TEST Engineering & Management

March – April 2020
ISSN: 0193-4120 Page No. 6389 - 6396

Graph based Automatic Odia Keyword Extraction from Text Document

Mamata Nayak
Faculty of Engineering and Technology,Siksha O Anisandhan(Deemed to be University)
Email: mamatanayak@soa.ac.in

Nilima R. Das and Usha M. Mohapatra
Faculty of Engineering and Technology,Siksha O Anisandhan(Deemed to be University)
Email: {nilimadas, ushamohapatra}@soa.ac.in

Article Info
Volume 83
Page Number: 6389 - 6396
Publication Issue:
March - April 2020

Article History
Article Received*: 24 July 2019*
Revised*: 12 September 2019*
Accepted*: 15 February 2020*
Publication*: 02 April 2020*

Abstract:
Keywords are the words in a document that create a center of attention to readers for better understanding and comprehension about the subject. The applications of keywords are enormous. Now a days, there is a vast availability of Odia text and not much research work has been published for keyword extraction from documents written in odia script. This paper proposes a novel unsupervised undirected weighted graph based approach for extracting keywords from odia text. Importance of extracted keywords is analyzed by computing the weights of the nodes in the graph generated from the text. The performance of the proposed technique has been reported in terms of precision, recall and F-measure. It is observed from the experimental result that the proposed graph based technique can effectively extract keywords from Odia text with minimum computational complexity because of its implementation simplicity.

Keywords: *Graph based model, Keyword extraction, Odia*

I. INTRODUCTION

Text is unstructured data within digital forms, which may contain underlying information. Hidden evidence in text can be terms including keywords and key phrases. Keywords are the collection of words that provide a compact representation of a document. Also the keywords support anchors as hyperlinks between documents that enable users to quickly access related materials. These words are extensively used for automatic indexing, topic modeling, questionanswering system, summarization of document, automatic clustering as well as classification and many more. Since finding keywords physically is a prolonged process and costly, the computerized techniques can be used which can save time and economy. Furthermore, the importance of keyword extraction for Odia script is also plays a vital role. Odisha was recognized as an individual state on 1st April 1936, during the British rule and it consisted of the places where generally Odia is spoken. Odia is one of the primary languages in India. Odia is one of the languages to which the Government of India has awarded the distinction of classical language. Classical language status isgiven to languages which have a rich heritage and independent nature. There is hardly any research work found in the literature based on Odia language. In this article the authors have tried to develop a system that can extract keywords from a given Odiatext which can be used for text summarization and other purposes.

The organization of the remaining part of this research article is described as

follow: Section 2 illustrates the existing literature on graph based approach as well as keyword/key phrase extraction for different languages, Section 3 describes the model used in this literature. As this literature is the first attempt towards finding of keywords, the data corpus used for implementation is illustrated in Section 4. Section 5 explains the models through examples with experimental results, and conclusion is given in section 6.

II. LITERATURE SURVEY

Currently there have been a great number of researches on keyword extraction. The algorithms of keyword extraction can be broadly divided into two categories, supervised and unsupervised. Some examples of unsupervised algorithms are TF-DF(Term Frequency-Inverse Document Frequency)[1], Text-rank[1]and LDA(Latent Dirichlet Allocation)[2]. The authors in[2]have combined the supervised technique SVM with unsupervised algorithms to optimize the the keyword extraction process. They have used SVM Ranking to rank the candidate keywords after extracting their important features using the unsupervised techniques. In[3] the authors have projected a graph-based key-phrase extractor. They have used a basic and straightforward graph-based syntactic representation for text and web documents. For each different word only one vertex is formed irrespective of the number of times it is present in the text. Thus, every vertex in the graph is distinctive. Using directed graph it tries to extract multi-word key phrase from the text by giving importance to the order of word occurrence. The work in [4]uses a Key-Rank approach to extract proper key phrases from English documents. It explores every possible candidate for the key phrases from the text and then assigns them some ranks to make a decision for the top N key phrases. It uses a sequential pattern mining method with gap constraints in order to extract key phrase candidates for assigning Key-Rank. An effectiveness evaluation measure pattern frequency with entropy is also proposed for ranking the candidate key phrases. The work in [5]suggests an unsupervised graph based keyword extraction method. The method is called as Keyword Extraction using Collective Node Weight which fixes the importance of a keyword by using a variety of effective factors. This method uses Node Edge rank centrality with node weight that depends on different parameters like frequency, centrality, position and strength of neighboring nodes. These factors are used to calculate the significanceof a node. The implementation of the model is divided into 4 stages, such as preprocessing, textual graph representation, node weight assignment and keyword extraction. In pre-processing stage the meaningless symbols are removed from tweets so that useful keywords can be taken out. In the second stage a graph is constructed in which one vertex represents one token. For each token there is a vertex in the graph. The edges are constructed for pairs of tokens present in the original texts without changing the order of appearance in the text. The model uses different significant parameters to estimate weight of a node based on the above mentioned parameters. The last phase is the keyword extraction which involves recognizing keywords from a text that can properly characterize the subject of the given text. In[6], the authors have used graph convolutional networks for text categorization. After building a single text graph for a corpus based on co-occurring words and related words, a text graph convolutional network is considered as the corpus. The text network is initialized with one-hot representation for word and document to learn the embeddings in the document. The generated features are trained with a

