Performance Evaluation of Keyword Extraction Techniques and Stop Word Lists on Speech-To-Text Corpus

Blessed Guda
Department of Electrical and Computer Engineering/AI, Carnegie Mellon University, Africa
blessedg@andrew.cmu.edu

Bello Kontagora Nuhu
Department of Computer Engineering, Federal University of Technology, Nigeria
Corresponding Authors: nuhubk@futminna.edu.ng

James Agajo
Department of Computer Engineering, Federal University of Technology, Nigeria,
james.agajo@futminna.edu.ng

Ibrahim Aliyu
Department of ICT Convergence System Engineering, Chonnam National University, Korea
Corresponding Authors: aliyu@ieee.org

Abstract: *The dawn of conversational user interfaces, through which humans communicate with computers through voice audio, has been reached. Therefore, Natural Language Processing (NLP) techniques are required to focus not only on text but also on audio speeches. Keyword Extraction is a technique to extract key phrases out of a document which can provide summaries of the document and be used in text classification. Existing keyword extraction techniques have commonly been used on only text/typed datasets. With the advent of text data from speech recognition engines which are less accurate than typed texts, the suitability of keyword extraction is questionable. This paper evaluates the suitability of conventional keyword extraction methods on a speech-to-text corpus. A new audio dataset for keyword extraction is collected using the World Wide Web (WWW) corpus. The performances of Rapid Automatic Keyword Extraction (RAKE) and TextRank are evaluated with different Stoplists on both the originally typed corpus and the corresponding Speech-To-Text (STT) corpus from the audio. Metrics of precision, recall, and F1 score was considered for the evaluation. From the obtained results, TextRank with the FOX Stoplist showed the highest performance on both the text and audio corpus, with F1 scores of 16.59% and 14.22%, respectively. Despite lagging behind text corpus, the recorded F1 score of the TextRank technique with audio corpus is significant enough for its adoption in audio conversation without much concern. However, the absence of punctuation during the STT affected the F1 score in all the techniques.*

Keywords: *Keyword, natural language processing, RAKE, textrank, stoplist, speech recognition.*

Received August 13, 2021; accepted August 31, 2022
https://doi.org/10.34028/iajit/20/1/14

1. Introduction

Keyword extraction is the process of extracting the semantically most important set of words from a document. These keywords could be unigrams or multi-grams (key phrases). Manual keyword extraction consumes a lot of time and requires a specialist in the subject matter. These keywords are very useful in various Natural Language Processing (NLP) methods such as text summarization [11], information recommender systems and document tagging [21].

There is currently widespread usage of voice-based Conversational User Interfaces (CUI) such as Alexa [16] Google Assistant [6], and Siri Assistant [20]. These voice-based inputs are more natural to human beings than typing. An example is Speech to Text (STT) dictation, which enables users to read out notes to computers instead of typing. Therefore, future NLP methods need to be adapted for these Speech Recognition texts.

However, all the keyword extraction techniques have only been developed and evaluated on typed texts. The novelty of this work makes two significant contributions. Firstly, as there are limited datasets on speech-to-text corpus, this research aims to develop such a dataset and make it free for further research. Secondly, this work equally evaluates keyword extraction techniques Rapid Automatic Keyword Extraction (RAKE) [17] and TextRank [13] on the data corpus to know which method is most suitable for speech-to-text corpus.

The rest of this paper is organized into a Review of keyword extraction methods, data collection and archiving process, experimental process, results and discussion, and finally, conclusion and recommendations.

2. Keyword Extraction Techniques

Different keyword extraction techniques have been developed based on statistics, and rule-based linguistics, which could be in the supervised or unsupervised machine learning setting [18]. We review some relevant works as follows.

