Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 3): pp. 4173-4183
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.03.431

Speech Summarization Using Extractive Text Summarization Approach

Vijay Singh Department of Computer Science & Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand India, 248002 vijaysingh.cse@geu.ac.in

Richa Gupta Department of Computer Science & Engineering Graphic Era Hill University, Dehradun, Uttarakhand India, 248002 richagupta@gehu.ac.in

Abstract:

This study describes how extractive text summarising algorithms may be used to accomplish speech-to-text summarization. Our goal is to determine which of the six summarization approach studied in this research is best suited for the job of audio summarization and to provide a suggestion. First, six text summarising methods have been selected: Luhn, LexRank, TextRank, KLSum, LSA, and SumBasic. Then, we analysed them using ROUGE measures on two datasets, DUC2001 and OWIDSum. Then, we picked five voice files from the ISCI Corpus collection and converted them employing the Automatic Speech Recognition (ASR) from the Google API. Findings revealed that Luhn and TextRank performed better at extracting audio summary on the analysed data.

Keywords: Natural Language Processing, Extractive Summary, Speech Recognition, Speech-to-text Summary

I. Introduction

Daily, new systems, tools, and programs are developed to handle massive amounts of data, the great majority of which is in multimedia content, like photos, audio, and video. Communication via speech is among the most successful strategies. Nevertheless, it is difficult to reuse, evaluate, or recover spoken documents that are stored as audio signals. It is difficult to extract usable information from audio recordings, particularly when the quantity of audio files or their duration is large. In addition, audio discussions may include redundant data, such as word segments, fillers, or repeats, as well as irrelevant material unrelated to the subject of interest or the purpose being pursued. Due to these factors, computerised summary of voive files might assist an individual in extracting the relevant data from a voice record without hearing to its whole. In addition to assisting those with impairments, speech-to-text tools convert the contents of a voice recording into a written file. Transfering voice files to textual content might facilitate the management and data processing included in sound recordings. Natural Language Processing (NLP) methods may be simply used to extract information from a written source.

The purpose of this study is to automatically summarise voice files collected as audio using approaches for extractive summarization approach. With this purpose, we want to reduce the amount of time needed to execute audio data. Six text summarising techniques, namely Luhn [1], LexRank [3], TextRank [2], KLSum [5], LSA [4], and SumBasic [6], have been examined. We compared their effectiveness against the OWIDSum [8] and DUC2001 [10] datasets using the ROUGE measures. Then, we compared the six techniques to the transcribed verbatim of five sound recordings from the ICSI-Corpus dataset, calculating the ROUGE metrics and recommending an extractive text summarising approach for audio summarization.

The following description illustrates how the document is structured. In Section 2, we conduct a concise analysis of the current technology with regard to extractive summarization strategies and the approaches that are employed to transcribe voice notes to text. Following that, in Section 3, we will discuss the six separate approaches of summarization that were used in this paper. In Section 4, the experimental design, the datasets, the outcomes of the experiments, and the comments are provided. In the Section 5 & 6, we will examine the conclusion as well as any more work that needs to be done.

II. Literature Study

2.1 Extractive Summary

A process known as automatic text summarization condenses the original text while retaining all of the information it originally included. The processes that are used to summarise texts are often categorised as either extractive or abstractive [7], dependent on the manner in which the final content is created. In this project, we are going to use the extractive method, which means that a summary will be constructed by choosing a number of full phrases from the original text and utilising those sentences as building blocks.

Text summarization methods have traditionally been based either on the frequency with which individual words occur in the document, as in the case of Luhn [1], or on graphs, as in the cases of TextRank [2] and LexRank [3]. The Latent Semantic Analysis (LSA) [4] approach depicts each text as a matrix, on which decomposition values are performed. Several schemas, such as KLSum [5] and Sum-Basic [6], are offered here based on the probability distribution of the words in the respective texts. When it comes to the duty of summarising the content of the Tor darknet, researchers have analysed the five algorithms that have previously been stated. In order to accomplish this goal, they presented OWID-Sum [8, which is a dataset that is made up of sixty text documents taken from Tor domains and organised into six distinct categories. TextRank [3] was suggested by the researchers who were working with the DUC2002 [10] dataset as well since it produced the superior ROUGE metrics out of the two datasets that were assessed.