March – April 2020
ISSN: 0193-4120 Page No. 6389 - 6396

supervised learning algorithm to classify new unlabeled documents.

III. UNSUPERVISED TECHNIQUES FOR RANKING AND KEYWORD EXTRACTION

A. Graph based Text-Rank model

Text-Rank is a graph based model which uses web-based page ranking method. It considers the document as a graph and every node in the graph represents a candidate for the keyword to be extracted from the document. An edge is formed between two words if they are present in the same sentence. The page-rank algorithm is used to calculate the weight of every node. The iterative formula for calculating the weight(rank) of every node is described as:

$$\boldsymbol{W(Vi) = (1-f) + f\sum_{j\in\in(V_i)} W\left(V_j\right)\frac{1}{|OUT(V_j)|}}(1)$$

V represents the set of vertices. E denotes the set of edges. N is the total number of words. f is damping factor which is the probability of jumping from a given vertex to another random vertex in the graph. Its value is set between 0 and 1. Generally, the damping factor is taken as 0.85, as used in TextRank implementation. $OUT(V_j)$represents the set of outgoing links of node *Vj*. $|OUT(V_j)|$ is the out degree of *Vj*. *In(Vi)* is the set of inbound links of node *Vi*. *W(vi)* is the weight of node vi. The top vertices having higher ranks are considered as the keywords. Finally keywords are collapsed into multi-word key phrases.

B.TF-IDF

TFIDF stands for term frequency–inverse document frequency. It tries to establish the importance of a word in a text using numerical statistics. It can be used as weighting factors for retrieving information, mining text and modeling. The tf–idf value raises proportionally as the number of times a word present in the text and is offset by the number of documents in the corpus that contains the word. tf–idf is one of the most accepted term-weighting method used nowadays. 83% of text-based recommender systems in digital libraries use TF–IDF[1].

In this article the authors have used a graph based Text-Ranking method for keyword extraction from a given Odia text. The Text-Ranking method has been used because of it's simplicity, popularity and efficiency.Text-Rank is an algorithm based on Page-Rank, which often used in keyword extraction and text summarization.Page-Rank is for webpage ranking, and Text-Rank is for text ranking. The webpage in Page-Rank is the text in Text-Rank, so the basic idea is the same. The proposed method uses a graph based Text-Rank method to calculate the rank of the words present in the text. It estimates the importance of a node from its linked neighbors and their neighbors. The algorithm used here can be described as:

Step 1 : Preprocessing of the document, i.e.,for each sentence only consider the nouns and objects and discard the rest of the text

Step 2 : Form windows of size k each

Step 3 : For each ordered pair of words in the window a directed edge is constructed

Step 4 : Weight of each node is calculated as equation 1

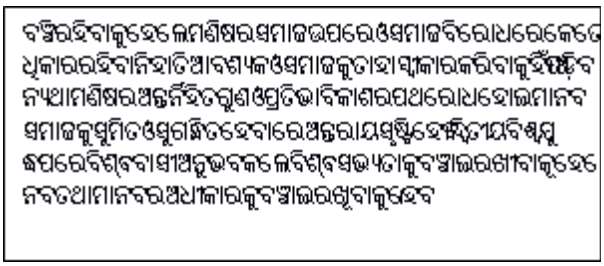
Step 5 : Repeat step 4 for some finite number of times

Published by: The Mattingly Publishing Co., Inc

March – April 2020
ISSN: 0193-4120 Page No. 6389 - 6396

Step 6 : After final iteration remove the words from the list having weights less than a pre-decided threshold value.