Blei *et al*., [4] propose a Bayesian probabilistic model to extract topics that are considered keywords from a given document. However, It takes the bag-of-words approach, which means that the order of words is ignored. Mihalcea and Tarau [13] developed TextRank, an unsupervised graph-based keyword extraction technique. The graph is set up based on a co-occurrence window of words. The score of a candidate keyword ($S(V_i)$) is determined based on the number of words linked to it and their corresponding scores ($S(V_i)$) as shown in Equation (1):

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(Vj)|} S(V_j) \qquad (1)$$

Where d is a damping factor defined to be 0.85

Furthermore, Rose *et al.*, [17] proposed RAKE, a high-speed keyword extraction technique. Candidate keywords are extracted at the occurrence of stopwords. All the extracted candidate keywords are considered as a co-occurrence window. The score of each candidate keyword (w) is computed as:

$$score(w) = freq(w) + \deg(w) + (\deg(w)/freq(w)) \qquad (2)$$

Kumbhar *et al*., [11] used a supervised keyword extraction technique based on Word2Vec and TextRank. The genism pre-trained word2vec extracts the features from the input documents while a shallow neural network is trained to extract the keywords. Word2Vec ensures that the relationships between words are factored in. Here, we use the Word2Vec representations to build the graph for the TextRank to extract the Keywords. Also, the cosine similarity metric is used for calculating edge weights instead of cosine distance. This is shown in Equation (3):

$$cosine - similarity = x.y/\|x\|.\|y\| \qquad (3)$$

Yao *et al.*, [23] combined TextRank [13] and Term Frequency-Inverse Document Frequency (TF-IDF) in extracting keywords to enable high-frequency word extraction. Wang and Ning [22] proposed a Chinese keyword extraction technique to solve the inability of TextRank to differentiate polysemous words by considering sentence-wise context information. Meanwhile, Pay [14] developed TAKE, an enhanced RAKE [17] to filter out single-word keywords that only appear once and outside the first tenth of the document. TAKE outperformed RAKE and TextRank on all the experiments carried out.

Pay and Lucci [15] developed a keyword extraction method from candidate keywords extracted using RAKE [17], TextRank [13] and TAKE [14] and proposed a heuristic filter is used to select the keywords for all the methods. The scores of keywords that are selected by multiple methods are increased, while single-word keywords extracted by only one method are discarded. A dynamic threshold is used to extract the desired number of keywords. The method outperformed the individual methods in F-Score using the dataset in [13].

Furthermore, Singhal and Sharma [19] used proposed a domain-independent keyword extraction technique using Renyi entropy. The Renyi entropy was used to compute the rank for each candidate word. The Renyi entropy was used for its divergence which is related to how grammatical words are spread in a text document. This is because of the assumption that keywords are not homogeneously distributed in a text, while less relevant words are more homogeneously spread across documents. Campos *et al.* [5] propose a statistical-based lightweight unsupervised key extraction technique. The method is multi-lingual and independent of document, dictionary and domain. However, the study is based on a text document. Similarly, Arts *et al.* [2] proposed keyword extraction technique to assist in patent claim processing and identification of novelty and impact of new technology and impact of claims using cosine similarity between the title, abstract and patent claims with prior patents. Deep learning has also been proposed to extract qualitative data from pathological reports to summarize information and reduce time consumption [9]. However, only three keywords were considered. Additionally, Koizumi *et al.* [10] address focuses on indeterminacy in word selection for automated audio captioning and proposed a transformer-based model with keyword estimation to reduce the indeterminacy. A summary of these methods is shown in Table 1.

As evident from related works, existing keyword extraction techniques have been used on only text/typed datasets. With the advent of text data from speech recognition engines which are less accurate than typed texts, the suitability of keyword extraction is questionable. This paper evaluates the suitability of conventional keyword extraction methods on a speech-to-text corpus. A new audio dataset for keyword extraction is collected using the World Wide Web (WWW) corpus. We choose to experiment with TextRank and RAKE because they are widely used in research as baselines and have open-source implementation libraries [14, 15].