The authors of reference [5] investigated probabilistic models for the synthesis of numerous documents using a variety of algorithms, including KLSum [6, SumBasic [7],

TopicSum [11], and HieSum [12], while operating on the DUC2006 dataset utilizing ROUGE metrics [13]. Calculating the relative relevance of text units for natural language processing is discussed in detail in [3, which presents a stochastic technique based on graphs for doing the calculation (NLP). On both the DUC2003 and the DUC2004 datasets, the LexRank-based algorithm achieved the highest possible scores. A novel algorithm that is based on LSA is suggested in [4] to analyse the replies of certain pupils to their instructor by evaluating the pre-defined records. This evaluation method is presented as a means of determining the quality of the responses.

Recent research shows a great number of research that are built on deep learning methods[23] and that attempt to solve the issue of text summaries by using a variety of neural network topologies have been published. A method that makes use of Recurrent Neural Networks (RNN) is provided in [14]. This approach makes use of RNNs for encoding, and unidirectional RRNs are used in hidden units for decoding. The reserachers assessed their approach using three different datasets, one of which being the DUC Corpus, which was used in our investigation. Using the ROUGE measure, they were able to get a score of 29.481 for their proposal. Reference [15] also employed RRN in their work, but they did so in order to do it. A Convolutional Neural Network (CNN) based on a new system is introduced in [16] and given the name PriorSum. Its purpose is to capture the preceding summary and combine it with the variable feature of the text while operating inside a vector autoregressive model. The ROUGE measure was calculated using this approach and applied to the datasets DUC2001, DUC2002, and DUC2004, yielding the following results: 37.13, 39.21, and 40.01, respectively. In a separate piece of research, author [17] described a CNN that could produce inlays of words to construct overlays of texts in the very same latent space. A system with centralized structure that was introduced in [18], in which CNN created the representation and RNN represented the document. A method that is built on autoencoder and recurrent network is suggested in [19] in order to enhance the capacity of the summary as well as its overall quality. The architecture of a sequence-to-sequence focused encoder-decoder that is coupled with a latent structure modelling component of varying auto-encoders serves as the foundation for this approach. As a consequence of this, they came up with a metric of 37.15 when using ROUGE-1 on the GIGA dataset, and they came up with a value of 38.89 while using ROUGE-1 on the LCSTS dataset.

The authors enlarged their prior studies on the Tor darknet in [9], where they recommended SummCoder as an unsupervised method for extractive summarization. This method is realized by means of autoencoders and the standings of sentences according to three distinct criteria: subject matter, innovation, and position. They attained state-of-the-art results on both their database and the DUC2002 standard, and they did this by extending the dataset that was suggested in [8] all the way up to 100 samples and adding two gold summaries.

2.2 Speech to text Summary

Speech technology is a multidisciplinary field of linguistic anthropology that produces approaches and systems that enable the identification and conversion of spoken language into textual. These methodology and technologies are referred to as "voice recognition systems." Also termed as computer voice recognition, automated voice recognising, automatic speech recognition (ASR), verbal synthesis, voice synthesis, and speech-to-text synthesis (STT). In the past, acoustic models and their reflection and transmission matrices were determined using frequency based [20] and relative linear dependency [21] approach. Deep Neural Networks (DNN) frameworks have recently become quite relevant in this industry as a result of considerable advances in computing technology and an increase in the amount of data that is accessible for training. When we combine some of the more contemporary approaches with some of the more traditional ones, we are able to get greater outcomes. In their paper [24], Villalba and colleagues provided a technique for identifying whether a speech detection method has administered the incorrect test. In some instances, the quality of the signals that are a part of the verification test is not as high as it would need to be in order to make a conclusion that is trustworthy. This approach is founded on simulating voice quality measures employing Bayesian Networks. Analyzing prior studies and using their method to get rid of unreliable tests have resulted in a significant improvement of the real diagnostic minimization problem.

