



An Approach for Multi-Document Text Summarization Using Extreme Learning Machine and LexRank

Wedad Abdul Khuder Naser

Department of Computer Science, College of Education,

Mustansiriyah University

Iraq.

ABSTRACT

Due to the exponential growth of online textual data and the variety of its sources, there is a need to produce an accurate text summary with the least time and effort. Extractive multi-document text summarization methods are intended to automatically generate summaries from a document collection, covering the main content and avoiding redundant information. In this study, a new method for extractive multi-document summarization has been proposed based on the combination of supervised and unsupervised learning. Throughout the supervised learning, a set of seven features was extracted from each sentence in the document collection and introduces to the Extreme Learning Machine (ELM), to distinguish between important and unimportant sentences. A graph of important sentences was generated and assigned scores by the LexRank algorithm during the unsupervised learning. The performance of the proposed method on the DUC-2002 dataset was calculated using ROUGE evaluation metrics. The proposed method achieved a 0.47472 ROUGE for 200-word summaries and 0.54641 ROUGE for 400-word summaries.

Keywords: Summarization , Duc-2002 , Lexrank , Extreme Learning Machine.

1. INTRODUCTION

With the growing availability of online documents, the volume of electronic documents on the Internet has increased with extraordinary speed [1]. This growth of information made it very difficult for humans to take advantage of information in its natural state. One way to attack this issue is by generating summaries [2]. Automatic Document Summarization (ADS) is a task of Natural Language Processing (NLP). ADS is the data science problem of creating a short, accurate, and fluent summary from a longer document or from a collection of documents. Summarization methods are greatly needed to consume the ever-growing amount of text data available online.

[3]. The main objective of an ADS system is to produce a summary that includes the main ideas in the input document in less space and to keep repetition to a minimum. Documents can be of various types such as video, text and speech [4].

Automatic Text summarization (ATS) deals with generating a summary of text documents. ATS can be produced from a single document or multi document. Single Document Summarization (SDS) is The process of generating a representative summary from the single input text, while Multi Document Summarization (MDS) is summary generated from two or more related text documents. MDS is more complicated than SDS, for two reasons: information overload and redundancy. The redundancy occurs because the summarized documents share the same topics. Reducing redundancy produces a high-quality summary [5].

The technique of ATS can be classified into two major categories, abstractive and extractive. Abstractive summarization required deep understanding of NLP strategies to analyze the sentences because several changes have to be made to the gist of the text. It utilizes generative techniques which can create meaningful sentences and at the same time keep the semantics of the original text fed to them. Whereas the Extractive summarization is relatively effortless, no modification will be applied to the sentences that are embedded in the created summary [6]. In addition, summaries can be generic or query-based. Most researchers' studies have been on the basis of generic summarization. In generic summarization a whole sense of document content is presented without any prior knowledge. On the other hand, In the query- based summarization, where the information presented in it should have some relevance with a given query or topic [7].

The main contribution of this paper is the combination of supervised and unsupervised learning for the MDS task by introducing a new method for extractive generic MDS. Based on extracting seven features from each sentence in the document collection. ELM was used to classify the sentences as important or unimportant depend on the extracted feature set. In the next stage, the graph of important sentences was created and assigned scores using LexRank algorithm.

2.RELATED WORK

Summary creation based on sentences or phrases scoring is the most popular method in extractive summarization. There are three methods for scoring: Word scoring, sentence scoring and graph scoring [8]. In the word scoring methods a higher score is given to the word based on its importance in the sentence such as the repetition of the word in the text, places and proper nouns. In the sentence scoring methods several factors influence the importance of the sentences such as sentence length where sentence with larger size are considered more important than smaller one. Also, sentences starting with terms like "Conclude","Finally" and so on are defined as significant sentences. In addition the position of the sentence in the text is an important factor to include the sentence in the created summary. Finally, graph scoring methods usually used in ATS as they provide very effective solutions [9].