3. Materials and Methods

The World Wide Web (WWW) [8] experimental dataset of computer science abstracts and corresponding manually assigned keywords were used. The WWW dataset is comprised of 1330 documents from the World Wide Web Conference between 2004-2014. The keywords are obtained from the keywords labelled by the human authors. Of the 1330 documents, 82 of the abstracts were read out and recorded on a Gionee S11 lite smartphone to generate the audio corpus stored in M4A format. The audio was collected from two male volunteers (15-25 years old). The audio was converted to Waveform Audio File Format (WAV) format, which is suitable for Speech Recognition. The total read time for the audio collected is 1h:55m:58s at a bitrate of 384kbps. The speech audio is then converted to text. This is because Google's STT engine is the most accurate [8]. The metadata is shown in Table 2, and the dataset generation process is illustrated in Figure 1. The audio and corresponding texts dataset is available at [7].

The collected dataset is used to evaluate the keyword extraction techniques, TextRank and RAKE, to establish the most suitable method for speech-to-text corpus. The TaxtRank uses graph-based algorithms to determine the importance of a vertex in a graph using global information drawn from the entire graph [13]. Formally, given a directed graph $G=(V,E)$ where V and E are set of vertices and edges, respectively. With E as a subset of VXV, and $In(V_i)$ and $Out(V_i)$ as a set of vertices that point to and out of a vertex V_i, the score of a vertex V_i is given as:

$$(V_i) = (1-d) + d \times \sum_{j \in In(V_i)} \frac{1}{|Out(V_i)|} S(V_i) \quad (4)$$

where is d is a damping factor.

The RAKE is an unsupervised keyword extraction method based on the notion that keywords often contain multiple words but seldom contan punctuations. Based on that, a keyword is characterized as “exclusive” or “essential” when extracted from all documents under consideration. On the contrary, a keyword is considered as “general” when is referenced in many given documents but extracted from few [17]. Cosidering referenced document frequency of a keyword, rdf (k), and extracted document frequency of a keyword, edf (k), a keyword exclusivity ($exc(k)$) and essentiality ($ess(k)$) is express as shown in Equations (5) and (6), respectively.

Table 1. Summary of keyword extraction methods.

Authors	Methods	Strengths	Weakness
Blei *et al*., [4]	Latent Dirichlet Allocation	It is useful in the supervised setting when expert-assigned keywords are available	It does not utilize the semantics of the order of words
Mihalcea and Tarau [13]	TextRank	It is useful in the unsupervised setting when expert-assigned keywords are not available.	It does not give high scores to meaningful words with low occurrence
Rose *et al*., [17]	RAKE	It is a fast approach.	It does not give high scores to meaningful words with low occurrence.
Kumbhar *et al*., [11]	Word2Vec and TextRank	It uses the semantic meaning relationship of co-occurring words.	It is a slower approach due to the huge corpus utilized in the word2vec model.
Wang and Ning [22]	a Chinese keyword extraction	Showed a good performance on the Chinese corpus	Suitability for other languages has not been shown
Pay [14]	TAKE: an enhancement on RAKE by using adopting a dynamic threshold for selecting the number of keywords.	Showed improved performances on the RAKE	It does not give high scores to meaningful words with low occurrence.
Pay and Lucci [15]	An ensemble of RAKE, TAKE, and TextRank.	Shows improved performance over the individual methods	It does not give high scores to meaningful words with low occurrence.
Singhal and Sharma [19]	Used Renyi entropy to rank keywords.	Shows improved performance in other entropy-based approaches	Been a statistical approach, it does not give high scores to meaningful words with a low occurrence
Campos *et al*. [5]	YAKE: unsupervised method	multi-lingual and independent from document, dictionary and domain.	Its focused text document
Arts *et al*. [2]	Cosine similarity	Identify patent novelty and impact of new technology.	Its focused text document
Kim *et al*. [9]	BERT	It's practically suitable for pathological keyword extraction	only three keywords were considered, thus its not robust.
Koizumi *et al*. [10]	Transformer-based audio-captioning model with keyword estimation	Reduces indeterminacy in word selection for automated audio captioning	Need to improve the keyword estimation for NLP and image captioning

Table 2. Metadata of the dataset.

