International Journal of Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: 10.31695/IJASRE.2022.8.8.7

Volume 8, Issue 8
August - 2022

Multi-Document Text Summarization Using Deep Belief Network

Azal Minshed Abid

Department of Computer Science

College of Education, Mustansiriyah University

Iraq

ABSTRACT

Recently, there is a lot of information available on the Internet, which makes it difficult for users to find what they're looking for. Extractive text summarization methods are designed to reduce the amount of text in a document collection by focusing on the most important information and reducing the redundant information. Summarizing documents should not affect the main ideas and the meaning of the original text. This paper proposes a new automatic, generic, and extractive multi-document summarizing model aiming at producing a sufficiently informative summary. The idea of the proposed model is based on extracting nine different features from each sentence in the document collection. The extracted features are introduced as input to the Deep Belief Network (DBN) for the classification purpose as either important or unimportant sentences. Only, the important sentences pass to the next phase to construct a graph. The PageRank algorithm is used to assign scores to the graph sentences. The sentences with high score selected to create a summary document. The performance of the proposed model evaluated using DUC-2004 (Task2) dataset using ROUGE more. The experimental results demonstrate that our proposed model is more effective than baseline method and some state-of-the-art methods, Where ROUGE-1 reached to 0.4032 and ROUGE-2 to 0.1021.

Key Words*: *DBN, PageRank, DUC-2004, ROUGE, Summarization.

---

1. INTRODUCTION

The availability of the massive amount of data in the Internet nowadays has reached such a vast volume. The massive amount of information has brought abundant information to users but also caused huge reading barriers, making it more difficult to efficiently access usable information [1]. One way to solve such a problem of information overload is by generating summaries. Automatic Text Summarization (ATS) is a process of automatically creating a shorter version of a document or a set of documents by reducing the document(s) in length by selecting important information and discarding unimportant and redundant information, keeping the key elements and the basic meaning of information [2]. The ATS systems allow users to quickly grasp the essential elements of a document without having to read the entire document. Users will profit from the automatically generated summaries, which will save them a great deal of time and effort [3].

Regarding the number of input documents to be summarized, the summary can be a single-document summarization (SDS), which produces the summary from a single document, or a multi-document summarization (MDS), which extracts the summary from a set of documents [4]. The process of determining relevant sentences from a single summary is much simpler. Assuming that the sequence of the selected sentences remains consistent with that of the original document, the summary retains its coherence. One issue with MDS is deciding which sentences to extract from the documents and how to show them (in what sequence); also, coherence is a difficult task to solve, and information duplication across documents is critical [5]. Because the summarized texts cover the same topics, there is a lot of repetition. There must be no redundant information in the resulting summary. SDS, on the other hand, did not have any redundant data [6].

Text summarization approaches can, also, be either extractive or abstractive according to the function to be performed. An extractive approach aimed to find and choose the most relevant sentences in the source documents exactly as they appear without any change. Abstractive summarizing is a technique for creating a summary of a text based on its key concepts rather than simply transcribing the most important sentences from the text. Abstractive summarization is more efficient than extractive summarization as it generates human-like summaries, but it's more difficult to implement since it required deep NLP understanding techniques [7]. Also, text summarization techniques can be divided into two categories based on the task of summarization: A generic summary presents the entire content of a document without any prior knowledge, whereas a query-related summary presents information that is relevant to a specific query or topic. Figure 1 shows the taxonomies of text summarization [8].