In 2015 Premjith et al. Use Maximum relevance and minimum redundancy for generic extractive multi document summarization. They proposed extracting four features for every sentence and suggest a metaheuristic optimization depend on solution population with multiple objective functions that take care of both statistical and semantic aspects of the documents[10]. In 2017 MirShojaee et al. A biogeography - based optimization algorithm (BBO), which is a metaheuristic algorithm used for extractive text summarization. This method is tested on a set of Doc’s standard documents in 2002 and is analyzed using ROUGE software[11]. In 2017 Yasunaga employ a Graph Convolutional Network (GCN) on the relation graphs, with sentence embedding obtained from Recurrent Neural Networks as input node features. Through a multiple layer-wise propagation, Then a greedy algorithm applied to extract important sentences with less redundancy [12]. In 2018 Cesar et al. proposes two algorithms based on the hybridize the metaheuristic of global best harmony search algorithm and the LexRank graph based algorithm called LexGbhs and GbhsLex. The system composed of two objectives functions coverage and diversity that must be optimized [13]. In 2020 Jinming Zhao Proposed SummPip: an unsupervised method for MDS, based on converting the original documents to a sentence graph, taking both linguistic and deep representations into account, then apply spectral clustering to obtain multiple clusters of sentences, and finally compress each cluster to generate the final summary [14].

3.PRELIMINARY CONCEPTS

In this section, the related backgrounds to the ELM and LexRank are described.

3.1 ELM

In 2006 Huang et al. have proposed a new learning algorithm Extreme Learning Machine (ELM). The proposed algorithm consists of Single hidden Layer Feedforward Neural Network (SLFN) When the hidden nodes selected randomly and the analytically determined the output weights [15]. Compared with other neural network algorithm, ELM has many advantages over the convergence speed and local optimization. In later studies the algorithm developed to be more than one hidden layer. ELM is widely used in batch learning, sequential learning, and incremental learning because of its fast and efficient learning speed [16]. ELM has been used in many applications such as text clustering[17], Bioinformatics[18] and face recognition[19].

Basically, ELM consist of three layers. Input layer, hidden layer, and output layer. Eq.1 illustrate the output of ELM.

$$y_i = \sum_{i=1}^h \beta_i g(w_i x_j + b_i) \quad \text{for } j=1 \text{ to } 1 \quad k \quad (1)$$

Where x_j is the input vector for j node, h is the number hidden node, g is the activation function, w_i is the weight vector between input layer and hidden layer, b_i is the bias, k is the number of input samples, finally β_i is the weight of the output layer. The main algorithm of ELM is shown below [15].

ELM Algorithm: given two vectors X, T that represents input with its target for the training phase, where $X=[x_1, x_2, \dots, x_k]$, $T=[t_1, t_2, \dots, t_k]$, h number of hidden nodes, and activation function $g(x)$

Step1:for $i=1$ to h

Randomly assign w_i and b_i

Step2: Calculate the hidden layer output matrix H.

Step3: Calculate the output weight β using

$$\beta = HtT$$

LexRank

LexRank is an unsupervised approach to text summarization, used for computing sentence importance depend on graph-based centrality scoring of sentences [20]. The concept is that, the sentences support other sentences to be more imperative to the per user. Along these lines, on the off chance that one sentence is fundamentally the same as numerous others, it will probably be a significant sentence. The significance of this sentence moreover comes from the significance of the sentences supporting it. In this manner, to get a high score and place in a summary, a sentence should be similar to many sentences. The cosine similarity was used to measure the similarity between sentences. The following algorithm shows the main steps of LexRank technique [21].

LexRank Algorithm: given a set of sentences S that represent a document collection and a threshold value.

Step1: calculate the cosine similarity matrix between the sentences of document collection.

Step2: delete the relation between two sentences if it less than threshold.

Step3: normalize the cosine matrix.

Step4: ensure that the matrix is irreducible, guarantee every node of the graph is accessible from any node.

Step5: calculate the score for every sentence in the graph.

4.METHODOLOGY FOR MDS

This section introduces formally the problem of MDS. Consider a corpus that consists of many clusters, each cluster contains a set of documents called D with the same topic. The set D can be defined as $D = \{d_1, d_2, d_3, \dots, d_n\}$ where n is the number of distinct document in D. Each d_i has a set of sentences S. The total number of sentences in D is M. The goal is to obtain a subset of the sentences which summarizes D. In the proposed method, the D is preprocessed and then a set of features is extracted. After that ELM used to classify sentences as important or unimportant. Then a graph is constructed for the important sentences. Finally TextRank algorithm is adopted to score the sentences, where best score sentences with less redundancy included in the summary.