Characteristics	Value
Number of Samples	82
Duration	1h:55m:58s
Speech Recognition Engine	Google web speech
Bit rate	384kbps
Number of Speakers	2 (Males)
Domain of Text	Computer Science
Audio Format	WAV

$$exc(k) = \frac{edf(k),}{rdf(k)} \quad (5)$$

$$ess(k) = exc(k) \times edf(k) \quad (6)$$

4. Experimental Process

This section discusses the pre-processing and the design parameters used in the keyword extraction. The evaluation methods used for assessing the performance of the methods are also discussed.

4.1. Pre-Processing

Firstly, all the stop words in the selected StopLists are removed before extracting the keywords. Irrelevant characters such as hyphens, brackets, newline characters and other such types of characters are removed. All the texts are taken to lowercase and are tokenized into words using the Natural Language Toolkit (NLTK) word tokenizer [1, 3].

For TextRank, the chosen Part of Speech (POS) filters are nouns, adjectives and foreign words after lemmatizing the text using a wordnet lemmatizer. The number of keywords to be extracted is not set to a fixed number because the word length of each document varies. The number of keywords is one-third of the graph's words.

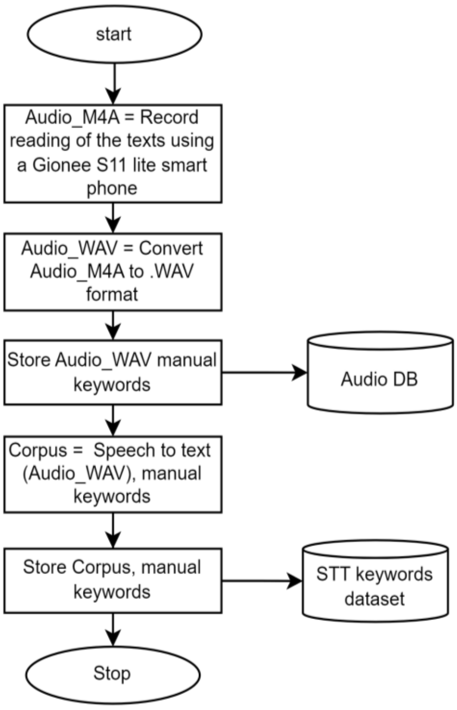


Figure 1. Dataset generation process.

4.2. Evaluation Process

The performance of the extracted keywords is evaluated against the gold standard keywords. It is based on the algorithm proposed by [12] which assigns scores to each extracted keyphrase and also assigns fractional scores if only certain words from the extracted keyphrase match the gold standard keyphrase and vice versa. The process is depicted in Figure 2.

The experiment was carried out with different datasets of StopList to determine the best-performing one. The performance of each model is evaluated using the Recall, Precision, and F1 across the corpus:

$$Recall\ (R) = \frac{scoreofCorrect}{totalExtracted} \tag{7}$$

$$Precision\ (P) = \frac{scoreofCorrect}{numberExtracted} \tag{8}$$

$$F1 = \frac{2.avgRecall.avgPrecision}{avgRecall+avgPrecision} \tag{9}$$

5. Results and Discussion

Firstly, we investigated the optimal hyperparameter for the TextRank. We considered three major hyperparameters- number of iterations, damping factor and convergence threshold as shown in Table 3. The number of iteration was investigated for the values 5, 10 and 15 using a default setting of 0.85 for the damping factor and 1e-5 for the convergence threshold. The optimal R (26.34), P (9.74) and F1 (14.22) were obtained at 10 iterations.