III. Methodology

Figure 1 is a schematic depicting the processes of the intended speech-to-text summarization workflow. Initially, the audio stream is segmented into little bits so that it may be analyzed using the ASR approach. The audio files are then converted into text using the ASR technique. Subsequently, the resultant content is pre-processed by inserting commas to distinguish text segments if the ASR algorithm does not do so effectively. In the subsequent stage, the six extractive summarization techniques are used to the captured speech to provide a summary. The ROUGE metrics are computed utilising dynamically created and gold summary.

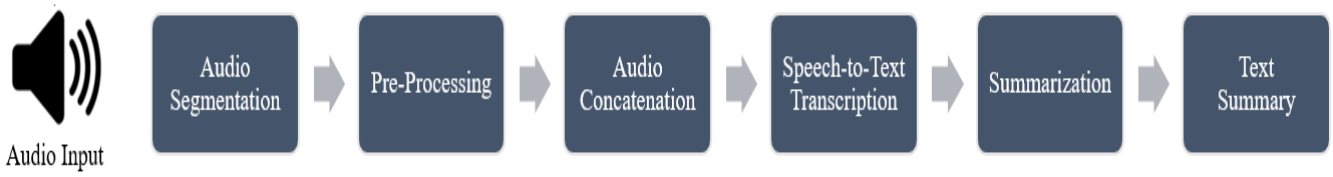

Figure 1: Proposed workflow

3.1 ASR

This method made use of the Google Cloud Speech API, which can be found in the SpeechRecognition3.8.1 module in the Python programming language. This application programming interface (API) makes it possible to transcribe audios that were physically

captured with a mic as well as export a sequence of records in order to apply neural network models to voice in order to recognise speech. The application programming interface is capable of recognising a total of 120 languages and their variations; for our projects, we made use of the Synchronous Recognition feature.

3.2 Summarization Approaches

Luhn

Luhn [1] is the most widely used methods for summarising text; it takes into account both the frequency with which words appear in the text and the distance among relevant terms, the latter of which is influenced by the proportion of non-relevant words among relevant ones. After the algorithm has identified the important words, it uses a significance factor based on the frequency with which those words appear and the linear distance between them that results from the presence of insignificant terms in between to decide how to punctuate each sentence.

LexRank

Another method for summarising text using graphs is LexRank [3]. The connections between vertex are calculated using the cosine similarity metric of words, which are expressed as TF-IDF measure vectors. After that, we have a similarity network where each phrase is a vertices and the cosine similarity behind them is a line with weighted data. To identify pairs of phrases that are statistically similar, a threshold-based method is used.

TextRank

Using a vertex-based representation of phrases, the authors of [2] suggested a method for summarising text called TextRank, which is founded on graph theory. A critical value is assigned to each vertex by considering global data derived recursively from the whole network. The PageRank algorithm does this by counting how many other pages link to the target page in question and assigning a value to that number. Every nodes in the text summary is a phrase, not a homepage, which allows PageRank to be used. Given that our article's instances lack inter-page linkages, we characterised sentence similarity by the amount of shared terms. After calculating the similarities between the phrases, a graph is shown in which no two adjacent vertices need to be linked if there is no internal similarity between the words. Each edge between vertex will also have a strength that indicates the strength of the connection between them.

KLSum

This approach gives the actual probability of the unigram of the prospective summary [6] . Unigram frequency (P) is a representation of the document's unigram distribution, whereas summary frequency (Q) is a representation of the document's summary frequency. The criteria's goal is to provide a summary that is as faithful to the original as possible.

LSA

It is possible to automatically extract the statistical link between words in a phrase using a method called latent semantic analysis (LSA) [4]. To make sense of the text, it is first broken down into sentences and then into words, each of which is a distinct string of letters. The text is then shown as a matrix, where the occurrence frequency of each word in a given paragraph is recorded in a given cell of the matrix. Then, a function that captures both the word's significance in the context of the passage and its role in the discourse domain is applied to each cell's frequency. When everything else fails, remove the diagonal matrix's coefficients using Single Value Decomposition (SVD) [25].

