IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 28, 2019, accepted August 12, 2019, date of publication August 22, 2019, date of current version September 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936832

An Approach for Combining Multiple Weighting Schemes and Ranking Methods in Graph-Based Multi-Document Summarization

ABEER ALZUHAIR AND MOHAMMED AL-DHELAAN
Department of Computer Science, King Saud University, Riyadh 11362, Saudi Arabia

Corresponding author: Abeer Alzuhair (abeer.alzuhair@yahoo.com)

This work was supported by the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia through research group no. RGP-264.

ABSTRACT Automatic text summarization aims to reduce the document text size by building a brief and voluble summary that has the most important ideas in that document. Through the years, many approaches were proposed to improve the automatic text summarization results; the graph-based method for sentence ranking is considered one of the most important approaches in this field. However, most of these approaches rely on only one weighting scheme and one ranking method, which may cause some limitations in their systems. In this paper, we focus on combining multiple graph-based approaches to improve the results of generic, extractive, and multi-document summarization. This improvement results in more accurate summaries, which could be used as a significant part of some natural language applications. We develop and experiment with two graph-based approaches that combine four weighting schemes and two ranking methods in one graph framework. To combine these methods, we propose taking the average of their results using the arithmetic mean and the harmonic mean. We evaluate our proposed approaches using DUC 2003 & DUC 2004 dataset and measure the performance using ROUGE evaluation toolkit. Our experiments demonstrate that using the harmonic mean in combining weighting schemes outperform the arithmetic mean and show a good improvement over the baselines and many state-of-the-art systems.

INDEX TERMS Combining ranking methods, combining weighting schemes, graph-based summarization, multi-document summarization.

I. INTRODUCTION

There is no doubt that the need for automatic text summarization nowadays is very arising because of the huge increase in the information available around the world. Automatic summarizers are designed to reduce the document text size by building a summary that has the most important ideas in that document and can give a better understanding of a lot of information in a very short time. Generally, the process of text summarization can be classified as *single document summarization* where systems generate a summary using only one input document or *multi-document summarization* in which systems can create one summary using several input documents that fall under the same topic. Moreover, text summarization can also be classified as *generic summarization* where systems use all the important information of the input documents in the generated summary or *query-focused summarization* in which systems summarize just the information in the input documents which is related to a particular user query.

Usually, writing an abstractive summary is considered difficult for automatic summarization because it requires the ability to reorganize, customize, and mix information that are found in different sentences in the input document. For that, most of the current automatic summarization systems depend on extracting sentences from the document [1], in which the system finds the most important information by picking top-ranked sentences from the input and presents them exactly as they occur in the documents to be summarized. Through the years, many important works with different approaches were proposed to identify the important sentences for extractive summarization such as supervised approaches,

The associate editor coordinating the review of this article and approving it for publication was Shuping He.

sentences clustering, and graph-based methods. In this paper, we aim to extract summary sentences using a graph-based ranking method for generic multi-document summarization.

Graph-based methods for sentence ranking have the advantage of using knowledge drawn from the entire text in making ranking decisions instead of depending only on local sentence information. Also, graph-based methods are fully unsupervised and depend simply on the text to be summarized without the need for any training data. In such approaches, the input text is described as a highly connected graph where similar sentences vote for each other and very important or main sentences can be estimated using some common graph-based ranking algorithm.

In this research, we intend to use a graph-based ranking method to enhance the results of multi-document summarization which will result in more accurate summaries that could be used as a significant part of some natural language applications that involves, e.g., products recommendations, news summarization, and search engines results. However, graph-based ranking needs to define some measure to calculate the similarity between two sentences and use it as the weight of the edges between the graph vertices that represent those sentences. It also needs to use some ranking technique to sort the sentences according to their importance. Many important works have been done before in this area that showed very encouraging results such as TextRank [4] and LexRank [3]. However, these approaches may have some limitations because they depended only on one weighting scheme which has its strengths but could have some weaknesses and also, they ranked the sentences using only one ranking method which might be good in some area but can be weak in another one. And since our goal is to enhance the multi-document summarization results, we propose to expand the previous work of TextRank [4] and LexRank [3] by combining multiple weighting schemes and multiple ranking methods in one graph framework.