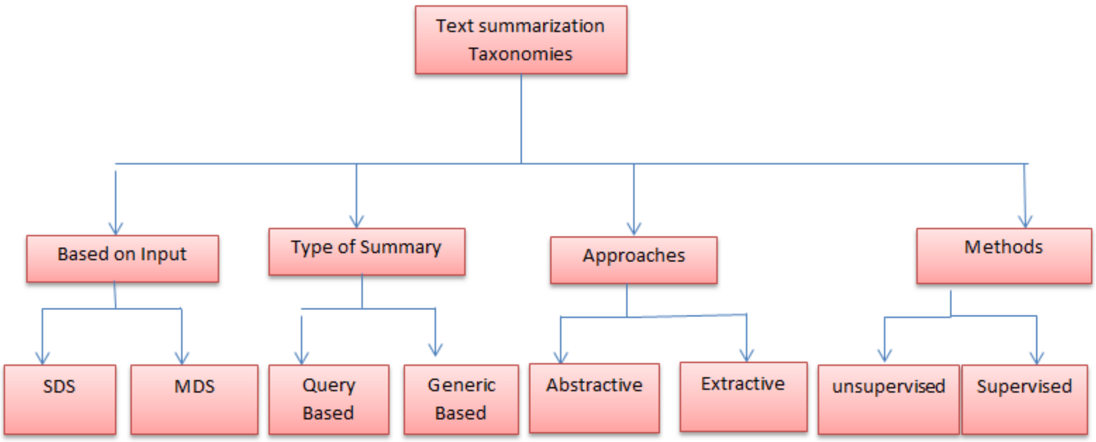

Figure 1: Taxonomies of Text Summarization

In this paper, a new model for extractive and generic MDS tasks is presented. The main idea of this model is based on extracting nine features from each sentence. A DBN is applied to classify the sentences as important or unimportant sentences. Lastly, A graph is built for the important sentences to assign a score. Only high score sentences are selected to generate the summary.

The main contribution of this paper is based on combining supervised with an unsupervised approach to build a model for MDS.

2. RELATED WORKS

Many methods have been used to automatically generate summaries for more than 50 years. In the literature, various classification schemes for these approaches have been presented. This section deals with two important approaches for document summarization, the graph-based approach, and the deep learning approach.

In 2015 Parveen suggest a graph-based technique for extractive document summarization. The input document is presented as a bipartite graph consisting of sentence and entity nodes with preserving the sentence importance, non-redundancy, and coherence. The sentence ranking is done by applying a graph-based ranking algorithm to preserve the importance, while the optimization step is used to ensure non-redundancy and coherence [9]. In 2017 Yasunaga proposed a method for MDS that incorporates sentence relation graphs. The Recurrent Neural Network used to produce sentence embedding. The Graph Convolutional Network uses these sentences with a relational graph to generate high-level hidden sentence features for salience estimation, which in turn is used by the greedy heuristic to select the most important sentences. Duc-2004 was used to evaluate the system [10]. In 2019 ALZUHAIR propose a method for extractive MDS. The method is based on combining two graph approaches. The system used four weighting schemes and two ranking methods for each graph. The results of these two graphs are computed using arithmetic mean. The proposed approach evaluated using DUC-2003 and DUC-2004 datasets [11].

In 2018 Singh proposed HNet which is used to capture document independent information and takes advantage of the numerous semantic and compositional factors implicit in a sentence. Important sentences are closer together in the vector as the network learns sentence representation. Ignoring non-important sentences. The document dependent features for sentence ranking are then concatenated with this semantic and compositional feature vector [12]. In 2019 cho proposed a hybrid model based on A Convolutional Neural Network(CNN) with Long Short Term Memory(LSTM) for extractive MDS. CNN uses point processes to measure sentence similarities semantically, while the LSTM is used to reconstruct pairwise sentences and add reconstruction loss to the final objective function [13]. In 2020 Mao proposed a model based on Relevance-guided Reinforcement Learning for MDS. The model is based on two important issues in MDS, large search space and solving high information redundancy. The reinforce learning adaptive to choose the best sentences from a high space dataset, while the Maximal Marginal Relevance used to solve the redundancy problem [14].

DOI: 10.31695/IJASRE.2022.8.8.7

3.THEORETICAL BACKGROUND

In this section, a brief background to the Restricted Boltzmann Machine (RBM) and Deep Belief Networks are described.

3.1 Restricted Boltzmann Machine (RBM)

In 2007 the RBM was introduced by Geoffrey Hinton. RBM consists of two layers called the visible layer or input layer and hidden layer, where v stands for input layer and h for the hidden layer. There is a set of neurons in each layer. The output layer does not exist [15]. RBM is similar to the Boltzmann machine in everything, but no connection exists between neurons of the same layer. Although the neurons of the hidden layer and visible layer can be connected to each other as shown in figure 2. The RBM is a generative and random neural network that can learn internal representations, describe complex combinatorial problems, and solve them [16].