SumBasic

SumBasic [5] is a redundancy-reducing technique that re-weights the probability of individual words depending on their frequency in sentences. First, it determines the average frequency with which each word appears in the input, and then it gives each word in the phrase a weight proportional to this average frequency. Once that's done, we look for the highest-scoring sentence that includes the most probable term. At long last, we update the likelihood of each word in the previously selected phrase. The summary's length may be further pruned by using the procedure repeatedly.

IV. Experimental Evaluation

4.1 Evaluation Metric

As a collection of measures for evaluating text summarising techniques, ROUGE [13] is becoming more popular. This statistic evaluates algorithm-generated summaries against those crafted by humans (termed "Gold Summaries"). The number of overlapping units, or the number of units that are included in both summaries, is calculated using ROUGE measure. In this section, we describe the ROUGE measures that were used to this study.

ROUGE-N is a statistical measure determined by the number of matched “n-grams” among the summary derived and the standard summary. n-grams are sequences of n characters, which might be characters, consonants, or phrases.

4.2 Experimental Setup

We conducted our experiments using a laptop equipped with a quad-core Intel Core i7 and 16GB of DDR4 RAM, and we found that cutting the acoustic input into chunks of 5-10 seconds made it easier for the Google API to transcribe the voice. Once the textual representation of each audio file has been created, the outputs will be joined together to form the final document. Since automated speech recognition (ASR) does not offer punctuations, they were inserted by hand to the transcribed text in order to demarcate the various sequences.

4.3 Dataset

Throughout the course of this research, we have made use of a total of three distinct datasets, including two text datasets for the purpose of assessing the extractive text summarization techniques, as well as a voice dataset for the purpose of assessing the whole process. Table 1 gives the brief description of the three dataset used in this work.

Table 1: Dataset Description

Dataset	Description
Document Understanding Conference, 2001 (DUC2001)	News stories on 30 different topics, like mad cow disease, Hurricane Andrew, and an aircraft accident in Sioux, are included in the 303 articles that make up the DUC2001 dataset.
Onion Web Illegal Document Summarization (OWIDSum)	Originally released as part of the Darknet Usage Text Addressed (DUTA) dataset [26], the OWIDSum is the precursor of the more recent Tor Illegal Domain Summarization (TIDSumm) [8]. The 6831 hidden Tor Network domains in DUTA are organised into 26 categories that include both legitimate and illicit pursuits on the dark web. Sixty papers in all and two standard summary for each paper make up OWID-Sum, which is organised into the following six categories: counterfeit credit, cybercrime, drug sales, counterfeit goods, the marketplace, and cryptocurrencies.
International Computer Science Institute (ICSI) Corpus	The ICSI has a collection that includes 75 audio talks involving many participants, ranging in length from 30 to 70 minutes. There are additionally 20 synopses culled from 17 different recordings. Metrics related to ROUGE may be computed using it. Out of the 17 audio samples available in the ICSI Corpus, we have used 5.

4.4 Results

Six methods are discussed, and their performance on the DUC2001, OWIDSUM, and ICSI datasets has been measured using ROUGE-N metrics and is shown in Table 2. The graphical representation of the results are depicted in Figure 2. As a matter of fact, it's easy to see that the ROUGE score for Luhn and TextRank are comparable better, but the findings for KLSum and SumBasic are much lower. Based on our evaluation of the source speech documents, we find that Luhn is the best appropriate algorithm.

Table 2: Performance measure

Approach	**Dataset**		
	DUC2001	OWIDsum	ICSI corpus
Luhn	**0.458**	**0.401**	**0.595**
LexRank	0.441	0.382	0.541
TextRank	0.433	0.395	0.552
KLSum	0.371	0.331	0.290
LSA	0.332	0.362	0.521
SumBasic	0.395	0.292	0.255