4.1 Preprocessing and feature extraction

There are four steps in the preprocessing: Sentence separation, tokenization, stop words removal and stemming. After the preprocessing, the feature extraction process is implemented. The feature extraction process plays an essential role since it influences the effectiveness of the system summarization. In this paper, a set of seven features $F = \{f_1, f_2, \dots, f_7\}$ similar to [6] is extracted as illustrated in the following.

1- Sentence length: This feature is used to compute the sentence length and ignoring the very short and very long sentences. The average sentence length can be computed by Eq.3.

$$AL(S) = \frac{\max(sentence_length) + \min(sentence_length)}{2} \tag{2}$$

Where AL(S_i) is the average sentence length in the summary.

$$f1 = \frac{|AL(S) - SL|}{\max(sentence_length)} \tag{3}$$

2- Sentence position: This feature is very significant because the first and last few sentences contain the most important sentences in the document. This feature can be calculated as in Eq.4.

$$f2 = \frac{\left| \frac{n-i}{2} \right|}{\frac{n}{2}} \tag{4}$$

Where i and n represent the position and the number of sentences in the document respectively.

3-Numerical data: sentences with numerical data has significant information to be involved in the summary. Can be computed as in Eq.5.

$$f3 = \frac{\text{count_num}(S)}{L(S)} \tag{5}$$

Where count_num is the number of numerical data in the S sentence and L is the length of sentence.

4-Named entities: sentences with this feature have more chance to be included in the summary. . Can be computed as in Eq.6.

$$f4 = \frac{\text{number of named entities in } S_i}{\text{Maximum number of name entities in a sentence}} \tag{6}$$

5-Title words: This feature shows how relevant the sentences of the document to the title. Can be calculated as in Eq.7.

$$f5 = \frac{\text{number of TW}(S)}{L(\text{title})} \tag{7}$$

Where Tw is the number of the title words in the sentence.

6-Sentence centrality: This feature computes the overlap between the sentence words and other sentences words in the document. This feature calculated as in Eq.8.

$$f6 = \frac{W_{S_i} \cap W_{S_j}}{W_{S_i} \cup W_{S_j}} \tag{8}$$

For i,j from 1 to n where $i < j$.

7-Frequent words: It is based on the premise that the relevant sentences contain more frequent words. This feature is calculated by computing the repetition of all words in the document divided by the total number of document words as in Eq.9.

$$f7 = \frac{\text{count_fw}(S)}{\text{no_of_w}(d)} \tag{9}$$

Where fw denotes the frequent words.

4.2 ELM as MDS classifier

The seven extracted features as in section 4.1 are introduced to the ELM for the classification purpose. A binary classifier is used where there are only two classes, and the classification is made based on the sentence present or not in the given summary. The classifier attempt to classify the sentences into one of two classes Important or unimportant sentences. There are two phases for the classification purpose, training and testing phase. The cosine similarity was used as in Eq.10 in the training phase to assign labels to every sentence in the document collection.

Cosine similarity is the most common measure that is used to compute the similarity between two vectors. In which there are more terms that are in common between two vectors. Its computations are fast and unaffected by the number of term appearances in the vector. The cosine similarity value range between [0,1], Where 0 means there is no similarity between two vectors, and 1 means that the two vectors are identical to each other. The comparison is done between document sentences and a reference summary [22]. The sentence is labeled as an important sentence when the result of the cosine similarity exceeded specified threshold.

$$\text{Similarity}(S, S_summary) = \frac{\sum_{i=1}^k S_i S_summary_i}{\sqrt{\sum_{i=1}^k (S_i)^2} * \sqrt{\sum_{i=1}^k (S_summary_i)^2}} \tag{10}$$

Where k is the total number of terms, S is a sentence from document collection and S_summary is a sentence from the reference summaries. Every sentence is compared with all reference summary sentences and highest score is assigned to it.

4.3 LexRank for sentences scoring

The LexRank algorithm applied to rank and identify the most important sentences to be extracted as a summary. The rank of sentences shows its importance which depends on the number of incoming and outgoing links and the strength of similarity links connecting each sentence to the rest of the sentences. The graph of the important sentences that pass from the ELM classifier is constructed, where vertices represent the sentences and edges are defined in terms of relations between pairs of sentences. The weight on the Edges are calculated using cosine similarity. The salience of a sentence S_i based on the LexRank algorithm is formulated as in Eq.11.

$$LexRank(S_i) = \frac{d}{P} + (1 - d) \sum_{S_j \in L(S_i)} \frac{w_{i,j} LexRank(S_j)}{C(S_j)} \tag{11}$$

Where

- d is a damping factor.
- $L(S_j)$ is a set of input link that point to S_i .
- $C(S_j)$ is a set of output link from vertex j
- $w_{i,j}$ is the cosine similarity between (S_i) and (S_j)
- P is the number of nodes in the graph.