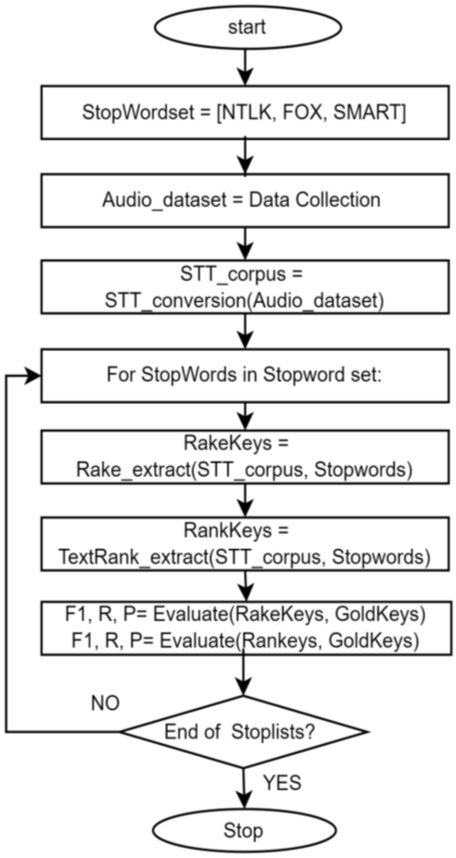


Figure 2. Flowchart of the keyword extraction process.

For the damping factor, the optimal R (26.70), P (9.97) and F1 (14.51) were attained at 0.90. Further, we investigated for optimal convergence threshold using the optimal number of iterations and the damping factor. The optimal threshold of obtained at 1e-3, yielding results of R (26.79), P (9.97) and F1 (14.51).

The results of the keyword extraction on the STT corpus and the originally typed corpus using different StopLists are shown in Table 4. The mean number of keywords extracted is equally recorded. Table 4 indicates that TextRank outperforms RAKE on both the STT and typed corpus, irrespective of the stopword list used. Also, the NLTK stoplist when used with the TextRank algorithm, provides the best performance on the speech-to-text corpus.

It can be observed from Figure 3-a) and 3-b) that the R and P of the different extraction techniques with the different StopLists and corpus datasets are higher in the typed corpus than in the STT. This is because more accurate POS information is assigned- After all, it is punctuated, unlike the STT corpus. However, the R value of the TextRank technique with FOXStopList is the highest in the class of STT. Rake with NLTKStopList, recorded the least value of R in the class of STT corpus, hence less suitable. A similar trend can be observed with P values in Figure 3-b).

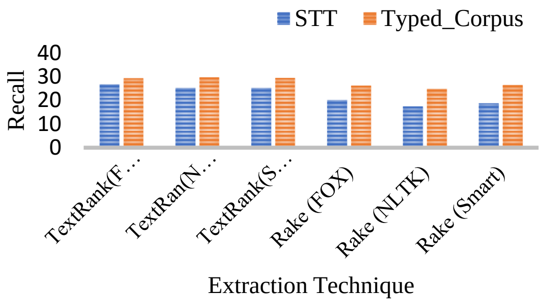
The mean number of keywords extracted is somehow constant for the TextRank technique with the different stoplist datasets for STT and maintains the highest as compared to RAKE using the same corpus as observed in Figure 4. The highest was recorded with NLTKStopList and SmartStopList. Compared with the typed corpus, RAKE extracted fewer keywords with STT corpus with the lowest number with the FOXStopList. However, the difference in the highest number of keywords extracted with the TextRank technique using the two corpora is insignificantly small, this value is recorded in STT when the TextRank technique was used with NLTKStopList and SmartStopList, and it was recorded in the Typed corpus when TextRank was used with NLTKStopList.

Table 3. Hyperparameter tuning of TextRank(FOXStopList).