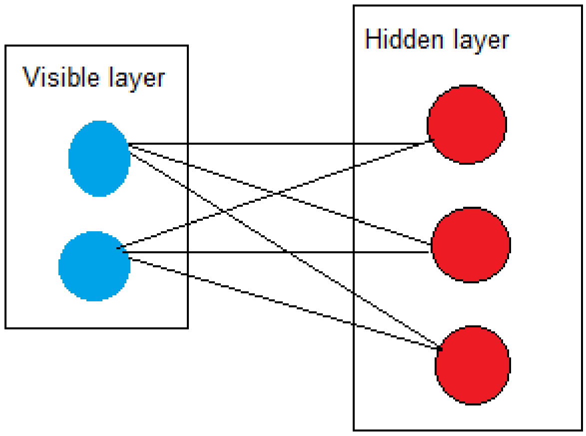

Figure 2: Structure of RBM

RBM network consists of two phases: forward phase and backward phase. The forward phase is concerned with activating the hidden layer by taking the input with the concept of weight in addition to the biased. This phase allows identifying the positive association when the association between the visible unit and the hidden unit is positive, and the negative association when the association between the visible unit and the hidden unit is negative. The backward phase is concerned with the reconstruction of the input layer using the hidden state that has been activated since there is no output layer [17].

3.2 Deep Belief Network

DBN is developed from the deep learning approach for feature learning as well as classification. DBN is a powerful generative model that uses a deep architecture of many stacks of RBM as shown in figure 3. DBNs can be applied in both supervised and unsupervised environments because they are generative models [18]. Meaning that DBNs are capable of performing feature learning, extraction, and classification, which are employed in several applications. Since each layer of DBN is constructed as RBM, training each layer of DBN is the same as training an RBM [19].

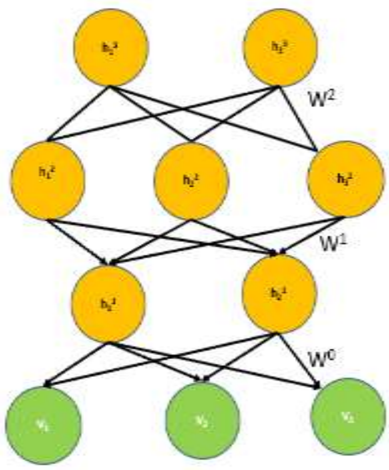

Figure 3: Deep Belief Network

DOI: 10.31695/IJASRE.2022.8.8.7

DBN is used for dimensionality reduction in an unsupervised mode and for classification problems in a supervised mode. There are two important steps to training the DBN: layer-by-layer training and fine-tuning. Layer-by-layer training is the term for unsupervised training. The extracted parameters from this step are fine-tuned using error back-propagation methods in the second step.

The greatest advantage of DBN is the capability of learning features, which is achieved by layer-by-layer learning strategies where the higher-level features are learned from the previous layer [20].

4. PROBLEM DEFINITION AND FORMULATION

The problem of MDS can be formulated and defined as follows: Given a set of document collection D={d1,d2,…,dn} where n is the number of documents in D. A single document di is represented as a set of sentences di={S1,S2,..,Sk} where k is the total number of sentences in di. In this study, the document collection D is represented as a set of all sentences in the collection D={s1',s2',…sp'} where p is the total number of sentences in the document collection. The goal of MDS is to transform a document collection D into a single output document Dout. The output document was represented as Dout={S1,S2,..Sm}. This transformation must satisfy the following three aspects of ATS.

- Relevance: The selected sentences should comprise informative textual units that are important to the users.
- Redundancy: The generated summary must not contain redundant information.
- Length: summary should be bound to a specific length.

To achieve these goals, each sentence Si in D is evaluated using a set of the most essential statistical and semantic features in order to indicate its significance. Finally, the DBN and graph theory are applied to create Dout by integrating the top-scoring sentences.

5. PROPOSED SUMMARIZATION METHOD

In this paper, a DBN and graph are presented to implement MDS. The proposed model consists of four main stages. The first stage is concerned with applying preprocessing to every document in the dataset. A set of features is extracted in the second stage. The third stage is concerned with applying DBN to classify the sentences as important or unimportant. While the last stage is concerned with building a graph to select the most important sentences.

5.1 Preprocessing and Feature Extraction

The preprocessing stage is the starting stage for most summary approaches. Its concerned with preparing the input documents for processing in other stages. This stage includes sentence segmentation, tokenization, removing stop words, and stemming. After the preprocessing, a feature extraction process is performed. The feature extraction process plays a critical role because it affects the efficiency of the proposed MDS model. The MDS that involves choosing sentences of high relevance or importance depends on employing a set of features to produce a good summary that reflects the main idea of the summarized documents. Therefore, choosing these features should be careful due to their impact on the efficiency of the resulting summary.