The algorithm starts by assigning equal weight to each sentence in the graph [21]. For each iteration the LexRank value of every sentence is updated based on Eq.11. Sentences are arranged in descending order according to the value of the resulting vector.

4.4 Summary generation

The final stage of the extractive summarization is to select the most appropriate sentences from documents and produce the final summary with a pre-specified length. The sentences are sorted in descending order of their similarity scores. The top ranked sentences are iteratively inserted to compose the final summary. In order to ensure good summary, three aspects must be considered: coherence, reduce redundancy of information and fixed length summary. The coherence can confirm by selecting sentences according to the order in the original documents. In addition, to reduce redundancy, The cosine similarity was used to compare the new selected sentence with the each of the sentence already in the summary, if the similarity less than a specific threshold, the new sentence will be added to the summary, otherwise neglected. This process was repeated until reaching a fixed summary length.

The choosing of the threshold value is very critical and affect the system performance. Very high threshold value leads to increase the redundant information in the summary. Very low threshold value leads to loss some useful information. The value of the threshold was empirically set to be 0.5.

5. EXPERIMENTAL RESULTS

In this section, The performance evaluation of the proposed method is discussed.

5.1 Dataset description

The Document Understanding Conference (DUC) dataset has been used to evaluated the performance of the proposed system. In This paper, DUC-2002 dataset was used for testing the proposed summarization system. DUC-2002 is a famous international evaluation conference contains documents for both extractive and abstractive summarization. The dataset divided into many clusters, each cluster contains documents related to the same topic. Table-1 show brief statistics of the dataset [23].

Table 1.Description of DUC-2002.

Description	DUC-2002
Number of clusters	59
Number of documents in each cluster	From 5 to14
Total number of documents	567
Number of sentences per document	At least 10 sentences
Summary length	200 and 400 words

5.2 Evaluation Standards

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) toolkit is the most popular assessment toolkit used to evaluate the quality of the summary. ROUGE measures the similarity between the automatic summaries generated by the system (also called candidate summary) and the human summaries (called reference summaries or model summaries) that generated manually. ROUGE-N used to compute N-gram overlap between both summaries. ROUGE-1 and ROUGE-2 metrics have been widely used For automatic summary evaluation. They based on computing unigrams and bigrams overlapping, respectively [24]. The outputs of ROUGE are calculated with respect to precision, recall and F-score, which will be calculated as in Eq.12, Eq. 13 and Eq.14 respectively [25].

$$Precision = \frac{System\ Summary \cap Reference\ summary}{System\ Summary} \tag{12}$$

$$Recall = \frac{System\ Summary \cap Reference\ summary}{Referencd\ Summary} \tag{13}$$

$$F - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{14}$$

5.3 ELM performance

The classification phase is the first step of the proposed method. The ELM classifier used to determine the sentences are important or unimportant to pass to the next step. There are two phases in the classification step, training phase and testing phase. The 3 fold cross validation was used, For the purpose of implementing these two phases. Where the dataset has been divided into roughly three equal size sets. Two of these sets were used for the training phase, while the third set was used for the testing phase. Table 2 shows the classification accuracy of the ELM.

Table2. The Classification Accuracy of ELM

Accuracy	Misclassification of Important sentences	Misclassification of Unimportant sentences
87%	5%	8%

Table 2 shows the accuracy of the ELM for MDS classifier. As shown in the Table 2 the loss of information for important and unimportant sentences reaches 5% and 8% respectively. The loss of unimportant sentences does not affect the final system performance, since these sentences will pass to the next step of MDS model and reevaluated again. The loss of important sentences occurred since most of theses sentences features are very closed to the unimportant sentences features. They are located in an area where there is a great overlap between important and unimportant sentences. On the other hand, the scores of these sentences are low, thus their loss does not have a significant impact on the final results.

5.4 Performance comparison against other methods

The results of the proposed method were compared with baseline methods as explained in Tables 3 and 4 together with Figures 1 and 2. The comparison based on 200 and 400-word summaries. The selected methods for comparison based on mathematical, statistical or a graph theory that similar to the proposed method.