The text used as input to the system is shown below.



After extracting the stop words the text contains the following words. The stop words are the words in a sentence are not useful to determine the importance. Only noun and verbs are considered in a sentence.

'ବଞ୍ଚି', 'ମଣିଷ', 'ସମାଜ', 'ସମାଜ', 'ବିରୋଧରେ', 'ଅଧିକାର', 'ନିହାତି', 'ଆବଶ୍ୟକ', 'ସମାଜ', 'ସ୍ୱୀକାର', 'ପଡ଼ିବ', 'ଅନ୍ୟଥା', 'ମଣିଷ', 'ଅନ୍ତର୍ନିହିତ', 'ଗୁଣ', 'ପ୍ରତିଭା', 'ବିକାଶର', 'ପଥରୋଧ', 'ମାନବ', 'ସମାଜ', 'ସୁମିତ', 'ସୁଗନ୍ଧିତ', 'ଅନ୍ତରାୟ', 'ସୃଷ୍ଟି'

IV. Implementation and Experimental Results

Figure 1 shows the transliteration of the selected words. The first column represents the node number, the second column represents the corresponding word for that node and the third column is the transliteration of that word in English. The words are grouped into a fixed number of windows. The size of each window is taken as 3 here. The windows formed are: ('ବଞ୍ଚି', 'ମଣିଷ', 'ସମାଜ'), ('ମଣିଷ', 'ସମାଜ', 'ସମାଜ'), ('ସମାଜ', 'ସମାଜ', 'ବିରୋଧରେ') and so on. A graph is created in which every word represents a node. If a particular word appears more than once only one node is created for that word. That means if a word is new in text then a node is formed in the Graph. An edge is added for two nodes(words) if they co-occur within a certain window. Figure 2 shows the Graph constructed for the words that are selected after performing step 1 of the algorithm. The graph has been generated using the NetworkX package of python.

NN	Representation	Transliteration in English
0	ସମାଜ	samaja
1	ମଣିଷ	manisa
2	ସ୍ୱୀକାର	swikara
3	ଅନ୍ୟଥା	anyatha
4	ପ୍ରତିଭା	pratibha
5	ବିରୋଧରେ	birodhare
6	ଗୁଣ	guna
7	ଅଧିକାର	adhikara
8	ବିକାଶର	bikasara
9	ସୃଷ୍ଟି	strusti
10	ଅନ୍ତର୍ନିହିତ	antarnihita
11	ଆବଶ୍ୟକ	abashyaka
12	ନିହାତି	nihati
13	ପଥରୋଧ	patharodha
14	ବଞ୍ଚି	banchi
15	ମାନବ	manaba
16	ସୁମିତ	sumita
17	ସୁଗନ୍ଧିତ	sugandhita
18	ଅନ୍ତରାୟ	antaraya
19	ପଡ଼ିବ	padiba

Figure 1. Transliteration of the selected words

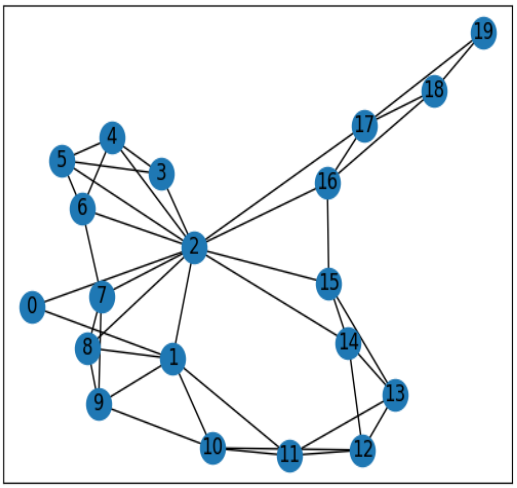


Figure 2 Graph of the matrix shown in fig 2

Figure 3 shows matrix that contains the count of inbound links to a particular node from other nodes. For every node there is a row in the matrix. Every column for that row shows the number of inbound links

Published by: The Mattingly Publishing Co., Inc

March – April 2020
ISSN: 0193-4120 Page No. 6389 - 6396

from the node present in the column to the node in that row. Any pair of words in a window is considered to have an edge between them. Figure 4 shows the calculated weights of the nodes after the first iteration of the step 4 of the algorithm. Figure 5 shows the calculated weights of the nodes after the second iteration. It can be seen that the weights are changed at the end of every iteration.