Many approaches for ATS are based on feature extraction. These features can be categorized into statistical features, similarity features, semantic features, and graph-based features [21]. The extracted feature affects the quality of the generated summary. Due to the lack of consideration for meaning and potential for repetition in the generated summary, using statistical features only may not produce meaningful summaries; also using semantic features only will miss out on some statistical facts. Therefore, this paper is based on using a combination of statistical and semantic features. The extracted features are similar to [22,23]. These features are briefly presented in the following.

1.Sentence Length: the selected sentences must not be very short or very long. Where short sentences do not reflect the main topic of the summarized documents. Similarly, a very long sentence may include important information in part and unimportant information in part. Therefore, sentences that are longer or shorter than a specified threshold are ignored, then the sentence length is computed as in Eq.1

$$sen\_len(S_i) = \frac{Number\ of\ word\ in\ S_i}{Max\ number\ of\ word\ in\ sentence} \qquad (1)$$

2. Sentence Position: The location of sentences in the document is very important, where most instructive information is put in the front of the document. The importance of the sentences decreases gradually as the location of sentences moves to the end of the document. This feature is calculated as in Eq.2.

DOI: 10.31695/IJASRE.2022.8.8.7

$$Sen\_pos(s_i) = \begin{cases} 1 \ \ if \ i \leq 3 \\ 1 - \frac{i-3}{i} \ if \ i > 3 \end{cases} \quad (2)$$

Where i is the sentence position in the document.

3. Numerical data: the existence of numerical data in the sentence increases its importance to be included in the summary. This feature can be calculated as in Eq.3.

$$Numdata(s_i) = \frac{num\_count(s_i)}{length(s_i)} \quad (3)$$

4. Proper noun: this feature can be used to refer to persons, locations, and organizations, among other things. More proper nouns in a sentence might be seen as an important sentence, and more likely to appear in a document summary. This feature is calculated as in Eq.4.

$$ProNoun(si) = \frac{number\ of\ ProNoun\ in\ S_i}{number\ of\ Words\ in\ S_i} \quad (4)$$

5.Sentence centrality: based on computing the similarity between the sentence (Si) and all other sentences in the document. This approach is predicated on the idea that a central sentence best describes the main information of a document. Eq.5.

$$SentCen(s_i) = \frac{Ws_i \cap Ws_j}{Ws_i \cup Ws_j} \quad (5)$$

6. Key-phrase Frequency: Key-phrase consists of a set of important keywords that give a brief overview of the main document topic. They could consist of a single word or a combination of many words. The existence of a key phrase in the sentence increases its importance. The frequency of key-phrase in the sentence can be calculated as in Eq.6.

$$kpf(s_i) = \frac{No.\ kps\ in\ s_i}{Kpf\_total} \quad (6)$$

Where

KPS is the number of key-phrase in the sentence i.

KPF_total is the total number of key-phrases in the document.

7. Similarity with the title: This feature based on computing the overlap between a sentence and a document title Eq.7.

$$Title\_sim(s_i) = \frac{S_i \cap Title}{length(Title)} \quad (7)$$

8. Lexical similarity: is based on the idea that powerful word chains, such as those created by synonymy and other semantic relationships, make up significant sentences. A higher score is assigned to the sentence that includes strong chains. Eq.8.

$$Lexsim(s_i) = \frac{No.of\ strong\ chain\ in\ s_i}{Max\ No.of\ strong\ chain\ in\ sentence} \quad (8)$$

9.Term Frequency - Inverse Document Frequency (TF-IDF): is a score that assigns to each term in the document set. It measures how important a term is within a document relative to a collection of documents. Eq.9.

$$TF = \frac{Frequency\ of\ term\ in\ document}{total\ NO.of\ terms\ in\ document}$$

$$IDF = \log\left(\frac{NO.of\ document\ in\ the\ dataset}{NO.\ of\ documents\ containing\ the\ given\ word}\right)$$

$$TF - IDF = TF * IDF \quad (9)$$

DOI: 10.31695/IJASRE.2022.8.8.7

5.2 DBN as classifier

The classification model is built using a DBN classifier which takes the extracted features as input and is used to construct the classification model. It then develops the model by learning from the training samples. Once the model has been established, only the labels that are already known can be used to classify newly received sentences. DBN is used to classify each sentence as either summary or non-summary sentence based on sentence features. DBN needs the observation labels to be available through the training of the top layer, so a training session includes first training the bottom layer, propagating the sentence features through the learned RBM, and then using that newly transformed sentences as the training data for the next RBM.

