International Journal of Electrical and Computer Engineering (IJECE)
Vol. 9, No. 5, October 2019, pp. 3642~3648
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i5.pp3642-3648

Speech to text conversion and summarization for effective understanding and documentation

Vinnarasu A., Deepa V. Jose
Department of Computer Science, CHRIST (Deemed to be University), India

Article Info

Article history:

Received Jan 17, 2019
Revised Apr 1, 2019
Accepted Apr 10, 2019

Keywords:

Feature extraction
Natural language processing
Natural language toolkit
Speech recognition
Text summarization

ABSTRACT

Speech, is the most powerful way of communication with which human beings express their thoughts and feelings through different languages. The features of speech differs with each language. However, even while communicating in the same language, the pace and the dialect varies with each person. This creates difficulty in understanding the conveyed message for some people. Sometimes lengthy speeches are also quite difficult to follow due to reasons such as different pronunciation, pace and so on. Speech recognition which is an inter disciplinary field of computational linguistics aids in developing technologies that empowers the recognition and translation of speech into text. Text summarization extracts the utmost important information from a source which is a text and provides the adequate summary of the same. The research work presented in this paper describes an easy and effective method for speech recognition. The speech is converted to the corresponding text and produces summarized text. This has various applications like lecture notes creation, summarizing catalogues for lengthy documents and so on. Extensive experimentation is performed to validate the efficiency of the proposed method.

Corresponding Author:

Vinnarasu A.,
Department of Computer Science,
CHRIST (Deemed to be University),
Bengaluru, Karnataka, India.
Email: vinnarasu.a@cs.christuniversity.in

1. INTRODUCTION

Speech is the most important part of communication between human beings. Though there are different means to express our thoughts and feeling, speech is considered as the main meduim for communication. Speech recognition is the process of making a machine recognize the speech of diffrent people based on certain words or phrases. Variations in the pronunciation are quite evident in each individual's speech. The original form of the speech is a signal, and a signal is processed such that all the information present in the signal is converted in to the text format. The feature extraction is the process of taking a signal and converting it to the required format with certain logic. Even though speech is the easiest way of communication, there exist some problems with speech recognition like the fluency, pronunciation, broken words, stuttering issues etc. All these have to be addressed while processing a speech. Text summarization is one of the major concepts used in the field of documentation. Lengthy documents are difficult to read and understand as it consumes lot of time. Text summarisation solves this problem by providing a shortened summary of it with semantics.

In the proposed work a combination of speech to text conversion and text summarisation is implemented. This hybrid method will aid applications that require brief summary of lengthy speeches which is quite useful for documentation. The flow diagram of the proposed approach is mentioned in Figure 1, in which the speech recognition and text summarization is given as two different modules. The combination

of these two modules aids any application in which summarization is required. The first and foremost step to work with NLP (Natural Language Processing) is to extract the features from the speech which has some values. If a word or a sentence is recognized as meaningless, then it becomes an obstacle to summarization process. Even the punctuation plays a vital role in summarization as semantics is important while summarising the text.

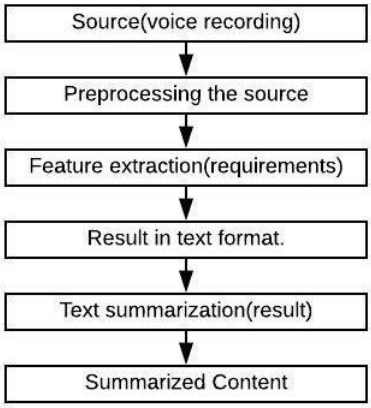

Figure 1. Speech recognition and text summarisation process flow

2. RELATED WORKS

Speech to text conversion finds applications in various scenarios. An effective method to gain fluency in English language that enhances the user's way of speech through correctness of pronunciation following the English phonetics was developed by Jose et al. [1]. A comparative analysis mentioning the benefits and demerits of the various sizes of vocabulary speech recognition systems was done by Sivakumar et al. [2]. This work demonstrated the role of language model in improving the accuracy of speech to text conversion with different scenarios with noises and broken words.

Yogita et al. [3] presented a multilingual speech-to text conversion system using Mel-Frequency Cepstral Coefficient (MFCC) feature extraction technique and Minimum Distance Classifier, Support Vector Machine (SVM) methods for speech classification. In [4] a model to convert natural Bengali language to text was proposed which used open source framework Sphinx 4. Authors claim an average of 71.7% accuracy for their approach in the tested dataset. English text summarisation based on association semantic rules is proposed by Wan [5].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.

