Application of Extractive Text Summarization Algorithms to Speech-to-text Media

Domínguez M. Victor[4], Fidalgo F. Eduardo[1,3], Rubel Biswas[1,3], Enrique Alegre[1,3], Laura Fernández-Robles[2,3]

[1] Department of Electrical, Systems and Automatics Engineering, Universidad de León, Spain
[2] Department of Mechanical, IT and Aerospatial Engineering, Universidad de León, Spain
[3] Researcher at INCIBE (Spanish National Institute of Cybersecurity), Leon, Spain
[4] Summer Research stay at GVIS Research Group
vdomim00@estudiantes.unileon.es, {eduardo.fidalgo, rbis, enrique.alegre, l.fernandez}@unileon.es

Abstract. This paper presents how speech-to-text summarization can be performed using extractive text summarization algorithms. Our objective is to make a recommendation about which of the six text summary algorithms evaluated in the study is the most suitable for the task of audio summarization. First, we have selected six text summarization algorithms: Luhn, TextRank, LexRank, LSA, SumBasic, and KLSum. Then, we have evaluated them on two datasets, DUC2001 and OWIDSum, with six ROUGE metrics. After that, we have selected five speech documents from ISCI Corpus dataset, and we have transcribed using the Automatic Speech Recognition (ASR) from Google Cloud Speech API. Finally, we applied the studied extractive summarization algorithms to these five text samples to obtain a text summary from the original audio file. Experimental results showed that Luhn and TextRank obtained the best performance for the task of extractive speech-to-text summarization on the samples evaluated.

Keywords: Audio signal summarization, Speech-to-text summarization, Extractive text summarization, Natural Language Processing.

1 Introduction

Every day new platforms, services and applications emerge and manage a large amount of information, the vast majority in multimedia format, such as images, audio or video. Speech is one of the most effective methods of communication. However, it is not very easy to reuse, review or retrieve speech documents if they are contained on an audio signal. Extracting useful insights from an audio file is a hard task, especially if the number of audio files or their length is high. Besides, conversations from audio might include redundant information, e.g. word fragments, fillers or repetitions, together with irrelevant information not related to the topic of interest or the objective followed. For these reasons, automatic summarization of audio files could help a user

to attain the essential information from an audio file without listening to the complete content.

Apart from supporting people with disabilities, speech-to-text applications serve also to transfer the content of an audio file into a readable text document. Converting audio files to text documents could allow managing the information contained in those audios for later processing. A text document can be quickly reviewed, its interesting parts can be easily extracted, and Natural Language Processing (NLP) techniques could be easily applied to extract some knowledge from the text document.

The aim of this work is the automatic summarization of speech documents recorded as audio signals with extractive text summarization techniques. With this objective, we aim at easing and reducing the time required for processing audio information. We have reviewed six text summarization methods, Luhn [1], TextRank [2], LexRank [3], LSA [4], KLSum [5] and SumBasic [6]. We evaluated their performance against OWIDSum [8] and DUC2001 [10] datasets through the following ROUGE metrics; ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU. Next, we have applied the six algorithms to the transcriptions of five audio files from the ICSI-Corpus dataset, computing the ROUGE metrics and making a recommendation about which extractive text summarization technique would be recommended for the task of speech document summarization.
Figure 1 gives an overview of the work that will be presented in this paper.

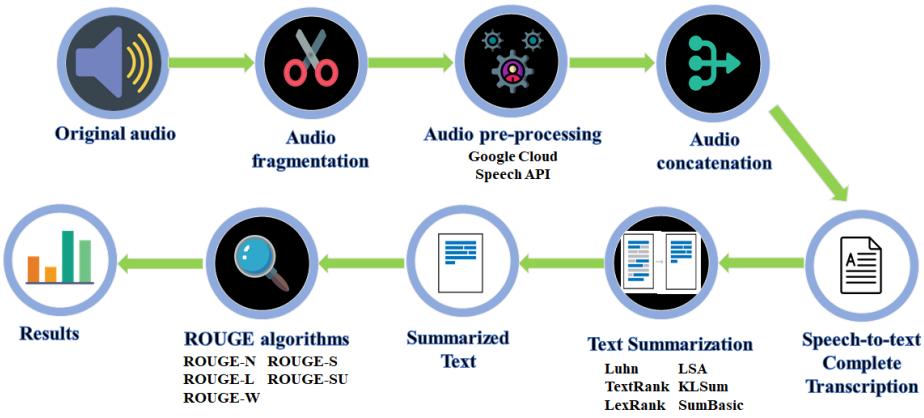

Fig. 1. Speech-to-text summarization through extractive text techniques

The organization of the paper is as follows. In Section 2, we briefly review the state-of-the-art about extractive text summarization techniques and the methods used to transcript audio signals to text. Next, in Section 3, we describe the six extractive text

summarization methods employed in this work. Experimental setup, datasets, experimental results and discussion are presented in Section 4. Finally, in Section 5, the conclusion and future work are discussed.