Two important phases are required for any classification problem: training and testing. The training phase contains input document collection and the ideal summaries. The input is represented as a feature vector for every sentence in the documents. A summary or non summary label is assigned to every sentence based on computing The Jaccard similarity between the input sentence and the ideal summaries sentences [24]. A sentence is labeled as a summary sentence when the similarity with ideal summaries sentences exceeds a specified threshold. For the testing phase, each sentence will be used as input to the trained CNN model to be classified as a summary or non-summary sentence, where the summary sentences pass to the next step of the proposed summarization model.

$$Jacc_{sim} = \frac{A \cap B}{A \cup B} \tag{10}$$

Where

A is a sentence from the document collection.

B is a sentence from the reference summaries.

5.3 Graph construction

The output sentences from DBN that are classified as summary sentences are passed to the next step to build a graph. The graph consists of nodes and edges, where sentences are represented as nodes and the edges represent the relationship between nodes. Obtaining the more important sentences is the primary goal of the graphical approach. Basically, the undirected and weighted graph is used to assess each node's importance within a graph. The sentences are represented as nodes of the graph. An edge between two sentences is built if there is some similarity between them. Similarity simply means word overlap. Thus, if two sentences share at least one word, there will be an edge between them. Jaccard's approach is used to compute the similarity between nodes of the graph.

After graph construction, the PageRank algorithm is used to compute nodes scores. The PageRank used by the Google Internet search engine that assigns a numerical weighting to each Web page [25]. The same idea of PageRank can be used to score nodes with the following differences: Pages are replaced with sentences and the edge weight between nodes are calculated using the Jaccard algorithm. The PageRank algorithm is formulated as in Eq.10.

$$PR(N_i) = (1-d) + d * \sum_{N_j\, In(N_I)} \frac{W_{j,i}}{\sum N_k \in out(N_j) W_{j,k}} PR(N_j) \tag{11}$$

Where

- d is a damping factor.
- In(Ni) is a set of input link that point to the node (Ni).
- Out(Nj) is a set of output link from node N(j).
- Wj,i is the weight between node i and j calculated using Jaccard.

5.4 Sentences Selection

After scoring all sentences. The most pertinent sentences from the graph are chosen in the last stage of extractive summarizing, and the final summary with a predetermined length is produced. The way typically used to choose sentences from documents is to score the sentences in descending order with keeping the order of sentences in the documents to ensure the readability of the summary. Also, to preserve the coverage of the summary the selected sentences must not contain repeated information. To bypass the problem of redundancy, the Jaccard algorithm was used to calculate sentences similarity.

The ranked sentences are arranged in descending order from highest to lowest. The sentences are chosen to form the final summary as in the following algorithm.

DOI: 10.31695/IJASRE.2022.8.8.7

Algorithm: summary Generation

Step1: Let f-summary=[]

Step2:for all candidate sentences Si to be in f-summary

Calculate the Jaccard similarity between Si and f-summary sentences

If(Jaccard similarity<threshold) insert Si into f-summary

Else ignore S

6. EXPERIMENTS AND RESULTS

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

6.1 Dataset

Document Understanding Conference (DUC) is an evaluation series, prepared by the National Institute for Standards and Technology (NIST) and supported by the Defense Advanced Research Projects Agency (DARPA), in the area of automatic text summarization. DUC-2004 dataset was used for evaluating the proposed summarization system. DUC-2004 can be used for both SDS and MDS. DUC 2004 consisted of five tasks. Task2 was used in this paper [26]. The dataset is divided into many clusters, each cluster contains documents related to the same topic. Table 1 show brief statistics of the dataset.

Table 1. Statistics of DUC-2004.

Description	DUC-2004
Type of summaries	Generic
Number of Documents	500
Number of clusters	50
Total number of documents per cluster	10
Number of references Summaries per cluster	4
Summary length	665 characters max length

6.2 Evaluation metrics

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) used to evaluate the proposed method [37]. ROUGE used to compute n-gram overlap. The overlap computed between the automatic summaries generated by the system (also called candidate summary) and the human summaries (called reference summaries or ideal summaries) that generated by many experts.