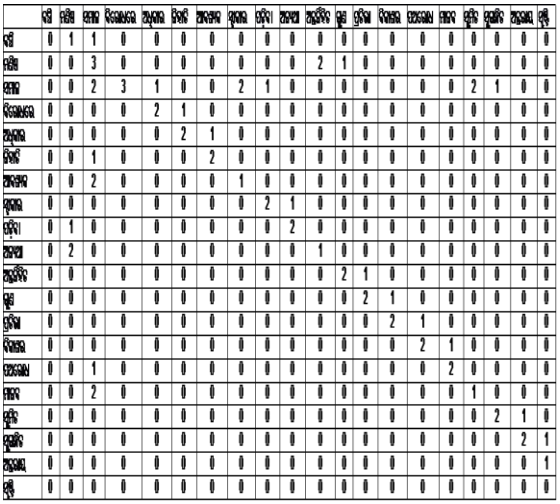


Figure 3 Matrix representing the inbound and outbound links

ବଞ୍ଚି	[0.43333333]
ମଣିଷ	[1.2125]
ସମାଜ	[3.125]
ବିରୋଧରେ	[1.0]
ଅଧିକାର	[1.0]
ନିହାତି	[0.7875]
ଆବଶ୍ୟକ	[0.575
ସ୍ୱୀକାର	[1.0]
ପଡ଼ିବ	[0.77]
ଅନ୍ୟଥା	[0.85833333]
ଅନ୍ତର୍ନିହିତ	[1.0]
ଗୁଣ	[1.0]
ପ୍ରତିଭା	[1.0]
ବିକାଶର	[1.0]
ପଥରୋଧ	[0.7875
ମାନବ	[0.575
ସୁମିତ	[1.0]
ସୁଗନ୍ଧିତ	[1.14166667]
ଅନ୍ତରାୟ	[0.575]
ସୃଷ୍ଟି	[0.15]

Figure 4 Weights of the nodes after first iteration

ବଞ୍ଚି	[0.43333333]
ମଣିଷ	[1.2125]
ସମାଜ	[3.125]
ବିରୋଧରେ	[1.0]
ଅଧିକାର	[1.0]
ନିହାତି	[0.7875]
ଆବଶ୍ୟକ	[0.575]
ସ୍ୱୀକାର	[1.0]
ପଡ଼ିବ	[0.37]
ଅନ୍ୟଥା	[0.85833333]
ଅନ୍ତର୍ନିହିତ	[1.0]
ଗୁଣ	[1.0]
ପ୍ରତିଭା	[1.0]
ବିକାଶର	[1.0]
ପଥରୋଧ	[0.7875]
ମାନବ	[0.575]
ସୁମିତ	[1.0]
ସୁଗନ୍ଧିତ	[1.14166667]
ଅନ୍ତରାୟ	[0.575]
ସୃଷ୍ଟି	[0.15]

Figure 5 Weights of the nodes after 2nd iteration

ସମାଜ	[2.76621013]
ମଣିଷ	[1.44976676]
ସ୍ୱୀକାର	[1.13985002]
ଅନ୍ୟଥା	[1.08970608]
ପ୍ରତିଭା	[0.9184044]
ବିରୋଧରେ	[0.91475251]
ଗୁଣ	[0.89125502]
ଅଧିକାର	[0.89114075]
ବିକାଶର	[0.86195244]
ଅନ୍ତର୍ନିହିତ	[0.85345979]
ଆବଶ୍ୟକ	[0.83162975]
ନିହାତି	[0.82452051]
ପଥରୋଧ	[0.75884009]
ବଞ୍ଚି	[0.68552528]
ମାନବ	[0.64382644]
ସୁମିତ	[0.400325]
ସୁଗନ୍ଧିତ	[0.334875]
ଅନ୍ତରାୟ	[0.21375]
ପଡ଼ିବ	[0.377865]
ସୃଷ୍ଟି	[0.15]

Figure 6 Weights of the nodes after final iteration

After a fixed number of iterations the algorithm terminates. Figure 6 shows the calculated weights of the nodes after the final iteration.

The nodes are arranged according to their weights.