2 Related Work

2.1 Extractive Text Summarization

Automatic text summarization reduces the size of the text by preserving the information it contains. Text summarization techniques are usually divided into extractive and abstractive [7], depending on how the resulting text is produced. In this work, we are going to work with the extractive approach, i.e. a summary will be generated using a selection of complete sentences from the original text.

Traditionally, text summarization algorithms are based on frequency appearance of words in the document, such as Luhn [1], or based on graphs, such as TextRank [2] and LexRank [3]. Latent Semantic Analysis (LSA) [4] method represents each document as a matrix to which it applies decomposition values. Some schemas are presented based on the probability distribution of words in documents like KLSum [5] and SumBasic [6]. Akanksha et al. evaluated the five already mentioned algorithms in the task of summarizing the content of Tor darknet. With this objective, they proposed OWIDSum [8], a dataset comprising 60 text documents from Tor domains, and grouped into six categories. Working also with DUC2002 [10] dataset, they recommended TextRank [3], since it obtained the better ROUGE metrics on the two datasets evaluated.

Haghighi and Vanderwende [5] explored probabilistic models for the synthesis of multiple documents where different algorithms like KLSum [6], SumBasic [7], TopicSum [11] and HieSum [12], working on DUC2006 dataset with ROUGE metrics [13]. Erkan and Radev [3] presented a stochastic method based on graphs to calculate the relative importance of text units for Natural Language Processing (NLP). The LexRank-based system obtained the best scores on DUC2003 and DUC2004 datasets. Hu et al. [4] proposed a new algorithm based on LSA to evaluate the responses of some students to their teacher by comparing them with a pre-established base response.

In recent years, numerous studies have appeared based on deep learning techniques to address the problem of text summaries using different neural network architectures. Nallapati et al. [14] presented a solution using Recurrent Neural Networks (RNN) where RNN with closed recurrent units, Gated Recurrent Unit (GRU-RNN), were used for encoding and unidirectional GRU-RRN were applied in hidden layers for decoding. The authors evaluated their proposal in three datasets, one of them the DUC Corpus, used in our work, where they obtained a score of 28.35 with ROUGE-1 metric. Chopra et al. [15] also used RRN in their work, but they presented it with a new conditional RNN based on convolutional attention for the abstract synthesis of a sentence. Cao et al. [16] proposed a Convolutional Neural Network (CNN) based on a novel system, named PriorSum, to capture the foregoing summary and concatenate it with the dependent characteristic of the document under a regression framework. This algorithm was applied on three datasets DUC2001, DUC2002 and DUC2004, obtaining 35.98, 36.63 and 38.91 for the ROUGE-1 metric, respectively. In another work, Cao et al. [17]

presented a CNN to generate inlays of sentences to form inlays of documents in the same latent space. A hierarchical framework proposed by Cheng and Lapata [18] where CNN generated the representation and RNN represented the document. Lil et al. [19] proposed a solution based on Variable Autocoder (VAE) and Deep Recurrent Generative Decoder (DRGN) to improve the performance and quality of the abstract summary. The idea is based on the sequence-to-sequence oriented encoder-decoder framework equipped with a latent structure modelling component of variable auto-encoders. As a result, they obtained a metric of 36.25 with ROUGE-1 on the GIGA dataset and a metric of 36.71 with ROUGE-1 on the LCSTS dataset.

Akanksha et al. [9] extended their work on Tor darknet, and they proposed SummCoder, an unsupervised approach for extractive text summarization based on the use of autoencoders and sentence ranking through three different criteria of content, novelty and position. They extended the dataset proposed in [8] up to the 100 samples with two gold summaries, and they obtained state-of-the-art results on their dataset and DUC2002 benchmark.

2.2 Speech to Text

Voice recognition is the interdisciplinary subfield of computational linguistic that develops methodologies and technologies that allow the recognition and translation of the spoken language to text. It is also known as automatic voice recognition or Automatic Speech Recognition (ASR), computer voice recognition, voice to text or Speech-to-Text (STT). Traditionally, the ASR Mel-frequency Cepstral Coefficient (MFCC) [20] or Relative Spectral Transform-Perceptual Linear Prediction (RASTA-PLP) [21] were used as the characteristic vectors and Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) [22] as an acoustic model. Currently, Deep Neural Networks (DNN) models have become very significant in this field like great computational advance and increase of data available for training. Combining these recent techniques with some of the techniques above allows us to obtain better results. Villalba et al. [23] presented a method for detecting the wrong test in a speaker verification system. In some situations, the quality of the signals involved in the verification test is not as good as it would be needed to make a reliable decision, which is based on modelling speech quality measures using Bayesian Networks. Reviewing previous works and using their system to eliminate the untrust-worthy test have achieved a dramatic improvement of the actual detection cost function.