Saiyed and Sajja [6] gave an brief introduction to the categories of summarization techniques highlighting their advantages and drawbacks. This works gives insights to the researchers for selecting specific methods based on their requirement. The sentence selection process modelled as a multi-objective optimization problem was described in [7]. The authors used human learning optimization algorithm for this purpose. In [8] feature extraction based on neural networks was proposed which the authors claim to be more effective compared to the online extractive options. Vythelingum et al.[9] had proposed a technique for error detection of grapheme to-phoneme conversion in text-to-speech synthesis. According to them their approach gave 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 summarization of the same is a necessity and hence this research work. Zenkert at el. [10] introduced a cross-dimensional text summarization which uses the concept of dimensional selection and filtering. The method was experimented using the results of Multidimensional knowledge representation database. A text analyzer was developed by Devasena and Hemalatha [11] which was used to identify the structure of the text given as input.The authors claims the proposed system was able to give the results effectively which had used the automatic text categorisation and text summarisation . There exists different text summarisation techniques.a detailed overview of the same is proposed in[12] by Rahimi et al. A similar study was done by Dalal and Malik also [13].

A modified approach of K Nearest Neighbor for achieving text summarization was done by Jo [14]. The author focussed more on the reliability aspect. A Vietnamese language based text summarization approach with three stages using graphs was proposed by Tran and Nguyen in [15].

The authors claims that the proposed approach was able to gather more meaningful text relevant to native speakers. Vimalaksha et al. [16] provided a method to summarize the video so as to same time and space as well as helps in archiving. An overview of text summarisation focussing more on the techniques to avoid redundancy was done in [17]. Matsubayashi et al. [18] proposed a system for effective text retrieval based on the query. The authors used automatic text summarisation approach for the same. The rest of the paper is organised as follows. Section 3 gives the details of the proposed model; Section 4 mentions the results obtained followed by Conclusions and Future scope in Section 5.

3. PROPOSED MODEL

3.1. Experimental setup

The speech from the source is recorded using a microphone and the feature is extracted in text format using Google Application Programming Interface (API). However, the text extracted using the Google API does not include period (.) at the end of the sentence. This can lead to confusion in the termination of the statements. In order to avoid this, in the proposed approach a custom code has been written to provide a period after a pause of 2e+6 μs or more. This makes the sentence clearer and it is pre-processed to add period (.) and question mark (?). In order to proceed with the concept of adding a period to the extracted text, 2e+6 μs has been considered as the minimum pause time. If there is a pause for more than the said time also, the system will wait for the user input due to validation.

Since period plays a vital role in the completion of a sentence, a new sentence will be started with the concept of conjunction in the absence of period. This problem is eliminated in the proposed model by the use of temporary storage. Therefore, whenever there is a pause, the period will be added to the text and will be temporarily stored in the temporary variable. If the next sentence begins with a conjunction, the temporary variable will be cleared and the sentence will be appended to previous sentence using conjunction. Conversely to the conjunction, if the sentence begins with a subject then the temporary variable value is used and the period will be appended to the sentence. Wh-questions are expected to end with a question mark(?). Hence, whenever the sentence begins with the wh-statement, the temporary variable will hold (?) on a pause of 2e+6 or more. In case the next sentence begins with the question tag statement then the value in the temporary variable will not be used. If the sentence begins with a new subject then question mark will be appended to the end of the sentence.

The proposed method summarizes the extracted text according to the rank of the sentences which can be determined through the frequency of occurence of words. The sentence tokenize and word tokenize techniques from the packages of python NLTK are used to find the frequeny of words. When the text is extracted from the input using Google API, the sentences and words in the text are obtained using sentence tokenize and word tokenize respectively. The input given by the user as speech will be converted to signal. And the signals will be converted to text format in collaboration with the Google API. In order to process the generated text with the proposed model, word tokenize and sentence tokenize is used. The complete set of a sentence is given as inputs to the sentence tokenize, every sentence is separated with the occurrence of the dot. All the sentences are given as inputs to the word tokenize, each word is separated with the occurence of the space.

