The automated method of Summarization of Audio Data

Jatin Pardhi (jatinpardhi00@gmail.com)
Visvesvaraya National Institute of Technology
Tanmay Kharik
Visvesvaraya National Institute of Technology

Research Article

Keywords: AssemblyAI API, Call centers, Graph-theoretic approach, Speech recognizer

Posted Date: April 22nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1565580/v1

The automated method of Summarization of Audio Data

Jatin Pardhi[1*†] and Tanmay Kharik[2*†]

[1*]Electronic and communication Enginering, Visvesvaraya National Institute of Technology , Nagpur, 440010, Maharashtra, India.

[2*]Electronic and communication Enginering, Visvesvaraya National Institute of Technology , Nagpur, 440010, Maharashtra, India.

*Corresponding author(s). E-mail(s): jatinpardhi00@gmail.com; ttkharik@gmail.com;

†These authors contributed equally to this work.

Abstract

In today's modern era, most of the organizations have their own call centers or they outsourced another company for handling calls. Customers have high expectations that their issues be addressed and handled quickly and efficiently. In call-centers when a customer problem is not resolved by one service person, it is forwarded to the person who is the superior one. Most of the time the call recording of the customer is used for further references. In that case whoever has reviewed the call recording has to go through the entire recording which is a very tedious task. In this paper, we designed a methodology by which we have summarized the audio data. First, we used a Speech recognizer for converting audio data to Text using AssemblyAI API and then summarized it using the Graph-theoretic approach.

Keywords: AssemblyAI API, Call centers, Graph-theoretic approach, Speech recognizer

1 Introduction

1.1 Speech to Text Conversion:

Fig. 1: Speech-To-Conversion[6]

With the popularity of Alexa, Siri, and even Fridges that can talk, speech-enabled devices have become an indispensable part of our lives.Tech giants such as Google and Facebook have published a lot of APIs which can perform really well in a lot of scenarios[1].Speech is the most widely used mode of communication, with the majority of the world's people relying on it to communicate with one another. The basic function of the Speech recognition system is to convert spoken languages into text[2]. Speech recognition internally converts Physical sound into an electrical signal, then this signal is digitized with the help of an analog to digital converter, once the digitized model is used it to transcribe the audio to text.

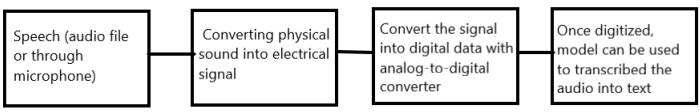

Fig. 2: Speech Recognition Process[1]

1.2 Text Summarization:

Before we move on to text-summarization, first we must understand what summary means.A summary is a text which gives us a quick or short review that conveys key information from the original text[3]. The purpose of a text summarizer is to generate concise and precise summaries from the vast text without losing the overall meaning of the original text[4].Text Summarizer helps us to reduce the reading time.
There are two main types of text summarization:
1)Extractive summarization.

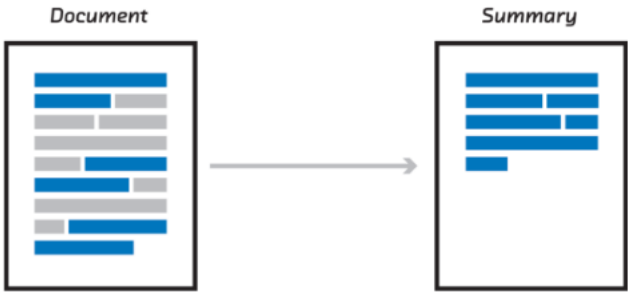

Fig. 3: Extractive Summarization[4]

2)Abstractive summarization.

An Extractive summarizer identifies the important section and extracts important sentences/information from the original text or given document and concatenates them to produce a condensed version.Unlike an extractive summarizer, an abstractive summarizer does not copy-paste important sections/sentences from the original text/document but comes up with new relevant phases.Abstractive summarization is implemented with the help of Natural Language Processing (NLP).The most recent effective approach to perform abstractive text summarization is using transformer models.This model is fine-tuned for summarization purposes[5].