Table 3. Comparison with baseline methods for 200-word summaries.

Method	ROUGE-1			ROUGE-2		
	Precision	Recall	F-score	Precision	Recall	F-score
LSA [26]	0.37012	0.37723	0.37342	0.08910	0.08950	0.08921
LexRank [20]	0.45984	0.48124	0.46997	0.17566	0.18507	0.18010
TextRank[27]]	0.43685	0.48417	0.45868	0.15195	0.16921	0.16000
Sumbasic[28]	0.45114	0.46128	0.45597	0.15584	0.15947	0.15757
KLsum[29]	0.36803	0.37464	0.37104	0.11938	0.12113	0.12018
Proposed	0.45844	0.49221	0.47472	0.16923	0.19810	0.18253

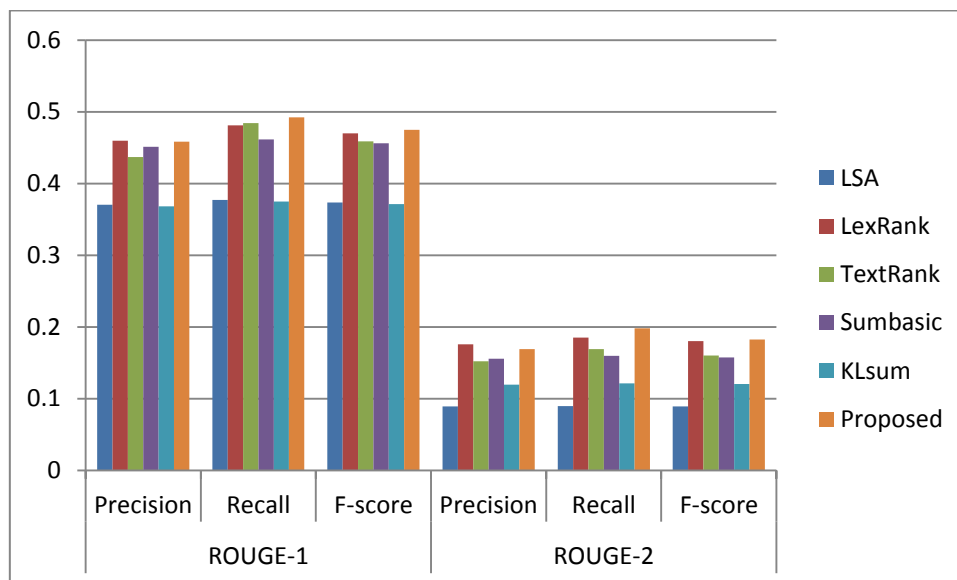


Figure 1. Graphical comparison with baseline methods for 200-word summaries.

As shown in Table 3 and Figure 1 The proposed method outperforms other methods, especially in recall and F-score, since the LexRank gives the best results and the proposed method based on improving the performance of LexRank. The improvement is the result of reducing the number of sentences during the classification stage prior to LexRank. As mentioned in section 3.2 the LexRank score the sentences according to their similarity with other sentences in the graph. The existing of important and unimportant sentences causes many sentences with less importance to get higher score and included in the summary. Eliminating unimportant sentences increase the score of more important sentences to be included in the summary.

It's clear that the LexRank outperformed the proposed method in term of precision, whereas the recall and F-score of the proposed method are higher. The F-score is the judgment between the recall and the precision, since it considers both.

Table 4. Comparison with baseline methods for 400-word summaries.

Method	ROUGE-1			ROUGE-2		
	Precision	Recall	F-score	Precision	Recall	F-score
LSA [26]	0.47293	0.46361	0.46804	0.16297	0.16001	0.16141
LexRank [20]	0.53431	0.55018	0.54188	0.23088	0.23706	0.23382

TextRank[27]	0.52849	0.57812	0.55179	0.23379	0.25654	0.24447
Sumbasic[28]	0.50610	0.52373	0.51462	0.18193	0.18916	0.18542
KLsum [29]	0.43254	0.43746	0.43490	0.16017	0.16256	0.16132
Proposed	0.51642	0.58012	0.54641	0.24310	0.26652	0.254271