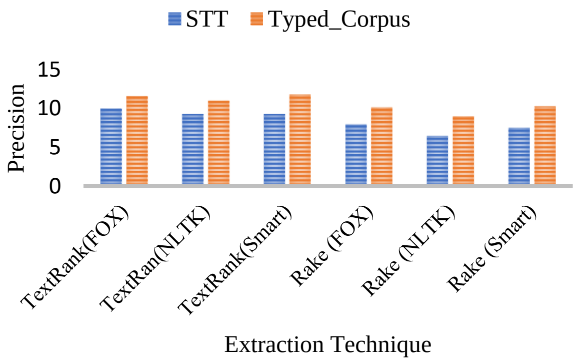
Hyperparameters		R	P	F1	Mean No. of Keywords
Number of Iterations	5	26.21	9.71	14.17	13.79
	10	**26.34**	**9.74**	**14.22**	**13.79**
	15	26.34	9.74	14.22	13.79
Damping Factor	0.75	25.87	9.52	13.91	13.79
	0.80	25.87	9.53	13.92	13.79
	0.90	**26.70**	**9.97**	**14.51**	**13.79**
	0.95	26.70	9.97	14.51	13.79
Convergence threshold	**1e-3**	**26.70**	**9.97**	**14.51**	13.79
	1e-5	26.60	9.92	14.45	13.79
	1e-7	26.70	9.97	14.51	13.79

Table 4. Performance analysis of the keyword extraction on both the speech to text and original corpora.

	Text from STT conversion				Original Typed corpus			
	R	P	F1	Mean No. of Keywords	R	P	F1	Mean No. of Keywords
TextRank(FOXStopList)	**26.70**	**9.97**	**14.51**	13.79	29.23	**11.58**	**16.59**	12.98
TextRan(NLTKStopList)	25.30	9.28	13.60	13.94	**29.70**	10.99	16.04	13.99
TextRank(SmartStopList)	25.30	9.28	13.60	13.94	29.46	11.79	16.84	13.06
Rake (FOXStopList)	20.13	7.93	11.38	12.01	26.14	10.12	14.59	12.82
Rake (NLTKStopList)	17.46	6.48	9.45	12.48	24.77	8.97	13.17	13.65
Rake (SmartStopList)	18.75	**7.48**	10.69	12.15	26.42	10.27	14.79	13.00



a) Recall comparison of TextRank and rake with different corpus.



b) Precision comparison of TextRank and rake with different corpus.

Figure 3. Performance of the extraction techniques.

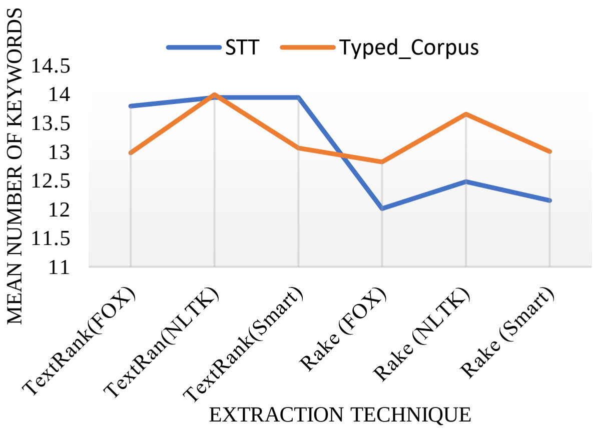


Figure 4. Number of keywords comparison of TextRank and rake with the different corpus.

6. Conclusions

We proposed an audio keyword extraction dataset (with African accents), that is useful for state-of-the-art conversational interfaces that are speech-based rather than typing based. To solve the extraction technique selction problem, Keyword extraction techniques were evaluated with the dataset using RAKE and TextRank. The experimental results indicate the effectiveness of TextRank over RAKE irrespective of the corpus and StopList dataset. Similarly, the TextRank hyperparameter tuning problem was solved through experimentation, with ten iterations, 0.90 damping factor and 1e-3 convergence threshold producing the best outcome. This work will aid future research in the speech recognition domain, particularly with the scarcity of such speech audio with African accents.