ROUGE has proven to be a very useful tool for assessing document summarization. An N-gram is a contiguous sequence of N words, and N-gram co-occurrence statistics are used to gauge how well a machine summary coincides with human summaries. Several ROUGE measures, including ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-S, are defined in accordance with various N and distinct techniques. The ROUGE-Nmeasure compares the number of matches between the N-grams in two summaries.

$$\mathrm{ROUGE-N} = \frac{\sum_{S}(s \in \text{reference summaries})\sum_{n-gram \in S} \text{count}_{match}\,(N-gram)}{\sum_{S}(s \in \text{reference summaries})\sum_{n-gram \in S} \text{count}\,(N-gram)} \quad (12)$$

Where N represents the length of the N-gram, count (N-gram) is the number of N-grams in the ideal summaries. Countmatch(N-gram) is the largest number of overlap between a system summary and a set of ideal summaries [27].

6.3 RESULTS AND DISCUSSION

The proposed model is compared with both traditional methods and state-of-the-art models. The DUC04 data set is used for the general summarization task. After the preprocessing and feature extraction, the DBN model is used to classify sentences as important or unimportant sentences. The process of classifying sentences is necessary to reduce the number of sentences in the second phase. It is well recognized that one of the issues with the graph is the abundance of sentences, which can make the graph larger and have an impact on the sentences' scores. Thus, reducing the number of sentences increases the system performance. In this paper, a comparison study was performed to show the outperform of the proposed system compared with

DOI: 10.31695/IJASRE.2022.8.8.7

seven types researches. The F-measure scores reached by the proposed model and those of the seven participated systems on DUC 2004 are displayed in Table 2 in terms of ROUGE-1 and ROUGE-2.

Table 2. F-measure scores of ROUGE-1, ROUGE-2.

ID	ROUGE-1	ROUGE-2
Baseline	0.3212	0.0640
MEDLAB_Fudan	0.37584	0.0839
Graphsum[28]	0.13	0.093
Redundancy-MDS[29]	0.377	0.077
LEXRank[[30]	0.3784	0.0857
WHAASum[31]	0.4167	0.0956
MDS-OP[32]	0.3861	0.0879
Proposed model	0.4032	0.1021

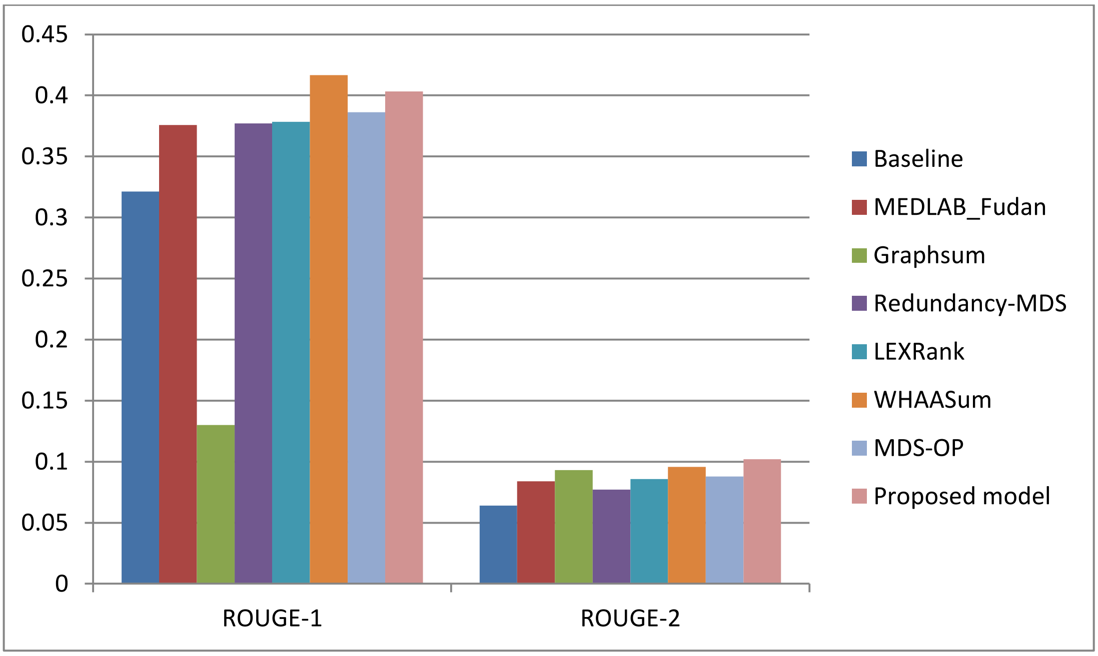

Figure 4: F-measure scores of ROUGE-1, ROUGE-2.