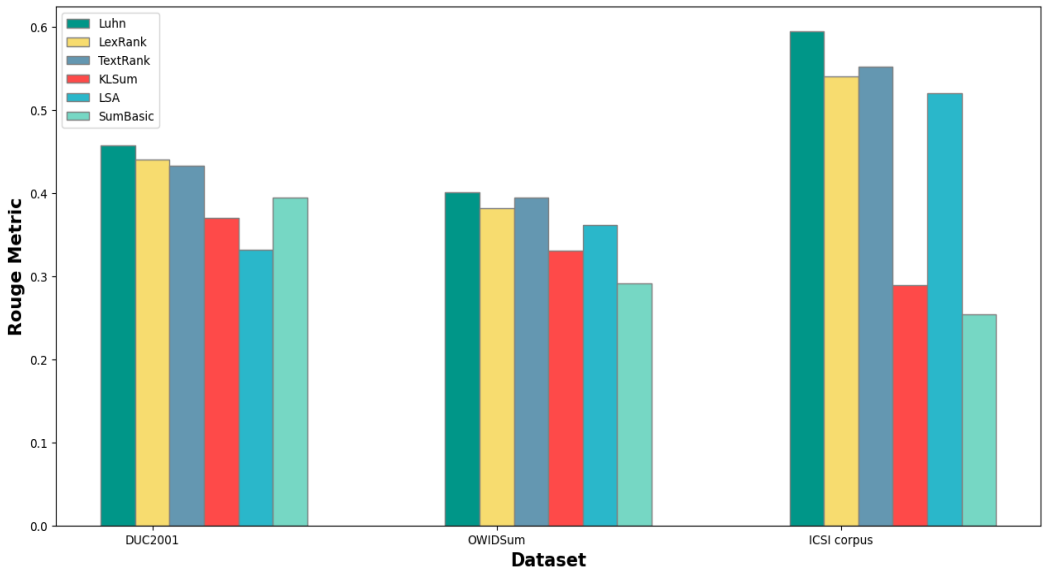

Figure 2: Performance Measure

V. Conclusion

In this paper, we introduced a workflow that automatically summarises audio information by the application of extractive text summarising algorithms to text transcribed from audio utilizing automated speech recognition technology. Six approaches were chosen from the assessed conventional text summary strategies and analysed using two summarization datasets, DUC2001 and OWIDSum, and ROUGE measures. The experimental findings demonstrated that the best results were produced by the Luhn and TextRank techniques in DUC2001 and OWIDSum, correspondingly. Next, Google Cloud Voice API ASR was used to transcribe five speech documents from the ICSI Corpus, and the generated text files were appraised employing the same extractive techniques. When it comes to extractive text summarising, the best results were achieved by the Luhn and TextRank algorithms, therefore they are the first suggestion for solving the automated speech-to-text summarization job.

Further research might build on this study by either automatically generating gold summaries for all of the speech documents in the ICSI Corpus dataset or manually elaborating standard summary for the remaining audio files in the dataset. The absence of extractive standard summary is the biggest barrier to testing this concept on other publicly available datasets. Following the development of suitable gold extraction summaries, the testing of other speech datasets might be undertaken as future study. Additionally, punctuation predictions on individual word patterns will be performed automatically.

References:

[1] Luhn, H.P. (1958). The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, 2(2), pp. 159-165.

[2] Mihalcea R., and Tarau P. (2004). Text Rank: bringing order into texts. Proceedings of EMNLP-04 and Conference on Empirical Methods in Natural Language Processing (EMNLP).

[3] Erkan G. and Radev D.R. (2004). Lex Rank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research, 22, pp. 457-479.

[4] Xiangen H., Zhiqiang C., Max L., Andrew O., Phanni P., and Art G. (2003). A revised al-gorithm for latent semantic analysis. In Proceedings of the 18th international joint conference on Artificial intelligence (IJCAI'03). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1489-1491.

[5] Aria H. and Lucy V. (2009). Exploring content models for multi-document summarization. In Proceedings of Human Language Technologies, Annual Conference of the Noth Ameri-can Chapter of the Association for Computational Linguistics (NAACL '09). Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 362-370.

[6] Nenkova A. and Wandervende L. (2005). The impact of frequency on summarization. Tech-nical report, Microsoft Research.

[7] Murthy Vishnu, Vishnu V. B., Mekala, Vijaypal S. P., and Reddy V. (2013). Text Classifi-cation using Text Summarization– A case study on Telugu Text. International Journal of Advanced Research in Computer Science and Software Engineering, pp. 1399-1403.