ସମାଜ	[2.76621013]
ମଣିଷ	[1.44976676]
ସ୍ୱୀକାର	[1.13985002]
ଅନ୍ୟଥା	[1.08970608]
ପ୍ରତିଭା	[0.9184044]
ବିରୋଧରେ	[0.91475251]

Figure 7 Weights of the nodes after applying threshold

Published by: The Mattingly Publishing Co., Inc

March – April 2020
ISSN: 0193-4120 Page No. 6389 - 6396

The final table contains the words having higher weights. The words are selected based on some threshold value, which is considered here as 0.9. The fig.7 shows the final keywords with their corresponding weights.

V. RESULT ANALYSIS

The proposed approach is used for indexing of documents written in Odia language. As no dataset is available for the said language, three different datasets have been created relevant to: geography, history and science referred as Doc1, Doc2 and Doc3.

After preprocessing of the text, each dataset contains 5000 words correspondingly. Due to unavailability of the database the keywords are first selected manually to be compared with the predicted keywords. Some persons were invited to identify the keywords manually from the documents. The intersection of the sets identified by the persons for each document is taken into consideration. The resultant keywords are compared with the experimentally extracted keywords. The results of the experiments executed on these three documents are analyzed to test the performance of the proposed method.

The measures used for evaluation of the keywords extracted by the proposed approach relevance to manually assigned keywords are the Precision, Recall and F-measure defined in equation 2, 3 and 4 respectively.

$$Precision\ (P) = \frac{\#TP}{\#N} \qquad (2)$$

$$Recall\ (R) = \frac{\#TP}{\#M} \qquad (3)$$

$$F\ score = 2 \times \frac{P \times R}{P + R} \qquad (4)$$

$\#N$: Total Number of keywords identified by the system

$\#M$: Total Number of keyword manually identified

The terms used in the above equations are True Positive (TP), True Negative (TN), False Positive (FP) and FN (False Negative). The TP is set of keywords detected by the algorithm which are present in the set of keywords detected manually, FN is the set of manually detected keywords not detected experimentally and FP is the set of words not defined as keyword manually but detectedexperimentally. FN is the set of manually detected keywords not detected experimentally Using the values of TP, TN, FP, FN, total keyword detected manually and total keywords predicted experimentally the precision, recall and F-measure are calculated based on the equations 2, 3 and 4. Table I shows the values of these terms. The experimental values shown in this table are generated when the size of each window is taken as 3. There are 3 rows in the table. Each row shows results for a particular document. The table II shows a comparison between the values of precision percentage, recall percentage and F-measure percentage for the results obtained with different window size. First column represents the percentage for a window size 3(w=3) and the second column shows the percentage for a window size 5(w=5). It can be observed that the results are improved with increased window size.

TABLE I MANUAL VERSUS PREDICTED RESULTS

Odia text Docs	Actual keywords found manually	Total keywords from EXP.	Actual keywords from EXP. (TP)	Falsely predicted keywords (FP)	Missed Keywords (FN)
Doc1	450	510	430	80	20
Doc2	400	460	360	100	40
Doc3	530	580	500	80	30

TABLE II MANUAL VERSUS PREDICTED RESULTS

Docs	Precision in %		Recall in %		F-measure in %	
	w=3	w=5	w=3	w=5	w=3	w=5

Published by: The Mattingly Publishing Co., Inc

Doc1	84	88	95	97	89	93
Doc2	78	84	90	93	83	89
Doc2	86	91	94	95	90	94

VI. Conclusion

In this paper the authors established an unsupervised graph-based Text-Ranking method for extracting keywords from a given Oriya text. The empirical results suggest that the proposed approach has the best precision. It's step complexity is O(N*I) which is linear, where N represents the number of nodes in the graph and I is the total number of iterations. Therefore it is better than other supervised algorithms which have high computational complexity because of their complex training process. The method used here is also language independent and for this it can be used with any language. The main disadvantage of this method is the procedure that is used for the removal of the stop words. The removal of the stop words is a crucial step in the algorithm. If all the stop words can be removed from the text the algorithm can be implemented efficiently to find the keywords. The authors have used Odia vocabulary to form a database of the stop words. However, some words are there like'ପଡ଼ିବ'(padiba) and 'ନିହାତି'(nihati) which are extracted as keywords but literally they are not keywords. Hence, these kind of words should have been removed from the text in the preprocessing stage. In the future work, the authors would try to remove all such words from the text before finding the keywords. The proposed method is also not capable of finding key-phrases which are combination of keywords from a given text. So this may also be considered as an extended research direction for the authors.