It's clear from Table 2 and figure 4 that WHAASum is the best method for ROUGE-1 measure. The proposed model is the second one and outperforms six systems in terms of ROUGE-1. The proposed model achieved the best ROUGE-2 score in comparison to the 6 participated systems and a baseline system. The effectiveness of the proposed system is shown through ROUGE-2 because it's closer to human summary than ROUGE-1.

7. CONCLUSION

Text summarization is a research field with many potential applications. This paper proposed a model for extractive, generic MDS. The model is based on extracting nine features from each sentence in the document collection. The feature vector is introduced as input to the DBN to classify as an important or unimportant sentence. A graph was constructed for the important sentences. When a graph has been constructed the relationship between sentences is integrated Which can more comprehensively evaluate the importance of sentences in documents, this would improve the accuracy of sentence scoring. The higher score sentences were selected to create the document summary. The experimental results on DUC-2004 show the good performance of the proposed model, especially for the ROUGE-2 measure which reached 0.1021. The main reason for the good performance of the proposed system is the high efficiency of sentence classification. Which reduces the number of sentences in the graph and

DOI: 10.31695/IJASRE.2022.8.8.7

eliminate unimportant sentences. These unimportant sentences may affect some sentences scores increasing their opportunity to be included in the created summary.

REFERENCES

[1] Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., & Pérez, C. J. (2021). The impact of term-weighting schemes and similarity measures on extractive multi-document text summarization. Expert Systems with Applications, 169, 114510.

[2] Anand, D., & Wagh, R.(2019). Effective Deep Learning Approaches for Summarization of Legal Texts, Journal of King Saud University - Computer and Information Sciences, doi: https://doi.org/10.1016/ j.jksuci.2019.11.015.

[3] CONGBO, M, WEI EMMA Z., MINGYU, G., & QUAN Z. SHENG.(2020). Multi-document Summarization via Deep Learning Techniques: A Survey. Vol.1, No. 1 35 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn.

[4] Wafaa S. El-Kassas , Cherif R. Salama , Ahmed A. Rafea, Hoda K. Mohamed ,(2021). Automatic text summarization: A comprehensive survey.Expert Systems with Applications. Vol.165.

[5] Pramita , A., Rustad, S., Fajar Shidik, G., Noersasongko, E., Syukur, A., Affandy, A., & Rosal , D. (2020). Review of Automatic Text Summarization Techniques & Methods. Journal of King Saud University - Computer and Information sciences. doi:10.1016/j.jksuci.2020.05.00610.1016/j.jksuci.2020.05.006.

[6] Tameem, A, Sayyed, A., Nesar, A., Areeba ,A., & Lakshita M.(2020). News Article Summarization: Analysis and Experiments on Basic Extractive Algorithms. International Journal of Grid and Distributed Computing Vol. 13, No. 2, pp. 2366-2379.

[7] Bidoki, M., Moosavi, M. R., & Fakhrahmad, M. (2020). A semantic approach to extractive multi-document summarization: Applying sentence expansion for tuning of conceptual densities. Information Processing & Management, Vol. 57, No.6.

[8] 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.

[9] Parveen,D., Hans-Martin R.(2015). Topical Coherence for Graph-based Extractive Summarization. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1949–1954 .

[10] Yasunaga,M., Rui, Z., Kshitijh, M., Ayush, P., Krishnan, S.,& Dragomir, R.(2017). Graph-based Neural Multi-Document Summarization. https://doi.org/10.48550/arXiv.1706.06681.

[11] Alzuhair, A., & Al-Dhelaan, M. (2019). An approach for combining multiple weighting schemes and ranking methods in graph-based multi-document summarization. IEEE Access, 7, 120375-120386.

[12] Singh,K., Manish, G., &Vasudeva, V.(2018). Unity in Diversity: Learning Distributed Heterogeneous Sentence Representation for Extractive Summarization . The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).

[13] Cho.S, Logan, L., Hassan F., & Fei Liu.(2019). Improving the Similarity Measure of Determinantal Point Processes for Extractive Multi-Document Summarization. In Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019). Florence, Italy, 1027–1038.

[14] Mao, Y., Yanru Qu, Yiqing Xie, Xiang Ren, & Jiawei Han.(2020). Multi-document Summarization with Maximal Marginal Relevance-guided Reinforcement Learning. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020). Online, 1737–1751.