For future work, a speech-to-text engine with automatic punctuation can be integrated into the STT conversion process to generate the texts. These punctuated texts would yield more accurate POS tags and extract more accurate keywords. Additionally, the performance of other keyword extraction techniques can be investigated on the dataset.

Conflict of Interest

The authors wish to report that there was no potential conflict of interest.

References

[1] Al-Jarrah A., Al-Jarrah M., and Albsharat A., "Dictionary Based Arabic Text Compression and Encryption Utilizing Two-Dimensional Random Binary Shuffling Operations," *The International Arab Journal of Information Technology*, vol. 19, no. 6, pp. 861-872, 2022.

[2] Arts S., Hou J., and Gomez J., "Natural Language Processing to Identify the Creation and Impact of New Technologies in Patent Text: Code, Data, And New Measures," *Research Policy*, vol. 50, no. 2, pp. 104144, 2021.

[3] Bird S., Klein E., and Loper E., *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, O'Reilly Media Inc, 2009.

[4] Blei D., Ng A., and Jordan M., "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993-1022, 2003.

[5] Campos R., Mangaravite V., Pasquali A., Jorge A., Nunes C., and Jatowt A., "Yake! Keyword Extraction from Single Documents Using Multiple Local Features," *Information Sciences*, vol. 509, pp. 257-289, 2020.

[6] Google Assistant, your own personal Google, https://assistant.google.com/, Last Visited, 2022.

[7] Guda B., Bello Kontagora N., Agajo J., and Aliyu I., STT Dataset, https://drive.google.com/file/d/1N3l4VifpV5BFexYxg9mDaCu1iY0LcFHt/view, Last Visited, 2022.

[8] Këpuska V. and Bohouta G., "Comparing Speech Recognition Systems (Microsoft API, Google API and CMU Sphinx)," *International Journal of Engineering Research and Applications*, vol. 7, no. 03, pp. 20-24, 2017.

[9] Kim Y., Lee J., Choi S., Lee J., Kim J., Seok J., and Joo H., "Validation of Deep Learning Natural Language Processing Algorithm for Keyword Extraction from Pathology Reports in Electronic Health Records," *Scientific Reports*, vol. 10, no. 1, pp. 1-9, 2020.

[10] Koizumi Y., Masumura R., Nishida K., Yasuda M., and Saito S., "A Transformer-Based Audio Captioning Model with Keyword Estimation," *arXiv preprint arXiv:2007.00222*, 2020.

[11] Kumbhar A., Savargaonkar M., Nalwaya A., Bian C., and Abouelenien M., "Keyword Extraction Performance Analysis," *in Proceeding of Conference on Multimedia Information Processing and Retrieval*, San Jose, pp. 550-553, 2019.

[12] Leung A., *Evaluating Automatic Keyword Extraction for Internet Reviews*, Lorraıne Realself INC, 2016.

[13] Mihalcea R. and Tarau P., "Textrank: Bringing Order Into Text," *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Barcelona, pp. 404-411, 2004.

[14] Pay T., "Totally Automated Keyword Extraction," *in Proceedings of IEEE International Conference on Big Data*, Washington, pp. 3859-3863, 20216.

[15] Pay T. and Lucci S., "Automatic Keyword Extraction: An Ensemble Method," *in Proceedings of* IEEE *International Conference on Big Data*, Boston, 2017.

[16] Ram A., Prasad R., Khatri C., Venkatesh A., Gabriel R., Liu Q., and et al., "Conversational Ai: the Science Behind the Alexa Prize," *arXiv preprint arXiv:1801.03604*, 2018.

[17] Rose S., Engel D., Cramer N., and Cowley W., *Automatic Keyword Extraction from Individual Documents*, Wiley Online Library, 2010.