3 Method

A diagram representing the steps of the proposed speech-to-text summarization pipeline is depicted in Figure 1. First, we preprocess the audio signal by segmenting it into small pieces that could be processed by the ASR method. Then, the ASR method is applied to convert the speech documents into text. Next, the resulting text is pre-processed by adding punctuation marks to separate text sequences if it is not automatically provided by the ASR method. After that, in the next step, we apply the six extractive

text summarization methods to obtain a summary of the transcribed speech. Finally, we compute the ROUGE metrics using the automatically generated and the gold summary.

3.1 ASR

The speech processing engine Google Cloud Speech API, included in the python library SpeechRecognition3.8.1, is used in this work. This API allows both to transcribe audios directly recorded with a microphone and to import a series of files as in this work, applying neural network models to audio for speech recognition. The API recognizes 120 languages and variants, and in our work, we used the Synchronous Recognition.

3.2 Text Summarization Techniques

Luhn. One of the most popular text summarization methods is Luhn which is proposed by Luhn [1], and it is based on the frequency of appearance of the words in the text plus the distance that exists between the relevant words which depend on the amount of non-relevant words among relevant ones.

Once the meaningful words have been obtained, the algorithm assigns punctuation to each phrase of the text through a significance factor, which reflects the occurrences of the significant words and the linear distance between them due to non-significant words in the middle.

TextRank. R. Mihalcea and P. Tarau proposed a text summarization algorithm based on graph theory, named TextRank [2], where the sentences are represented through the vertices of the graph. Each of the vertices has a critical value which is determined by taking into account the global information calculated recursively from the entire graph. It is performed by the PageRank algorithm, which is used to calculate the rank, grade or quality of a web page through the number of existing links. In order to use PageRank in the text summarization, each node is represented by a text entity, or sentence, instead of a web page.

Since in the cases of study in our paper no links among pages exist, we defined a similarity between two sentences depending on the number of words that overlap in both.

After obtaining the similarity among all the sentences, a graph is presented where each vertex may not be connected to any other vertex because no similarity is found within the sentences. The edges that connect two vertices will have an associated weight which represents the force with which they are connected.

LexRank. Another graph-based text summarization technique presented by Erkan, G. and Radev is called LexRank [3]. It is based on TextRank, but edges between the vertices are obtained with the cosine similarity scores of sentences that are represented as TF-IDF measurement vectors.

Then a similarity graph is obtained, in which each sentence is represented with a vertex and the cosine similarity between sentences with an edge that holds weight information. A threshold-based approach is used to determine which sentences are considered to be similar to each other.

LSA. Latent Semantic Analysis (LSA) [4] is an automatic technique that obtains the statistical relationship of words in a sentence. The text is parsed into words defined as unique strings of characters and is separated into meaningful sentences.

Subsequently, the text is represented as a matrix where each cell contains the frequency of appearance of the word of such row in the passage indicated by its column. Next, each cellular frequency is weighted by a function that expresses both the importance of the word in the particular passage and information about the type of word in the domain of discourse in general. Finally, Single Value Decomposition (SVD) [24] is applied to the matrix by eliminating the coefficients in the diagonal matrix.

SumBasic. SumBasic [5] is an algorithm that uses a sentence selection component based on the frequency with a component to re-weight the word probabilities in order to minimize redundancy. Firstly, it calculates the probability distribution of the words that appear in the input and, then it computes its average and assigns it a weight equal to this value in each sentence. After that, the best scoring phrase that contains in the most likely word is considered. Finally, the probability of each word in the sentence that has been chosen in the previous step is updated. The algorithm can be applied iteratively to reduce further the length of the summary.

KLSum. The KLSum [6] algorithm for text summarization generates an output summary based on Sum=minKL(Pdoc||Psumm) where Psumm is the empirical distribution of the unigram of the candidate summary, and minKL(P||Q) represents the divergence of Kullback-Lieber. P represents the distribution of the document unigrams and Q is the distribution of the document summary. The criterion tries to generate the summary that closely matches the source document.