2 Literature Review

2.1 Speech recognition:

Machine recognition was invented in the early 1920s.[6]Radio Rex (toy) was the first commercially named machine to detect speech and was manufactured in 1920.During 1950, researchers mainly focused on spectral resonances in the vowel region of each syllable[7].Fry and Denes attempted to create a phoneme recognizer to recognize four vowels and nine consonants in 1959 at University College in England[8]. In the 1960s computers were slow so special purpose hardware was used for speech recognition[9].Hardware vowel recognizer, Suzuki and Nakata were discovered in Radio Research Lab, Tokyo[10].In the early 1980s, Moshey J. Lasry created a feature-based voice recognition system. In his research, he looks into voice spectrograms of letters and numerals[11].In 1980s key technologies like Hidden Markov Model(HMM) were developed[12–14].Neural Net was also reintroduced in the late 1980s.Different new ways were proposed for implementing systems[15].DARPA program[16], HMM, Robust speech recognition, Noisy speech recognition these new technologies were discovered in the period 1990-2000s.Variational Bayesian (VB) estimation and clustering algorithms

were developed around the year 2000[17].This VB estimation technique was based on the posterior distribution of parameters[18].For the speech Hidden Markov model, a unique "regression-based Bayesian predictive classification" (LRBPC[20]) was developed in 2003.

In 2007 SR methods were investigated by Sadaoki Furui[19] which can adapt to speech variation using clustering techniques.In 2016, the authors[22] implemented real-time speech-to-text conversion using a bidirectional Kalman filter.Their system was tested in a real-time noiseous environment.Yogita H. Ghadage, Sushama D. Shelke[23] performed speech-to-text conversion for multilingual languages using the "Mel-frequency Cepstral Coefficient (MFCC)" feature extraction technique, Support Vector Machine (SVM), and Minimum Distance Classifier. In 2020, the authors[24] used a probabilistic graphical model for speech recognition.[25]

Sl.No	Past	Present(new)
1)	Template matching	Corpus-based statistical modeling, e.g. HMM and n-grams
2)	Filter bank/spectral resonance	Cepstral features, Kernel-based function, group delay functions
3)	Heuristic time normalization	DTW/DP matching
4)	Distance-based methods	Likelihood-based methods
5)	Maximum likelihood approach	Discriminative approach e.g. MCE/GPD and MMI
6)	Isolated word recognition	Continuous speech recognition
7)	Small vocabulary	Large vocabulary
8)	Context Independent units	Context-dependent units
9)	Clean speech recognition	Noisy/telephone speech recognition
10)	Single speaker recognition	Speaker-independent/adaptive recognition
11)	Monologue recognition	Dialogue/Conversation recognition
12)	Read speech recognition	Spontaneous speech recognition
13)	Single modality(audio signal only)	Multimodal(audio/visual)speech recognition
14)	Hardware recognizer	Software recognizer

Fig. 4: Summary of Technology Progress[15]

2.2 Text Summarization:

The authors[26] discussed different text summarizing techniques and presented a comprehensive survey of both extractive and abstractive text summarization.Neelima Bhatia; Arunima Jaiswal[27] in their survey paper focused on single and multiple document summarizations and different extractive techniques. The paper "A survey of text summarization extractive techniques"[32], discussed various extraction-based summarization algorithms and also included the broader comparison of these algorithms.

3 Methodology

3.1 Speech to Text Conversion method:

Converting audio data into text data is quite a tough task. There are multiple tools and APIs present out there to simplify this task. Each one of them offers different sets of features and control over speech recognition and conversion. Few api are good at speech recognition but they are comparatively slower than others while some are quicker than others but the results obtained are not satisfactory. To choose the best suitable API which is suitable to our needs we have to test these APIs on multiple parameters.

The API that we have considered for our purpose are Google Speech Recognition API, IBM Watson API, AssemblyAI API, etc. We have made comparisons based on multiple factors like speed, accuracy, flexibility, language support, etc. After multiple trials, we have come up with the conclusion that the assembly AI API is the most suitable API for our purpose. The results obtained are quite satisfactory compared to other APIs.

Audio	IBM (Watson)	Amazon	Google	Microsoft Az...	Speechmatics
1	14.72	8.97	10.34	8.21	9.38
2	32.57	24.3	35.29	31.47	24.25
3	14.83	12.34	14.76	19.48	8.39
4	9.37	8.19	10.92	9.55	6.79
5	31.75	19.34	23.31	20.93	17.48
6	10.22	7.36	7.01	7.49	6.13

Fig. 5: Comparison of different AP[7]

3.2 Text Summarization:

Text summarization is a technique for extracting the short and precise summary of lengthy paragraphs without losing its actual meaning. There

are two ways of doing this task, Abstraction-based text summarization, and extraction-based text summarization[32]. In the Abstraction method, computationally complex and time-consuming ML algorithms are used. On the other hand Extraction, based text summarization can be done by simple loops and data structures. In this project, we have implemented Extraction based text summarization.