REFERENCES

[1] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts."

[2] Q. Wang, V. S. Sheng, and X. Wu, "Document-specific keyphrase candidate search and ranking," *Expert Syst. Appl.*, vol. 97, pp. 163–176, May 2018.

[3] M. Litvak, M. Last, H. Aizenman, I. Gobits, and A. Kandel, "DegExt - A language-independent graph-based keyphrase extractor," in *Advances in Intelligent and Soft Computing*, 2011, vol. 86, pp. 121–130.

[4] X. Cai and S. Cao, "A keyword extraction method based on learning to rank," in *Proceedings - 2017 13th International Conference on Semantics, Knowledge and Grids, SKG 2017*, 2017, vol. 2018-January, pp. 194–197.

[5] S. K. Biswas, M. Bordoloi, and J. Shreya, "A graph based keyword extraction model using collective node weight," *Expert Syst. Appl.*, vol. 97, pp. 51–59, May 2018.

[6] L. Yao, C. Mao, and Y. Luo, "Graph Convolutional Networks for Text Classification," Sep. 2018.

[7] R. B. Tchoua *et al.*, "Creating Training Data for Scientific Named Entity Recognition with Minimal Human Effort."

[8] T. Sexton, M. Hodkiewicz, M. P. Brundage, and T. Smoker, "Benchmarking for Keyword Extraction Methodologies in Maintenance Work Orders."

[9] S. Beliga, "Keyword extraction: a review of methods and approaches."

[10] M. Timonen, T. Toivanen, Y. Teng, C. Cheng, and L. He, "Informativeness-based keyword extraction from short documents," in *KDIR 2012 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, 2012, pp. 411–421.

[11] M. Timonen, T. Toivanen, M. Kasari, Y. Teng, C. Cheng, and L. He, "Keyword Extraction from Short Documents Using Three Levels of Word Evaluation," 2013, pp. 130–146.

[12] D. Zhao, N. Du, Z. Chang, and Y. Li, "Keyword extraction for social media short text," in *Proceedings - 2017 14th Web Information Systems and Applications Conference, WISA 2017*, 2018, vol. 2018-January, pp. 251–256.

[13] J. Rafiei-Asl and A. Nickabadi, "TSAKE: A topical and structural automatic keyphrase extractor," *Appl. Soft Comput. J.*, vol. 58, pp.

620–630, Sep. 2017.

[14] H. Mirisaee, E. Gaussier, C. Lagnier, and A. Guerraz, "Terminology-based Text Embedding for Computing Document Similarities on Technical Content," Jun. 2019.

[15] K. Sarkar, M. Nasipuri, and S. Ghose, "Machine learning based keyphrase extraction: Comparing Decision Trees, Naïve Bayes, and Artificial Neural Networks," *J. Inf. Process. Syst.*, vol. 8, no. 4, pp. 693–712, 2012.

[16] M. Ruhul Amin and M. Chakraborty, "Algorithm for Bengali Keyword Extraction," in *2018 International Conference on Bangla Speech and Language Processing, ICBSLP 2018*, 2018.

[17] C. Caragea, F. Bulgarov, A. Godea, and S. Das Gollapalli, "Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach."

[18] H. M. M. Hasan, F. Sanyal, D. Chaki, and M. H. Ali, "An empirical study of important keyword extraction techniques from documents," in *Proceedings - 1st International Conference on Intelligent Systems and Information Management, ICISIM 2017*, 2017, vol. 2017-January, pp. 91–94.

[19] H. H. Kian and M. Zahedi, "Improving Precision in Automatic Keyword Extraction Using Attention Attractive Strings," *Arab. J. Sci. Eng.*, vol. 38, no. 8, pp. 2063–2068, 2013.

[20] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic Keyword Extraction from Individual Documents," in *Text Mining: Applications and Theory*, John Wiley and Sons, 2010, pp. 1–20.

[21] M. Hanumanthappa, M. N. Swamy, and N. M. Jyothi, "Automatic Keyword Extraction from Dravidian Language," 2014.

[22] J. R. Thomas, S. K. Bharti, and K. S. Babu, "Automatic keyword extraction for text summarization in e-newspapers," in *ACM International Conference Proceeding Series*, 2016, vol. 25-26-August-2016.