[15] Brakel,P., Dieleman,S., & Schrauwen, B.(2012). Training restricted Boltzmann machines with multi-tempering: harnessing parallelization. In M. Verleysen, editor, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pages 287–292. Evere, Belgium: d-side publications.

[16] Br¨ugge,K., Fischer, A., & C. Igel.(2012). The flip-the-state transition operator for restricted Boltzmann machines. Machine Learning 13, pp. 53-69, 2013. K. Br¨ugge, A. Fischer, and C. Igel. The flip-the-state transition operator for restricted Boltzmann machines. Machine Learning 13, pp. 53-69.

DOI: 10.31695/IJASRE.2022.8.8.7

[17] Aurélien, D. & Cyril, F.(2020). Restricted Boltzmann Machine, recent advances and mean-field theory. Chinese Physics B, IOP Publishing, 2020, ff10.1088/1674-1056/abd160ff. ffhal-03143314f.

[18] Mingyang ,J. , Yanchun, L., Xiaoyue, F., Xiaojing, F., Zhili, P. & Xu, R.(2016). Text classification based on deep belief network and softmax regression. Neural Comput & Applic.

[19] Liu T (2010) A novel text classification approach based on deep belief network. In: Proceedings of the 17th international conference on neural information processing, pp 314–321. doi:10.1007/978-3-642-17537-4_39.

[20] Zhou S, Chen Q, Wang X (2014) Active semi-supervised learning method with hybrid deep belief networks. PLoS One 9(9):e107122. doi:10.1371/journal.pone.0107122.

[21] Uçkan, T., & Karcı, A. (2020). Extractive multi-document text summarization based on graph independent sets. Egyptian Informatics Journal. doi:10.1016/j.eij.2019.12.002.

[22] Ferreira, Rafael; de Souza Cabral, Luciano; Lins, Rafael Dueire; Pereira e Silva, Gabriel; Freitas, Fred; Cavalcanti, George D.C.; Lima, Rinaldo; Simske, Steven J.; Favaro, Luciano (2014). Assessing sentence scoring techniques for extractive text summarization. Expert Systems with Applications, Vol.40, No.14,p.5755–5764. doi:10.1016/j.eswa.2013.04.023 .

[23] Qaroush, Aziz; Abu Farha, Ibrahim; Ghanem, Wasel; Washaha, Mahdi; Maali, Eman (2019). An efficient single document Arabic text summarization using a combination of statistical and semantic features. Journal of King Saud University - Computer and Information Sciences, (), S1319157818310498–. doi:10.1016/j.jksuci.2019.03.010.

[24] Suphakit, N., Jatsada, S., Ekkachai, N. & Supachanun, W.(2013). Using of Jaccard Coefficient for Keywords Similarity. Proceedings of the International MultiConference of Engineers and Computer Scientists Vol I, Hong Kong.

[25] Page L, Brin S, Motwani R, Winograd T.(1999). The PageRank citation ranking: Bringing order to the web. Stanford Info Lab.

[26] Nobata, C.; Sekine, S. CRL/NYU summarization system at DUC-2004. In Proceedings of the 2004 Document Understanding Conference, Boston, MA, USA, 6–7 May 2004; NIST: Gaithersburg, MD, USA, 2004.

[27] Lin, Chin-Yew (2004). ROUGE: A Package for Automatic Evaluation of summaries. In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004).

[28] Baralis, E., Cagliero, L., Mahoto, N., Fiori, A,(2013) "Graphsum: Discovering Correlations Among Multiple Terms for Graphbased Summarization", Journal of Information Sciences, pp. 1-14.

[29] Hiram, C., Pabel, C. & Alexander, G.(2018). On redundancy in multi-document summarization. Journal of Intelligent & Fuzzy Systems. Vol.34, p.3245–3255.

[30] Erkan, G., Radev, D.R., (2004). Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. 22, p.457–479.

[31] Canhasi,E. & Kononenko,I. (2016). Weighted hierarchical archetypal analysis for multi-document summarization, Computer Speech and Language, Vol. 3,P, 24–46.

[32] Al-Saleh,A. & Menai, M. (2018). Solving Multi-Document Summarization as an Orienteering Problem. Algorithms,Vol. 11, NO.7.

DOI: 10.31695/IJASRE.2022.8.8.7