[18] Siddiqi S. and Sharan A., "Keyword and Keyphrase Extraction Techniques: A Literature Review," *International Journal of Computer Applications*, vol. 109, no. 2, pp. 18-23, 2015.

[19] Singhal A. and Sharma D., "Keyword Extraction using Renyi Entropy: A Statistical and Domain Independent Method," *in Proceedings of 7th International Conference on Advanced Computing and Communication Systems*, Coimbatore, pp. 1970-1975, 2021.

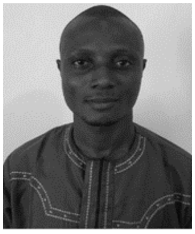
[20] Siri-Apple, https://www.apple.com/siri/, Last Visited, 2022.

[21] Timonen M., Toivanen T., Kasari M., Teng Y., Cheng C., and He L., "Keyword Extraction from Short Documents using Three Levels of Word Evaluation," *in Proceedings of International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*, Barcelona, pp. 130-146, 2012.

[22] Wang X. and Ning H., "Chinese Keyword Extraction Method Based On Context And Word Classification," *in Proceedings of International Conference on Computer Information and Big Data Applications*, Guiyang, pp. 344-347, 2020.

[23] Yao L., Pengzhou Z., and Chi Z., "Research on News Keyword Extraction Technology Based on TF-IDF and Textrank," *in Proceedings of IEEE/ACIS 18th International Conference on Computer and Information Science*, Beijing, pp. 452-455, 2019.

Blessed Guda received his bachelor's degree in computer engineering from the Federal University of Technology, Minna, 2021. He made several contributions to the ITU Focus Group (FG) on ML for 5G and Autonomous Networks (AN). He has research interests in AI for NLP, network security, 5G and autonomous networks and embedded systems. He is a mentor with WINEST Research group and founder of AI4Africa Research group. He received the Mentors Encouragement award from ITU AI/ML in 5G Challenge, 2021. He is currently an AI engineer at Prunny Technologies and also mentor student research projects with ITU FG-AN.



Bello Kontagora Nuhu is a Lecturer in the Department of Computer Engineering at the Federal University of Technology Minna, Nigeria. He obtained his M. Tech in Computer Science and Engineering at the Ladoke Akintola University of Technology Ogbomoso, Nigeria. And had his B.Eng in Electrical & Computer Engineering from the Federal University of Technology Minna, Nigeria in the year 2010. He is currently a doctoral student at the Ahmadu Bello University, Zaria, Nigeria. His research interests include Artificial and Computational Intelligence, localization in sensor networks, Computer/network security, Internet of Things (IoT) and Software-Defined Networking.

James Agajo received a Bachelor of Engineering (B.Eng) degree in electrical and computer engineering from the Federal University of Technology Minna, and a Master's of Engineering (M.Eng.) degree in electronics and telecommunication engineering from Nnamdi Azikiwe University, with a PhD in telecommunication and computer engineering from Nnamdi Azikiwe University. Dr James Agajo is presently an associate professor and the Head of Department, Computer Engineering with the Federal University of Technology Minna. Presently, he is a visiting professor at I.C.T University U.S.A, Nile University Abuja, Baze University Abuja, Kebbi State University of Science and Technology Kebbi, Nigeria, University of Pretoria, South Africa, Federal University of Petroleum Resources Effurun, Nigeria.

Ibrahim Aliyu received his PhD in Computer Science and Engineering from Chonnam National University, South Korea, in 2022. He also holds a B.Eng and M.Eng Degrees in Computer Engineering at the Federal University of Technology, Minna, Nigeria, in 2014 and 2018, respectively. He is currently a Postdoc researcher at Hyper Intelligence Media Network Platform Lab, Department of ICT Convergence System Engineering, Chonnam National University, Gwangju, South Korea. His research focuses on source routing, in-network computing and cloud-based computing for massive metaverse deployment. His other research interests include Federated Learning, data privacy, Network Security and AI for autonomous networks.