Abstractive text summarization works just the way the human brain thinks. It first tries to understand the meaning of the text paragraph and accordingly delivers a meaningful summary. On the other hand, Extractive text summarization doesn't write the summary on its own. It just selects a few meaningful sentences in the paragraph and delivers the summary by joining them. The order and meaning of sentences won't change.

The text data obtained from the speech to text part is first preprocessed and cleaned using various methods to remove the insignificant data. First of all, all the stopwords are removed from the paragraph since these words do not carry any significance in the sentence. Later a hashmap has been generated to store the frequency of occurrence of each word in the entire paragraph. By using this hashmap we can calculate the weighted frequency of each sentence by simply adding the frequencies of all the significant words in the sentence and then dividing it by the number of significant words in the sentence to ensure that the long sentences do not carry unnecessarily high weightage. Now to select the sentences eligible for summarization we have to calculate the threshold frequency which is simply the average frequency of all the sentences present in the paragraph. Now by selecting all the sentences whose frequency is above the threshold frequency,we can serve these sentences as the summary of the paragraph.

In this way, without using any computationally complex algorithm we can extract the summary of the paragraph easily using only simple data structures and for loops.

3.3 Graph-Theoretic Approach for Text Summarization:

This approach of text summarization is based on Graph theory and Cosine Similarity. In this approach, the sentences of the document can be represented as the vertices of the graph while the relation between two vertices or sentences can be calculated using cosine similarity.

The text data that we have obtained is first converted into lowercase to maintain uniformity. As this data consists of lots of unnecessary elements like stopwords, non-alphabetical characters, special characters etc, we have to preprocess this data to remove this data which carries no value in the respective sentence. Since this algorithm is also an extraction-based algorithm, we have to separate all the sentences so that we can select particular sentences at the end. The document can contain different words with the same meaning but in different forms to solve this issue we can perform lemmatization on the entire document. Lemmatization is the process of extracting words in their root form

so that the same words in different forms can be analyzed as a single one. It links words with similar meanings to one word.

The next task is to find out the relation between any two sentences. This can be done by using the cosine similarity technique. This method is a vector-based method so we have to first convert all the sentences in vector form. We can find multiple ways of doing this. Like Bag of Words, Bag of N-Grams, Words Count Strategy, Term Frequency Strategy, TF-IDF Strategy, Doc2Vec, etc. All of them are kind of similar approaches. Each value in vector representation represents the weight of the respective word in the text. The order of the sentence can be lost while vectorization.

To calculate the similarity between two sentences we can use the Cosine Similarity approach. It is a method for approximating how similar two-sentence vectors are. It just calculates the angle between two vectors to quantify how similar two sentences are. Suppose we have two non zero vectors A and B, the similarity can be calculated using the formula given by

Similarity= Cos(x)=A.B/ (A*B)

Smaller the angle, the higher the similarity. This way we can quantify how similar two vectors or two sentences are.

Since we are trying to find the summary of the entire text or document we need cosine similarity between all the sentences present in the document. For this purpose, we can create a Similarity Matrix[30]. The elements of Similarity Matrix store pairwise similarities of objects, the greater the similarity higher the value. It is a square matrix whose dimensions are equal to the total number of sentences present in the document. Each index represents the respective sentence in the document. And each cell represents the cosine similarity between sentences represented by the row and column number of that particular cell. It will be a symmetric matrix.

After forming a Similarity matrix it has to be converted into a Graph format because we will be using the PageRank algorithm on this data. PageRank is the algorithm used by Google to rank their web pages in their search engine results. This Graph based algorithm[29] is mainly used to decide the importance of a vertex within a graph based on global information recursively drawn from the entire Graph. That is the main motive behind converting the similarity matrix into a graph format. In this Graph each vertex will represent a particular sentence and edges will represent the cosine similarity between two vertices or sentences. The PageRank algorithm is traditionally applied on directed graphs, a recursive graph-based ranking algorithm can be also applied to undirected graphs.