When a text with proper format is used for summarising, it is less complex to process as it is in the exact format and is often precise and clear. But this is not the case when a speech is taken as the input. Here the speech has to be converted to text and then it should be summarized. The problems to be tackled here are the occurrence of repeated words, broken words, different dialect and synonyms used to convey the message etc. Therefore, to overcome such problems, the words with less importance is eliminated. For this, a minimum and maximum range is set for the occurrence of any specific word. Even though the sentence and word frequency are used, to find the important sentence in the whole content, a ranking model is applied.

Finally, after the words are tokenized, the frequency of every word is calculated by the by the summarization algorithm in proposed model. The weight of the sentence is found with the consideration of the frequency of words. The index is ranked according to the weight of the sentence and with the identification of the index, the sentence is summarized. Python nlargest function is used to rank the sentence based on the weight of the sentence. The text will be summarized based on the weight of the sentences. The flow chart of implementation procedure of the proposed model is as shown in Figure 2.

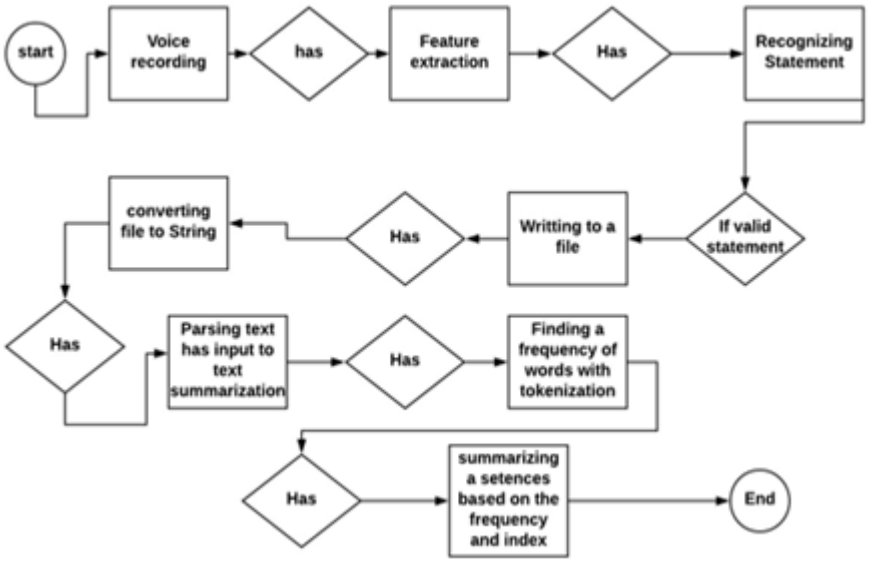

Figure 2. Flow chart of the proposed model.

3.2. Proposed procedure

The algorithm for the proposed method is given below.

Step 1 − START
Step 2 − declare microphone as a source
Step 3 − declare three lists audioRecorded, textFormatOfRecord, temproaryList
Step 4 -- while sentence! = exit exit
audioRecorded = listen(source)
extractedText = recognize_google(audioRecorded)

If (pause && next sentence of extractedText starts with subject)
sentence=”.”+sentence
else if(pause && next sentence of extractedText starts with conjunction)
sentence=”,”+sentence
end while
[The while loop is exited with text “exit exit”]

Step 5 − declare webSpoken as file
webSpoken=sentence
Step 6− declare two lists sentences, words
Step 7 - sentences=sent_tokenizer(webspoken)
words =word_tokenize(sentences)
compute_frequencies(words)

Step 8 – initalize minCut=0.1 and maxCut=0.9
If word frequency>maxCut and frequency<minCut
remove the word
While ranking index! (sorted)
ranking=nlargest(Sorted list of sentence)

Step 9 – Print the sentence in the order of ranking
Step 10 – STOP

4. RESULTS

The recorded speech can be converted to text with the help of Google API. It is difficult to separate the text into sentence which is generated using Google API, because the extrated text does not have a period(.). To make the sentences distinct, in the proposed model, a period is appended at the end of

the sentence when there is a pause. If the sentence is a wh-sentence, a question mark(?) is appended to the end of the sentence. This makes it easier to tokenize the sentences, as python string tokenization uses period to differentiate sentences. If the sentence has a pause and if it begins another sentence with the conjunction, a comma(,) is appended to the end of the sentence. This makes it easier to tokenize the sentences, as python string tokenization uses period to differentiate sentences. The proposed model considers the punctuations ('.', ',' and '?') in the recognized text. The proposed model recognition is faster when compared to the basic model (sentences without '.', ',' and '?') recognition. The basic model summarizes the recognized text without any pre-processing. But in the proposed approach, pre-processing is used to add a period(.) at the end of each sentence to indicate the termination of a sentence. In python sentence tokenization, sentences are tokenized based on the presence of period. Though there are many punctuation marks that can be included in a sentence, the focus in the proposed model is only on period and question mark. Table 1 shows the time taken to recognize sentences with and without period and question mark respectively. Based on the recogniztion time, we can say that the senteces which includes period and question mark are recognized faster than the sentences without it.