[8] Joshi A., Fidalgo E. and Alegre E. (2018). Summarization of text from illegal documents in Tor domains using Extractive Algorithms. International Conference on Applications of In-telligent Systems (APPIS), Las Palmas de Gran Canaria.

[9] Joshi, A., Fidalgo, E., Alegre, E., Fernández-Robles, L. (2019). SummCoder: An Unsuper-vised Framework for Extractive Text Summarization Based on Deep Auto-encoders. Expert System with Applications.

[10] DUC 2002. Document Understanding conference (2002). https://duc.nist.gov/. Last ac-cessed: 07/04/2019.

[11] David M. B., Andrew Y. N., and Michael I. J. (2003). Latent Dirichlet allocation.JMLR.

[12] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. 2004. Hierarchical topic models and the nested Chinese restaurant process. In NIPS.

[13] Chin-Yew Lin (2004). Rouge: A package for automatic evaluation of summaries. In Pro-ceedings of the ACL Workshop, pp. 74-81.

[14] Nallapati, R., Zhou, B., Santos, C. D., Gulcehre C., and Xiang B. (2016). Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond.

Proceedings of the 20th SIGNAL conference on Computational Natural Language Learning, pp. 282-290.

[15] Chopra S., Auli M., and Rush A. M. (2016). Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 93-98.

[16] Cao Z., Wei F., Li S., Li W., Zhou M., and Wang H. (2005). Learning Summary Prior Rep-resentation of Extractive Summarization. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 829-833.

[17] Cao Z, Wenjie L., Sujian L., Furu W. and Yanran L. (2016). Attsum: Joint learning of fo-cusing and summarization with neuronal attention. COLING, pp. 547-556.

[18] Cheng J., and Lapata M. (2016). Neural Summarization by Extracting Sentences and Words. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 484-494.

[19] Lil P., Lam W., Bing L., and Wang Z. (2017). Deep Recurrent Generative decoder for Ab-stractive text Summarization. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Proceedings, pp. 2091-2100.

[20] Martinez J, Perez-Meana H, Escamilla-Hernandez E., and Suzuki M.M. (2012). Mel-Fre-quency Cepstral Coefficients For Speaker Recognition: A review. (2015. International Jour-nal of Advanced Engineering and Research Development, pp. 248-251.

[21] Zulkifly M. A., and Yahya N. (2017). Relative spectral-perceptual linear prediction (RASTA-PLP) speech signal analysis using singular value decomposition (SVD). IEEE 3rd International Symposium on Robotics and Manufacturing Automation (ROMA).

[22] Xuan G., Zhang W. and Chai P. (2001). EM Algorithms of Gaussian Mixture Model and hidden Markov model. Proceedings of the 2001 International Conference on Image Processing, pp. 145-148.

[23] Ravichandra, S., Siva Sathya, S., Lourdu Marie Sophie, S. (2022). Deep Learning Based Document Layout Analysis on Historical Documents. In: Rout, R.R., Ghosh, S.K., Jana, P.K., Tripathy, A.K., Sahoo, J.P., Li, KC. (eds) Advances in Distributed Computing and Machine Learning. Lecture Notes in Networks and Systems, vol 427. Springer, Singapore. https://doi.org/10.1007/978-981-19-1018-0_23

[24] Villalba J., Lleida E., Ortega A. and Miguel A. (2012). Reliability Estimation of the Speaker Verification Decisions Using Bayesian Networks to Combine Information from Multiple Speech Quality Measures. Advances in Speech and Language Technologies for Iberian Language, pp. 1-10.

[25] Gliozzo A. M., Giuliano C., and Strapparava C. (2005). Domain Kernels for Word Sense Disambiguation. 43rd Annual Meeting of the Association for Computational Linguistics, pp. 403-410.

[26] Al-Nabki, M., Fidalgo, E., Alegre, E., and Fernández-Robles, L. (2019). ToRank: Identify-ing the most influential suspicious domains in the Tor network. Expert System with Appli-cations, vol. 123, pp.212–226.