3.3 ROUGE metrics

ROUGE [13] is a set of metrics used to determine the quality of text summarization methods. This metric compares the summary result obtained with an algorithm with the one created by human experts, named Gold Summaries. ROUGE metrics determine the number of units that overlap; this is, the number of units that appear in both summaries. The ROUGE metrics used in this work are introduced below.

ROUGE-N. This metric is based on the number of matching “n-grams” between the summary obtained and the reference summary. “n-grams” is a group of “n” units written consecutively, the units can be letters, syllables or words.

ROUGE-L. This method is based on the largest sub-chain (strictly increasing) measurement that is coincident between the candidate summary and the reference summary. It uses the union of the proportion of chain words in the reference summary contained in the candidate summary, and the union of the proportion of chain words in the candidate summary present in the reference summary.

ROUGE-W. This method is similar to ROUGE-L, but it allows differentiating chains of subsequences of different spatial relationships within the original sequence.

ROUGE-S. This method uses bigrams with all the possible jumps between the two bigrams.

ROUGE-SU. This method is an extension of the ROUGE-S that solves the problem by which no value is given to the candidate statement if it does not have any matching pairs of words, even if it has individual words that match. Thus, sentences that do not contain matching pairs of words and do not have matching individual words are given less importance in the metric calculation. To do this, ROUGE-S is extended using the unigram instead of the bigram as the counting unit.

4 Experimentation and results

4.1 Experimental Settings

We made our experimentation with an Intel Core i7 with 16GB of DDR4 RAM. To improve the performance of Google Cloud Speech API, the audio signals were segmented into fragments of 5-10 seconds, which facilitates the transcription of the audio. Once each audio is translated into text, the resulting outputs will be concatenated to obtain the final text. Besides, punctuation marks were manually added in the transcription output to separate the different sequences since ASR does not automatically provide it.

4.2 Datasets

In this work, we have used three different datasets: two text datasets for evaluating the extractive text summarization algorithms, and a speech dataset for the evaluation of the complete method.

DUC2001. Document Understanding Conference 2001 (**DUC2001)** dataset consists of 303 documents that contain news from newspapers grouped into 30 categories, such as Hurricane Andrew, mad cow disease and a plane crash in the city of Sioux.

OWIDSum. The Onion Web Illegal Document Summarization (OWIDSum) is the initial version of the newer Tor Illegal Domain Summarization dataset (TIDSumm) [8].

This dataset consists of texts extracted from Tor darknet domains of the Darknet Usage Text Addressed (DUTA) dataset [25]. DUTA contains 6831 domains of the dark Tor Network and is grouped into 26 categories related to legal and illegal activities. OWID-Sum comprises 60 documents gathered in six categories, i.e. credit cards counterfeit, hacking, drugs selling, money counterfeit, market and crypto-currency, together with two gold summaries for each document.

ICSI Corpus. The International Computer Science Institute Meeting Corpus (ICSI Meeting Corpus) dataset comprises 75 audio conversations between several people with durations of approximately 30 to 70 minutes. It also contains 20 extractive summaries coming from 17 of the audios. Thus, it can be used to calculate ROUGE metrics. We have considered five out of 17 audios from ICSI Corpus in the experiments

4.3 Results and Discussion

Table 1 presents the evaluation of the six algorithms described over DUC2001 dataset using six ROUGE metrics. It can be noticed that Luhn attained the highest performance for four ROUGE metrics, i.e. ROUGE -2, ROUGE -L, ROUGE -W and ROUGE -S, and LexRank did for the other two, i.e. ROUGE -1 and ROUGE –SU with 0.42293 and 0.17252 respectively**.** It can be recommended that for DUC2001, the algorithms that perform better, i.e. obtain higher quality extractive summaries are Luhn followed by LexRank and TextRank.

In the case of OWIDSum dataset, Table 2, TextRank algorithm yielded the best scores for almost all the ROUGE metrics, except ROUGE-1, in comparison to the rest of the methods. Moreover, Luhn and LexRank achieved similar scores for every ROUGE metric.

Table 1. Results obtained in dataset DUC2001. **R-1** and **R-2** stand for the ROUGE-1 and ROUGE-2 respectively. **R-L**, **R-W**, **R-S** and **R-SU** stands for the ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU metrics respectively.