Table 1. Recognition time for sentence with and without (.) and (?)

Recognized sentence	Recognition time for sentence without appending (.) or (?) (in µs)	Recognition time for sentence appending (.) or (?) (in µs)
Technology solves problems and in turn creates problems	360996	183981
Standards are always out of state that is what makes them standard	98255	452107
The greatest enemy of knowledge is not ignorance it is the illusion of knowledge	240178	110367
We have to stop optimizing for programmers and start optimizing for users	396658	78134
Low level programming is not that easy for beginners	66141	108326

The graph for time taken to recognize sentences with and without period(.) is as shown in Figure 3. The graph is plotted for number of sentences against the time taken to recognize sentences that are mentioned in Table 1. The blue line symbolizes recognition time for text without period recognition and the orange line stands for recognition with period. The recognition time is computed as the difference as the end time and start time of the recognition process. The time library in python is used to record time. The difference time can be computed using (1).

$$\textbf{Difference time = End time - Start time} \quad (1)$$

The time taken by the gensim library and the proposed method to summarize text is shown in Table 2. For validating the text summarisation, data was given as a continuous speech, documents of fifteen pages, and various websites for evaluating the performance. According to the data in the Table, the proposed method comparitively takes lesser time for summarization of the same number of lines as compared to Gensim library.

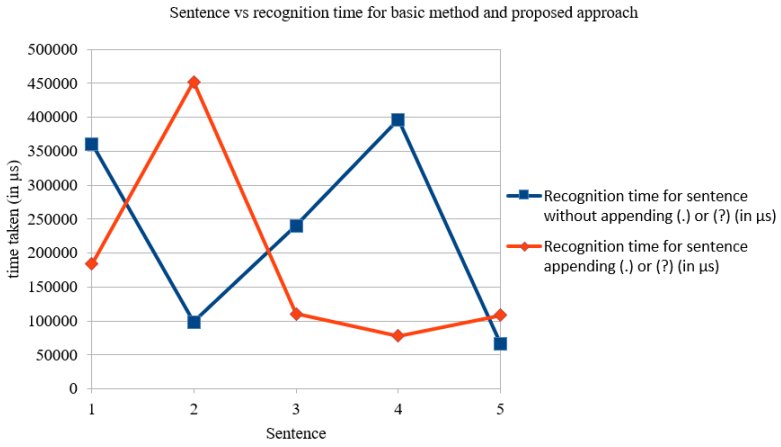

Figure 3. Comparison of speech recognition with dot and without dot

Table 2. Summarization time using Gensim library and proposed method

Number of lines	Summarized lines	Summarization time for Gensim library (in µs)	Summarization time for proposed method (in µs)
10	1	163680	6190
5	1	2336	1886
8	1	3675	3640
5	1	873887	3279
6	1	5576	3279

Comparing to the Gensim summarization, proposed algorithm consumes less time to produce the result. In Gensim summarization, though document is very big, result of the algorithm would be a line mostly. Proposed algorithm can summarize the whole document into a number of lines by the user requirement. The frequency of words are computed using (2) which is used to rank the sentences.

$$\text{Frequency [word]} = \frac{\text{frequency [word]}}{\text{total frequency value}} \quad (2)$$

By applying this formula, the frequency of the single word can be found. From here the index of the sentence can be found by the ranking method. In Figure 4, blue line stands for the gensim library and the orange line stands for the proposed approach. The x-axis shows the number of lines given as input that is shown in the Table 2 and y-axis shows the time taken. The time consumed and performance of the proposed approach is consistent for different inputs.

