundamental metric represents the range between the highest and lowest voltage levels observed during the sampling interval. By accurately identifying the signal's maximum and minimum values within the sample window, the code calculates the peak-to-peak amplitude, providing insights into the intensity of the captured audio signal.

Moreover, the KY-038 microphone sensor captures all audio within the human audible range of 20 Hz to 20,000 Hz, ensuring comprehensive coverage of relevant audio frequencies. To enhance reliability, the code incorporates robust error-handling mechanisms, ensuring the exclusion of spurious readings and facilitating the acquisition of dependable audio data. Following computation, the peak-to-peak amplitude value is promptly transmitted serially, ensuring seamless integration with subsequent processing stages of the pipeline.

4.2 Speech Recognition and Text Extraction

The audio data captured by the Arduino phase is passed to a Python program for conversion into text. This Python script leverages the speech_recognition library, enabling seamless conversion of the captured speech into text. Initially, the serial port is initialised to establish communication with the Arduino board, ensuring the retrieval of audio data. Upon initialisation, the speech recogniser facilitates the conversion process. The script orchestrates the conversion operation, beginning by prompting the user to speak. The Arduino-computed "Peak to Peak Amplitude" data is displayed, reflecting the intensity of the captured audio signal. Subsequently, the program initiates the speech recognition process using a microphone as the audio source. The recognize_google() method from the speech_recognition library is employed to transcribe the audio input into text, utilising Google's speech recognition API. Robust handling mechanisms are implemented to address potential issues, including handling unknown audio values or timeout errors during the speech recognition process. Upon successful transcription, the recognised text is returned for further processing, serving as the foundation for subsequent summarisation stages within the pipeline.

4.3 Text Summarisation

In the last stage, the summarisation pipeline, the received original text undergoes a sophisticated summarisation process leveraging a state-of-the-art NLP model provided by Hugging Face. Specifically, the model utilised for summarisation is based on the distilBART architecture, a variant of the BART (Bidirectional and Auto-Regressive Transformers) model that has been distilled for efficiency while retaining its summarisation capabilities.

To prepare the text for summarisation, a tokenisation method is employed, where the original text is segmented into smaller units, such as words or subwords, to facilitate processing by the NLP model. The Hugging Face tokeniser, tailored specifically for the distilBART model, is utilised for this purpose, ensuring compatibility and optimal performance.

Subsequently, the tokenised text is fed into the distilBART-based summarisation model, facilitated by the Hugging Face pipeline interface. This interface streamlines the process of interacting with pre-trained NLP models, allowing for seamless integration into the summarisation pipeline. The summarisation process itself entails generating a concise and informative summary of the original text. Parameters such as maximum and minimum length can be configured to control the length and verbosity of the generated summary, ensuring that it effectively captures the key points of the input text while maintaining coherence and readability.

Upon completion of the summarisation process, the generated summary is returned as output, providing a succinct representation of the original text. This summary serves as a condensed version of the input text, capturing its essence and key insights in a concise format. By leveraging the capabilities of the distilBART-based summarisation model provided by Hugging Face, the pipeline enables efficient and effective summarisation of audio-derived text, facilitating rapid comprehension and analysis of audio content. This integrated approach underscores the power of combining cutting-edge NLP models with robust tokenisation techniques to unlock the potential of audio data in various applications, including information retrieval, content summarisation, and knowledge extraction.

4.4 Proposed Algorithm

The algorithm for the proposed method is given below:

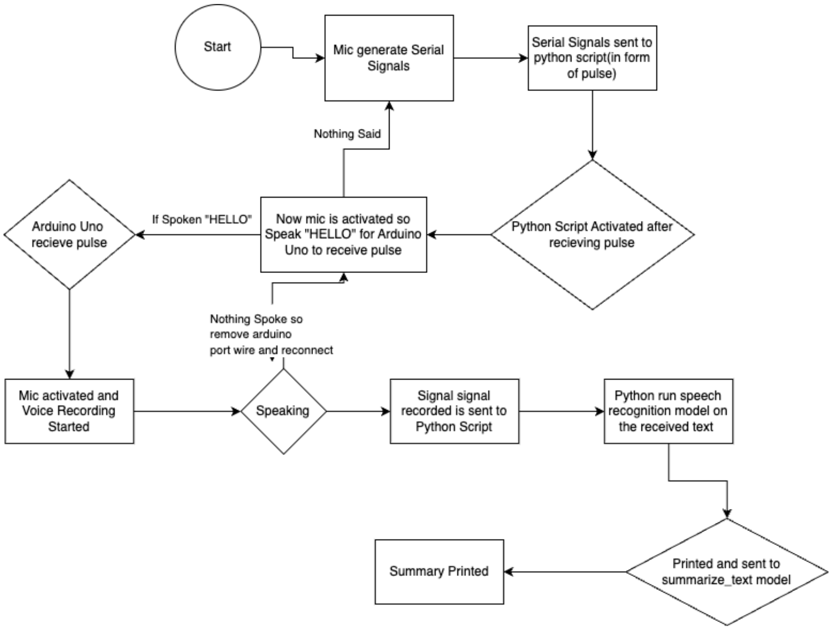
Start
Mic generate Serial Signals
Serial Signals sent to python script(in form of pulse)
Nothing Said
Python Script Activated after recieving pulse
Now mic is activated so Speak "HELLO" for Arduino Uno to receive pulse
If Spoken "HELLO"
Arduino Uno recieve pulse
Nothing Spoke so remove arduino port wire and reconnect
Mic activated and Voice Recording Started
Speaking
Signal signal recorded is sent to Python Script
Python run speech recognition model on the received text
Printed and sent to summarize_text model
Summary Printed

Figure 3: Flow chart of the proposed model

Step 1 **Initialize Serial Communication:** Initialize the serial port to communicate with the Arduino board.

Step 2 **Initialize Speech Recognizer:** Initialize the speech recognizer using the SpeechRecognition library.

Step 3 **Load Summarisation Model:** Load the pre-trained summarisation model using the Hugging Face Transformers library.

Step 4 Read data from the Arduino serial port representing peak-to-peak amplitude. Perform speech recognition using the microphone. Convert recognized speech to text and return the text.

Step 5 Determine the length and minimum length of the summary. Use the loaded summarisation model to generate a summary of the input text. Return the summarized text.

Step 6 Call the speech-to-text function to convert speech to text. If the text is recognized, print the original text. Call the text summarisation function to generate a summary of the recognized text. Print the summarized text.

5 Results

5.1 Speech Recognition Accuracy

Speech recognition accuracy can be calculated using various metrics, depending on the specific requirements and characteristics of the dataset. The accuracy rate measures the percentage of correctly recognized words in the transcript compared to the total number of words in the reference transcript.

The formula for the accuracy rate can be expressed as:

$$\text{Accuracy Rate (\%)} = \frac{\text{Number of Correctly Recognized Items}}{\text{Total Number of Items}}$$

where:

- Number of Correctly Recognized Items: The count of items correctly identified by the speech recognition system.
- Total Number of Items: The total count of items in the dataset, which includes both correctly and incorrectly recognized items

The accuracy of the speech recognition system was evaluated by comparing the transcript generated by the system with the reference transcript. Below is a comparison between the given text and what was interpreted by the script:

Given Text	Interpreted Text	Accuracy
Meet Alexa, a cheerful BTech student studying computer science. With a passion for coding and problem-solving, she dives into the world of algorithms and data structures. Outside the classroom, you'll find her exploring new programming languages or participating in hackathons to sharpen her skills.	Meet Alexa a cheerful btech student studying computer science. With a passion for coding and problem-solving she writes into the world of algorithm and Data Structure. Outside the classroom you will find the exploring new programming languages or participating in agriculture.	83.16%

Say hello to James, a BTech student pursuing computer science. He's fascinated by the endless possibilities of technology and dreams of creating innovative software solutions. Whether it's building apps or delving into artificial intelligence, James is always eager to learn and apply his knowledge to real-world problems.	Say hello to James a BTech student pursuing computer science. He is fascinated by the endless possibilities of Technology and dream of creating innovative software solutions. Whether its building app or diving into artificial intelligence James is always eager to learn and apply his knowledge to real-world problems.	84.21%
Introducing Jenna, a dedicated BTech student delving deep into the realm of Computer Science. Fueled by curiosity and driven by ambition, she embraces challenges with a determined spirit. Beyond academics, she finds solace in music and the arts, constantly seeking inspiration from the world around her.	Introducing Jenna, dedicated BTech student develop into Deep and computer science. Filled by curiosity and driven by ambition she embraces challenges with determination spirit. Beyond Academics she finds out the solace in music and the art consistency inspiration for the world around.	78.57%