The PageRank algorithm[31] will initially assign equal value to all nodes. After each iteration, the values of nodes will be updated to new values. The new PageRank value will be proportional to the sum of all the PageRank values of linked nodes. The precision of the PageRank algorithm depends upon the number of iterations. We can manually set this value. After completion of all the iterations, the nodes will be sorted according to PageRank values in

descending order. Top nodes or sentences can be selected as the summary of the document[33]. As per the requirement, the top sentences can be selected.

4 Pseudocode

Input: Entire text
tokenize()
For each sentence in input:
lowercase()
stopwords removal()
nonalphabets removal()
lemmatize()
similarity matrix ¡- cosine similarity()
Graph ¡- similarity matrix
Page Rank(Graph)
Output ¡- Summary()

5 Results and Discussion

First we checked the Speech to text algorithm using AssemblyAPI on different speech inputs. The results obtained were satisfactory and accurate. Then the Text summarization algorithm has been tested against different text inputs. It's working well and providing an extractive and meaningful summary out of it. The whole program is then tested for audio samples. For each speech sample, the algorithm is generating a meaningful text summary.

Table 1: Rogue Scores[32]

Column 1	rogue-1 score	rogue-2 score
Frequency driven approach	0.8842	0.7127
Graph approach	0.9696	0.9384[1]

6 Future Work

In this paper we have discussed the audio summarization limited to only the English language. In the future, the same algorithm can be implemented in different possible languages.

Declarations

- Funding : Not Applicable
- Conflict of interest: The authors declare that they have no conflict of interest.

- Ethics approval : Not Applicable
- Consent to participate: We the authors provide our consent for participation in this journal.
- Consent for publication: We the author provide our concern for publication in this journal
- Availability of supporting data: The data used for this project was acquired through Kaggle.com keeping in mind their Open Access license(CC0: Public Domain)
- Code availability : The code generated during the current study are available from the corresponding author on reasonable request.
- Competing interest : The authors declare that they have no competing of interest.
- Authors' contribution : The authors contributed equally to this work.
- Authors' information : Affiliations Department of Electronics and Communication Engineering, Visvesvaraya National Institute of Technology, Nagpur, Maharashtra, India
  Mr. Jatin Pardhi
  Department of Electronics and Communication Engineering, Visvesvaraya National Institute of Technology, Nagpur, Maharashtra, India
  Mr. Tanmay Kharik
- Acknowledgments: This work was partially supported by the department of Electronics and Communication Engineering, Visvesvaraya National Institute of Technology- Nagpur. We thank Dr. Prabhat Sharma for guiding us throughout the project and study.

References

[1] Bhupinder Singh, Neha Kapur, Puneet Kaur "Speech Recognition with Hidden Markov Model:A Review" International Journal of Advanced Research in Computer and Software Engineering, Vol. 2, Issue 3, March 2012.

[2] Hakan Erdogan, Ruhi Sarikaya, Yuqing Gao, "Using semantic analysis to improve speech recognition performance" Computer Speech and Language, ELSEVIER 2005.

[3] F. D. Malliaros and K. Skianis, "Graph-based term weighting for text categorization", Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, 2015.

[4] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez and C. J. Pérez, "Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach", Knowledge-Based Systems, 2017

[5] A. Khan, N. Salim, H. Farman, M. Khan, B. Jan, A. Ahmad, et al.,"Abstractive Text Summarization based on Improved Semantic Graph Approach", International Journal of Parallel Programming, pp. 1-25, 2018.

[6] Stefan Windmann and Reinhold Häb-Umbach, Approaches to Iterative Speech Feature Enhancement and Recognition, IEEE Transactions On Audio, Speech, And Language Processing, Vol. 17, No. 5, July 2009.

[7] Sadaoki Furui, 50 years of Progress in Speech and Speaker Recognition Research, ECTI Transactions on Computer and Information Technology,Vol.1. No.2 November 2005.

[8] D.B.Fry, Theoretical Aspects of Mechanical Speech Recognition, and P.Denes, The Design and Operation of the Mechanical Speech Recognizer at University College London, J.British Inst. Radio Engr.,19:4,211-299,1959.

[9] J.Suzuki and K.Nakata, Recognition of Japanese Vowels Preliminary to the Recognition of Speech, J.Radio Res.Lab37(8):193-212,1961.

[10] T.Sakai and S.Doshita, The phonetic typewriter, information processing 1962, Proc.IFIP Congress, 1962.