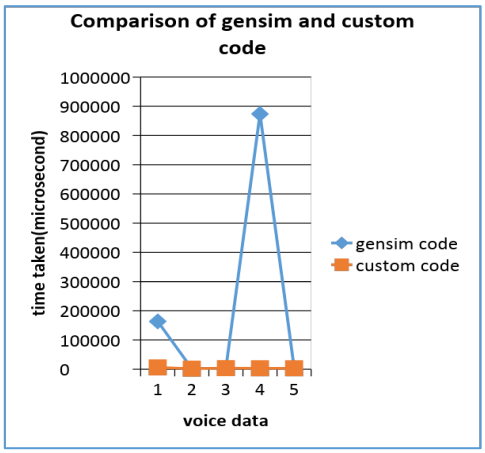

Figure 4. Comparison of summarization time for gensim library and proposed method

5. CONCLUSION

Speech recognition and text summarization are two vast areas to be explored. The proposed research work aims to reduce the time and effort of manual documentation of lengthy speeches in an event. Speech recognition and text summarization can ease the work of documentation. Even for the verification of the summarized content, the system can be automated to read out the summarised content with the help of text to speech conversion. As of now, speech summarization for sentences terminating with a full stop or containing a small pause shown by comma is experimented. The future work is to include all punctuation marks in the recognized speech which helps in improving the text summarization performance. This model can be used where ever there is a requirement of summarising lengthy lectures into precise documents as the automated system will convert the speech to text and also summarise the content. It can be of great help for students to archive lecture notes from classes, conferences or seminars.

REFERENCES

[1] Jose D V, Alfateh Mustafa, Sharan R, "A Novel Model for Speech to Text Conversion," *International Refereed Journal of Engineering and Science (IRJES)*, vol 3, no. 1, 2014.

[2] K. M. Shivakumar, V. V. Jain and P. K. Priya, "A study on impact of language model in improving the accuracy of speech to text conversion system," *2017 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, pp. 1148-1151, 2017.

[3] Y. H. Ghadage and S. D. Shelke, "Speech to text conversion for multilingual languages," *2016 International Conference on Communication and Signal Processing (ICCSP)*, Melmaruvathur, pp. 0236-0240, 2016.

[4] Umar Nasib Abdullah, Kabir Humayun, Ahmed Ruhan, Uddin Jia., "A Real Time Speech to Text Conversion Technique for Bengali Language," *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, pp. 1-4, 2018.

[5] 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. 307-310.

[6] Saiyed S., Sajja P. S., "Review on text summarization evaluation methods," *Indian Journal of Computer Science and Engineering*, vol. 8, no. 4, pp. 497, 2017.

[7] 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.

[8] 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.

[9] 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, 2017.

[10] J. Zenkert, A. Klahold and M. Fathi, "Towards Extractive Text Summarization Using Multidimensional Knowledge Representation," *2018 IEEE International Conference on Electro/Information Technology (EIT)*, Rochester, MI, pp. 0826-0831, 2018.

[11] 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, 2012.

[12] S. R. Rahimi, A. T. Mozhdehi and M. Abdolahi, "An overview on extractive text summarization," *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, pp. 0054-0062, 2017.

[13] V. Dalal and L. Malik, "A Survey of Extractive and Abstractive Text Summarization Techniques," *2013 6th International Conference on Emerging Trends in Engineering and Technology*, Nagpur, pp. 109-110, 2013.

[14] T. Jo, "K nearest neighbor for text summarization using feature similarity," *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, Khartoum, pp. 1-5, 2017.

[15] T. Tran and D. T. Nguyen, "Text Generation from Abstract Semantic Representation for Summarizing Vietnamese Paragraphs Having Co-references," *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, Ho Chi Minh, Vietnam, pp. 93-98, 2018.

[16] A. Vimalaksha, S. Vinay, A. Prekash and N. S. Kumar, "Automated Summarization of Lecture Videos," *2018 IEEE Tenth International Conference on Technology for Education (T4E)*, Chennai, India, pp. 126-129, 2018.

[17] S. Biswas, R. Rautray, R. Dash and R. Dash, "Text Summarization: A Review," *2018 2nd International Conference on Data Science and Business Analytics (ICDSBA)*, Changsha, China, pp. 231-235, 2018.

[18] Matsubayashi A. Yamashita H. Nonaka and Y. Konno, "A Research on Document Summarization and Presentation System Based on Feature Word Extraction from Stored Informations," *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, Taichung, Taiwan, pp. 60-63, 2018.