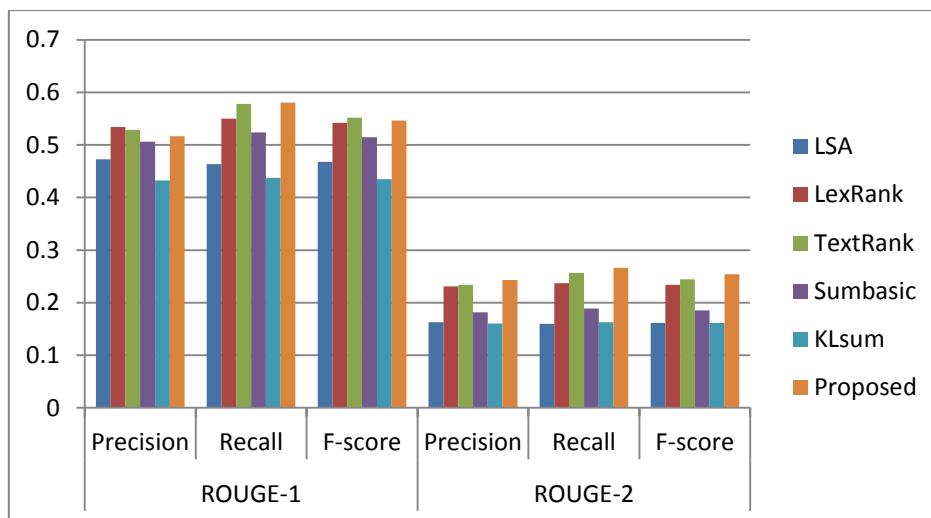


Figure 2. Graphical comparison with baseline methods for 400-word summaries.

As shown in table 4 and Figure 2 The proposed method gives the best results for ROUGE-2, while the TextRank gives the best results for ROUGE-1. The TextRank algorithm based on assigning scores to the extracted keywords instead of sentences. Thus, when the length of the summary is a longer, TextRank algorithm produces a suitable summary especially for ROUGE-1 which based on computing a unigram similarity. ROUGE-2 is more important than ROUGE-1, since it's closer to human summary.

6. CONCLUSION

MDS is an interesting research topic among the NLP community that helps produce concise information. This paper, used a new method for MDS system that exploits the power of ELM and LexRank to produce a good summary. The proposed method based on extracting seven features for each sentence, The ELM used to classify the sentences as important or unimportant sentence. Finally, the LexRank was used to assign a sentence score. The best ranked sentences are included in the summary. Good results were obtained compared with baseline methods. The performance of LexRank decreases with increasing the number of sentences, therefore reducing the number of sentences during the ELM phase very necessary to produce a good summary quality.

REFERENCES

- [1] Ermakova, L., Cossu, J. V., & Mothe, J. (2019), A survey on evaluation of summarization methods. Information Processing & Management, Vol 56,pp. 1794-1814.
- [2] Yousefi-Azar,M. & Hamey,L. (2017), Text summarization using unsupervised deep learning, Expert Systems with Applications, vol. 68, pp. 93-105 .
- [3] Alguliev, R,M. , Aliguliyev,M. & Isazade,N.,R., (2013), Multiple document summarization based on the evolutionary optimization algorithm,Expert Systems with Applications, vol. 40, no. 5, pp. 1675-1689.
- [4]J Sanchez-Gomez,M.,(2018) , Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach, Knowledge-Based Systems, vol. 159, pp. 1-8.
- [5] Zuhair H., A., Ahmed K., H., Haithem, K., A., Elham F.1,(2021), Extractive multi document summarization using harmony search algorithm , TELKOMNIKA ,Vol. 19, No. 1,, pp. 89~95.