[11] R.K.Moore, Twenty things we still don't know about speech, Proc.CRIM/ FORWISS Workshop on Progress and Prospects of speech Research and Technology, 1994.

[12] J.Ferguson, Ed., Hidden Markov models for speech, IDA, Princeton, NJ,1980.

[13] L.R.Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc.IEEE,77(2),pp.257-286,1989.

[14] L.R.Rabiner and B.H.Juang, Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliff, New Jersey, 1993.

[15] S.Katagiri, Speech pattern recognition using neural networks, W.Chou and B.H.Juang (Eds.) Pattern Recognition in Speech and Language Processing, CRC Press, pp.115-147,2003.

[16] Michael K.Brown et.al, Training set Design for Connected Speech Recognition, IEEE Transaction on Signal Processing, Vol.39.No.6.June 1991.

[17] Mohamed Afify and Olivier Siohan,Sequential Estimation With Optimal Forgetting for Robust Speech Recognition , IEEE Transactions On Speech And Audio Processing, Vol. 12, No. 1, January 2004.

[18] Xinwei Li et.alSolving large HMM Estimation via Semidefinite programming, IEEE Transactions on Audio, speech and Language processing, Vol.15,No.8, December 2007.

[19] Jen-Tzung Chien et.al., Predictive Hidden Markov Model Selection for Speech Recognition,IEEE Transactions On Speech And Audio Processing, Vol. 13, No. 3, May 2005.

[20] Rita Singh, Bhiksha Raj et. al.,Automatic Generation of Subword Units for Speech Recognition Systems, IEEE Transactions On Speech And Audio Processing, Vol. 10, No. 2, February 2002.

[21] Hui Jiang et.al., A Minimax Search Algorithm for Robust Continuous Speech Recognition,IEEE Transactions On Speech And Audio Processing, Vol. 8, No. 6, November 2000.

[22] N. Sharma and S. Sardana, "A real time speech to text conversion system using bidirectional Kalman filter in Matlab," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016, pp. 2353-2357, doi: 10.1109/ICACCI.2016.7732406.

[23] Y. H. Ghadage and S. D. Shelke, "Speech to text conversion for multilingual languages," 2016 International Conference on Communication and Signal Processing (ICCSP), 2016, pp. 0236-0240, doi: 10.1109/ICCSP.2016.7754130.

[24] M. B. priya and E. Kannamal, "Investigation of Speech recognition system and its performance," 2020 International Conference on Computer Communication and Informatics (ICCCI), 2020, pp. 1-4, doi: 10.1109/ICCCI48352.2020.9104068.

[25] S. Bano, P. Jithendra, G. L. Niharika and Y. Sikhi, "Speech to Text Translation enabling Multilingualism," 2020 IEEE International Conference for Innovation in Technology (INOCON), 2020, pp. 1-4, doi: 10.1109/INOCON50539.2020.9298280.

[26] N. Andhale and L. A. Bewoor, "An overview of Text Summarization techniques," 2016 International Conference on Computing Communication Control and automation (ICCUBEA), 2016, pp. 1-7, doi: 10.1109/ICCUBEA.2016.7860024.

[27] N. Bhatia and A. Jaiswal, "Automatic text summarization and it's methods - a review," 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), 2016, pp. 65-72, doi: 10.1109/CONFLUENCE.2016.7508049.

[28] P. R. Dedhia, H. P. Pachgade, A. P. Malani, N. Raul and M. Naik, "Study on Abstractive Text Summarization Techniques," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1-8, doi: 10.1109/ic-ETITE47903.2020.087.

[29] R. Mihalcea. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004) (companion volume), Barcelona, Spain.

[30] Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Japan, August.

[31] Mihalcea, P. Tarau, and E. Figa. 2004. PageRank on semantic networks, with application to word sense disambiguation. In Proceedings of the 20st International Conference on Computational Linguistics (COLING 2004), Geneva, Switzerland.

[32] V Gupta and G.s Lehal, "A survey of text summarization extractive techniques", Journal of Emerging Technologies in Web Intelligence, vol. 2, no. 3, pp. 258-268, 2010.

[33] H Luhn, "The automatic creation of literature abstracts", IBM Journal of Research Development, vol. 2, no. 2, pp. 159-165, 1958.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- RPTheautomatedmethodofSummarizationofAudioData1.zip