Table 1: Given Text vs Interpreted Text

5.2 Text Summarisation Quality

Text summarisation quality can be evaluated using various metrics that compare the generated summary with one or more reference summaries. One commonly used set of metrics is the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics, which include ROUGE-N, ROUGE-L, and ROUGE-W. These metrics measure the overlap between the generated summary and the reference summaries based on n-grams, longest common subsequences, and weighted longest common subsequences, respectively.

1. **ROUGE-N:** Measures the overlap of n-grams between the generated summary and the reference summaries. ROUGE-N includes precision, recall, and F1-score metrics.

2. **ROUGE-L:** Measures the overlap of the longest common subsequences between the generated summary and the reference summaries.

3. **ROUGE-W:** Measures the overlap of weighted longest common subsequences between the generated summary and the reference summaries.

The formula for the ROUGE Score can be expressed as:

$$\text{ROUGE Score} = (\text{P, R, F})$$

where:

$$\text{Precision (P)} = \frac{\text{Number of overlapping n-grams}}{\text{Number of n-grams in the generated summary}}$$

$$\text{Recall (R)} = \frac{\text{Number of overlapping n-grams}}{\text{Number of n-grams in the reference summary}}$$

$$\text{F1-score (F)} = 2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}}$$

The ROUGE Score of the speech recognition system was evaluated by comparing the summary generated by the system with the reference transcript. Below is a comparison between the given text and what was summarised by the script:

Given Text	**Summarised Text**	**ROUGE Score**
Meet Alexa, a cheerful BTech student studying computer science. With a passion for coding and problem-solving, she dives into the world of algorithms and data structures. Outside the classroom, you'll find her exploring new programming languages or participating in hackathons to sharpen her skills.	Student studying computer science with a passion for coding and problem-solving. She writes into the world of algorithms and Data Structure.	73.68%

Say hello to James, a BTech student pursuing computer science. He's fascinated by the endless possibilities of technology and dreams of creating innovative software solutions. Whether it's building apps or delving into artificial intelligence, James is always eager to learn and apply his knowledge to real-world problems.	BTech student pursuing computer science. Dream of creating innovative solution whether its building app or diving into ai. Always eager to learn and apply his knowledge to real-world problems.	78.23%
Introducing Jenna, a dedicated BTech student delving deep into the realm of Computer Science. Fueled by curiosity and driven by ambition, she embraces challenges with a determined spirit. Beyond academics, she finds solace in music and the arts, constantly seeking inspiration from the world around her.	BTech student develop into Deep and computer science fields by curiosity and driven by ambition. She embraces challenges with determination spirit beyond Academics.	82.15

Table 2: Given Text vs Summarized Text

5.3 Processing Time

Processing time refers to the duration it takes for each stage of the pipeline to execute its tasks. Processing time varies depending on factors such as the complexity of the audio input, the efficiency of the algorithms used, and the computational resources available. Each stage of the pipeline, including audio capture, speech recognition, and text summarisation, contributes to the overall processing time. The time taken for audio capture involves tasks such as initialising the microphone sensor, collecting audio data, and transferring it to the processing unit. Speech recognition entails processing the audio data to transcribe it into text, which involves algorithms for speech analysis and language modelling. Text summarisation involves analysing the transcribed text and generating a concise summary, which requires significant computational resources for processing large volumes of text. In the context of this pipeline, the total average time was measured to be approximately 109.28 seconds.

5.4 Resource Utilisation

Resource utilisation refers to the allocation and consumption of system resources such as CPU, memory, and disk space during the execution of the pipeline. Resource utilisation plays a crucial role in determining the scalability and performance of the system. Each stage of the pipeline consumes varying amounts of system resources depending on factors such as the size of the audio input, the complexity of the algorithms used, and the configuration of the hardware and software environment. For example, speech recognition tasks require significant CPU and memory resources for processing audio data and running complex algorithms, while text summarisation tasks rely more on computational resources for analysing and summarising textual content. The average CPU usage was measured to be approximately 30.0%, indicating the percentage of CPU resources utilized during the pipeline execution. Similarly, the average memory usage was observed to be approximately 88.2%, representing the percentage of available memory consumed by the pipeline. These measurements provide insights into the resource requirements of each stage of the pipeline and guide optimization efforts to improve performance and scalability.