- [6] Oliveira, H., Ferreira, R., Lima, R., Lins, R. D., Freitas, F., Riss, M., & Simske, S. J. (2016). Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization. *Expert Systems with Applications*, vol. 65, 68–86.
- [7] Alguliyev, R., (2017)., A model for text summarization,” *International Journal of Intelligent Information Technologies*, vol. 13, no. 1, pp. 67-85.
- [8] Ferreira, R. et al., (2013) Assessing sentence scoring techniques for extractive text summarization, *Expert systems with applications*, vol.40, pp. 5755-5764.
- [9] Uçkan, T., & Karçıl, A. (2020). Extractive multi-document text summarization based on graph independent sets. *Egyptian Informatics Journal*. doi:10.1016/j.eij.2019.12.002
- [10] Premjith, P.S.; John, A.; Wilscy, M., (2015), Metaheuristic Optimization Using Sentence Level Semantics for Extractive Document Summarization. In *Mining Intelligence and Knowledge Exploration*; Prasath, R., Vuppala, A.K., Kathirvalavakumar, T., Eds.; Springer International Publishing: Cham, Switzerland, pp. 347–358.
- [11] MirShojaee, H.; Masoumi, B.; Zeinali, E.A., (2017), Biogeography-Based Optimization Algorithm for Automatic Extractive Text Summarization. *Int. J. Ind. Eng. Prod. Res.* , 28.
- [12] Yasunaga, M. , Zhang, R., Meelu, K. , Pareek, A., , Srinivasan, K., & Radev, D., (2017). Graph-based Neural Multi-Document Summarization.
- [13] Cuéllar C., Mendoza M., Cobos C. (2018) Automatic Generation of Multi-document Summaries Based on the Global-Best Harmony Search Metaheuristic and the LexRank Graph-Based Algorithm. In: Castro F., Miranda-Jiménez S., González-Mendoza M. (eds) *Advances in Computational Intelligence. MICAI . Lecture Notes in Computer Science*, vol 10633. Springer, Cham. https://doi.org/10.1007/978-3-030-02840-4_7.
- [14] Jinming Z., Ming L., Longxiang G., Yuan J., Lan D., He Z., He Z. & Gholamreza H., (2020), SUMMPIP: UNSUPERVISED MULTI-DOCUMENT SUMMARIZATION WITH SENTENCE GRAPH COMPRESSION, *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* pp. 1949–1952
- [15] Huang, G., B., Zhu, Q., Y., & Siew, C., K., (2006) ,Extreme learning machine: theory and applications, *Neurocomputing*, vol. 70, no. 1, pp. 489–501.,
- [16] Yang, L., Li, Y., & Li, Z. (2017). Improved-ELM method for detecting false data attack in smart grid. *International Journal of Electrical Power and Energy Systems*, vol. 91, pp. 183–191.
- [17] Zhao, X., G., Wang, X. , Gong, P and Y. Zhao, (2011), XML document classification based on ELM, *Neurocomputing*, Vol.74, no. 16, pp.2444-2451.
- [18] Wang, G., Zhao, Y. & D. Wang, (2008) A protein secondary structure prediction framework based on the Extreme Learning Machine, *Neurocomputing*, vol.72, no.3, pp.262-268.
- [19] Choi, K., . Toh, K., & Byun, H., (2012), Incremental face recognition for large-scale social network services, *Pattern Recognition*, Vol. 45, no.8, pp.2868-2883.
- [20] Erkan ,G., and Radev, D.R. ,(2004), LexRank: Graph-based Lexical Centrality as Saliency in Text Summarization, *Journal of Artificial Intelligence Research* vol.22, pp. 457-479.

- [21] Mao X, Yang H, Huang S, Liu Y, Li R.,(2019), Extractive summarization using supervised and unsupervised learning. *Expert Syst* vol.133,pp.173–81.
- [22] Lin,Y. , Jiang.J & Lee,S.(2014)." A Similarity Measure for Text Classification and Clustering". *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, Vol. 26, No. 7,PP.1575-1590
- [23] NIST. Document understanding conferences. Available: [https:// www-nlpir.nist.gov/projects/duc](https://www-nlpir.nist.gov/projects/duc).
- [24] Lin, Chin-Yew (2004). ROUGE: A Package for Automatic Evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)* .
- [25] Gholamrezazadeh, S., Salehi, M. A., & Gholamzadeh, B. (2009). A comprehensive survey on text summarization systems. *Computer Science and its Applications*, 1-6.
- [26] Landauer TK, Foltz PW, Laham D.,(1998), An introduction to latent semantic analysis.*Discourse Process* , Vol.25,(2-3),pp.259-84.
- [27] Mihalcea R.,(2005), Language independent extractive summarization. In: *Proc. ACL interact*. Poster demonstrates. *ACL no.05*, p. 49-52.
- [28] Vanderwende L, Suzuki H, Brockett C, Nenkova A. Beyond,(2007) SumBasic: task focused summarization with sentence simplification and lexical expansion. *Inf Process Manage* , Vol.43, no.6,pp.1606-18.
- [29] Haghighi A, Vanderwende L.(2009, Exploring content models for multi-document summarization ,vol. 362.