	DUC2001					
	Luhn	TextRank	LexRank	LSA	SumBasic	KLSum
R-1	0.42071	0.40418	**0.42293**	0.35848	0.36031	0.35847
R-2	**0.16814**	0.15399	0.15796	0.11992	0.11243	0.11695
R-L	**0.25817**	0.24616	0.25291	0.21103	0.2132	0.21582
R-W	**0.11247**	0.10645	0.10556	0.09747	0.08175	0.08582
R-S	**0.16818**	0.15427	0.16781	0.11911	0.11605	0.11968
R-SU	0.17218	0.15823	**0.17252**	0.12382	0.12123	0.12447

Finally, Table 3 shows the results obtained after applying the extractive text summarization to the transcription of the five audio files from ICSI Corpus dataset. The best results were obtained by Luhn with 0.57215, 0.35778, 0.3735 0.32403 and 0.32413 for

ROUGE-1, ROUGE2, ROUGE-W, ROUGE-S and ROUGE-SU, respectively. TextRank yielded the best score of 0.3368 for ROUGE-L. Luhn and TextRank achieved similar results for all ROUGE metrics while the rest of methods perform worse, especially KLSum and SumBasic. We can conclude that Luhn is the most suitable algorithm for the analyzed speech documents.

Table 2. Results obtained in dataset OWIDSum.

	OWIDSum					
	Luhn	TextRank	LexRank	LSA	SumBasic	KLSum
R-1	**0.39176**	0.39015	0.38062	0.38515	0.21643	0.34228
R-2	0.24468	**0.26119**	0.21066	0.23293	0.09417	0.19686
R-L	0.34888	**0.35759**	0.32921	0.33783	0.1851	0.30422
R-W	0.17938	**0.18348**	0.15871	0.16776	0.08257	0.14758
R-S	0.17758	**0.18787**	0.14754	0.15428	0.05424	0.12889
R-SU	0.18116	**0.19145**	0.15229	0.15903	0.05781	0.13343

Table 3. Results obtained in ICSI Corpus dataset after transcribing the audio to text files

	ICSI Corpus					
	Luhn	TextRank	LexRank	LSA	SumBasic	KLSum
R-1	**0.57215**	0.56974	0.5441	0.52978	0.28808	0.40229
R-2	**0.35778**	0.35741	0.3382	0.32095	0.16248	0.24832
R-L	0.33572	**0.33684**	0.31893	0.30217	0.16124	0.23187
R-W	**0.03735**	0.03732	0.03463	0.03111	0.01484	0.02155
R-S	**0.32403**	0.32154	0.29613	0.26965	0.06006	0.13555
R-SU	**0.32413**	0.32163	0.26923	0.26976	0.06016	0.13567

5 Conclusions and future work

In this work, we presented an automatic pipeline to summarize audio content by applying extractive text summarization algorithms to a previously transcription from audio to text using ASR.

To this end, traditional text summarization techniques have been reviewed, and from them, six methods have been selected and evaluated on two text summarization datasets, DUC2001 and OWIDSum employing six ROUGE metrics. Experimental results proved that Luhn and the TextRank algorithms mainly obtained the best results in DUC2001 and OWIDSum, respectively.

Then, five speech documents of ICSI Corpus were transcribed with Google Cloud

Speech API ASR. The resulting text documents were evaluated using the same six extractive text summarization algorithms. Again, Luhn and TextRank algorithms yielded the best results, so they are the initial recommendation to solve the automatic speech-to-text summarization task using extractive text summarization techniques.

Possible future work from this study is the extension of the experimentation performed on ICSI Corpus dataset to all of the speech documents that come with gold summaries, or even the manual elaboration of gold summaries for the rest of speech documents in the dataset for further assessment.

The main issue to test this proposal in other public datasets is the lack of extractive gold summaries. As future work, other speech datasets could also be tested after creating appropriate gold extractive summaries. Moreover, predicting the punctuation marks from word sequences would be done automatically.

Acknowledgement

This research is supported by the INCIBE grant "INCIBEC-2015-27359" corresponding to the "Ayudas para la Excelencia de los Equipos de Investigación avanzada en ciberseguridad" and by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 22 and 01.

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 algorithm 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 American 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. Technical report, Microsoft Research.
7. Murthy Vishnu, Vishnu V. B., Mekala, Vijaypal S. P., and Reddy V. (2013). Text Classification 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 Intelligent Systems (APPIS), Las Palmas de Gran Canaria.

9. Joshi, A., Fidalgo, E., Alegre, E., Fernández-Robles, L. (2019). SummCoder: An Unsupervised 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 accessed: 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 Proceedings 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 Representation 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 focusing 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 Abstractive 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-Frequency Cepstral Coefficients For Speaker Recognition: A review. (2015. International Journal 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. 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.
24. 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.
25. Al-Nabki, M., Fidalgo, E., Alegre, E., and Fernández-Robles, L. (2019). ToRank: Identifying the most influential suspicious domains in the Tor network. Expert System with Applications, vol. 123, pp.212–226.