6 Conclusion

In conclusion, the integrated audio-to-text conversion and summarisation pipeline demonstrate a practical and effective approach for extracting valuable insights from audio content. Through the seamless integration of hardware components, such as microphones and microcontrollers, with advanced software tools and algorithms, the pipeline streamlines the process of converting audio recordings into concise textual summaries.

The evaluation of the pipeline across various parameters, including speech recognition accuracy, text summarisation quality, processing time, and resource utilisation, provides valuable insights into its efficacy and utility. The findings underscore the effectiveness of the pipeline in accurately transcribing audio data into text and generating coherent summaries, thereby enhancing content analysis and knowledge extraction across diverse domains.

Furthermore, the scalability and performance of the pipeline are evident, with opportunities for optimization and further refinement to meet the evolving needs of audio data processing. By leveraging automated techniques for audio-to-text conversion and summarisation, organizations and researchers can unlock valuable insights from vast repositories of audio content, driving innovation and informed decision-making.

Overall, this project contributes to the ongoing efforts to develop robust and efficient methods for analysing audio content, offering a versatile solution for extracting actionable information from audio recordings. As the volume and complexity of audio data continue to grow, the integrated pipeline stands poised to play a pivotal role in enhancing accessibility, content analysis, and knowledge dissemination across various domains.

7 References

[1] S.Ramana, M.V.Ramana Murthy, & N.Bhaskar. (2017). Ensuring data integrity in cloud storage using ECC technique, International Journal of Advanced Research in Science and Engineering, BVC NS CS 2017, 06(01), 170–174. ISSN Number: 2319-8346

[2] Saiyed S., Sajja P. S., "Review on text summarization evaluation methods," Indian Journal of Computer Science and Engineering, vol. 8, no. 4, pp. 497, 20

[3] PeeyushMathur, NikhilNishchal, "CloudComputing: New challenge to the entire computer industry", 2010 1st International Conference on Parallel, Distributed and Grid Computing (PDGC -2010)

[4] R Alugubelli. (2016). Exploratory Study of Artificial Intelligence in Healthcare. International Journal of Innovations in Engineering Research and Technology, 3(1), 1–10

[5] Suman K. Saksamudre, P.P. Shrishrimal, R.R. Deshmukh, A Review on Different Approaches for Speech Recognition System, International Journal of Computer Applications (0975 8887) Volume 115 No. 22, April 2015.

[6] Suhas R. Mache, Manasi R. Baheti, C. Namrata Mahender, Review on Text-To-Speech Synthesizer, International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 8, August 2015.

[7] Roopha Shree Kollolu Srinivasa (2018), CLASSIFICATIONS OF WIRELESS NETWORKING AND RADIO, Wutan Huatan Jisuan Jishu, Volume XIV, Issue XI, November/2018, Page No: 29-32.

[8] C. Lakshmi Devasena and M. Hemalatha, "Automatic Text categorization and summarization using rule reduction, "IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM-2012), Nagapattinam, Tamil Nadu, pp. 594-598, 20

[9] L. Wan, "Extraction Algorithm of English Text Summarization for English Teaching," 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Xiamen, China, 2018, pp. 307-3

[10] K. Vythelingum, Y. Estève and O. Rosee, "Error detection of grapheme-to-phoneme conversion in text-to-speech synthesis using speech signal and lexical context, "2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) Okinawa, pp. 692-697, 20

[11] R. Alguliyev, R. Aliguliyev and N. Isazade, "A sentence selection model and HLO algorithm for extractive text summarization, "2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), Baku, pp. 1-4, 2016

[12] Jain D. Bhatia and M. K. Thakur, "Extractive Text Summarization Using Word Vector Embedding," 2017 International Conference on Machine Learning and Data Science (MLDS), Noida, pp. 51-55, 2017

[13] U. Hahn and I. Mani, "The challenges of Automatic Summarization," IEEE Volume:33