Particularly, we propose to compute the edge weights using a combination of four well-known unequal weighting schemes, which are: Jaccard similarity coefficient, TF*IDF cosine similarity, Topic signatures similarity, and the Identity similarity measure. We also propose to enhance the sentence ranking scores by combining two of the most important graph-based ranking methods which are: PageRank algorithm [5] and HITS algorithm [7]. To combine these methods, we suggest taking an ‘‘average’’ value of their results using two ways of the Pythagorean means which are: the arithmetic mean and the harmonic mean.

The motivation behind combining multiple approaches is that it is better to rely on multiple signals instead of relying on only one because it could take the best of each method and avoid the weaknesses that come from each one of them. Besides, we can consider that taking the average scores is like using a voting system wherein only sentences with high similarity in all weighting measures will get high results. And the sentence will get a great ranking score and considered important if it is important in all ranking methods.

To show the effectiveness of our proposed approaches, we measure the performance of both methods separately and jointly and compare their results with two important baselines (TextRank [4] and LexRank [3]). For evaluation, we exploit a benchmark dataset that is commonly used by the community for text summarization, the (DUC 2003 & DUC 2004) dataset. Moreover, we measure the performance of our proposed approaches using the ROUGE evaluation toolkit, which is a very important and effective measure that is found to be correlated with human evaluations [9].

The rest of this paper is organized as follows: *Section II* presents some of the most important work in the graph-based summarization. *Section III* describes our proposed approach and algorithms design. *Section IV* presents and analyzes the evaluation results of all our experiments. Finally, *Section V* concludes our work and suggests future work directions.

II. RELATED WORK

Ever since the need to solve the automatic text summarization arose, many important works with different approaches were proposed, such as: using labeled data to train a supervised statistical classifier for distinguishing important sentences [22], [26], [28], using data-driven neural networks techniques to avoid the dependence on human-engineered features for sentences extraction [30], [31], using sentence clustering to make groups of similar sentences then choose one representative sentence from every main cluster [32]–[34], or using optimization algorithms to identify the most important sentences while eliminating redundant sentences and maintaining the summary under specific length [13], [19], [35].

Graph-based methods for sentence ranking is considered one of the most important approaches that have attracted the attention of many researchers in this field [18], [20], [23]. Two of the most important work on graph-based methods that show very encouraging results in sentence ranking are TextRank [4] and LexRank [3]. Both works identify the sentences in the text and add them as vertices in a weighted undirected graph then draw edges between several sentence pairs in the text based on their similarity relation. In TextRank, the similarity relation can be determined simply as the number of common words between two sentences divided by their length for normalization. Wherein LexRank, the similarity between sentences is computed using the TF*IDF cosine similarity measure. Then to find the most important sentences in the text, both TextRank and LexRank propose to rank the graph vertices in a random walk framework using the PageRank algorithm [5]. Moreover, in an extended work of TextRank [6], the graph vertices were ranked using two additional graph-based ranking algorithms, which are: HITS algorithm [7] and Positional Power Function algorithm [8]. However, although the results of TextRank and LexRank are very encouraging, their work is heavily affected by the performance of the chosen weighting scheme and ranking method since they both depend on only one weighting scheme to build their graph and only one ranking method to rank the sentences. They both use as a weighting scheme a measure that gives weight to each word based on its frequency in the input document without considering its occurrence frequency in another background corpus, which may have some strengths but could have some weaknesses as well. Also, both works depend on the functionality of one ranking method, which might be robust in some area but can be weak in another one.

Instead of depending on the frequency of the words only, other works incorporate the syntactic and semantic role information in the process of building the graph. In [10], the similarity measures can be identified based on a syntactic tree and a shallow semantic tree in a random walk framework. Also, [25] make use of semantic graph structure for document summarization, where the vertices are words and phrases instead of sentences, and the edges are syntactic dependencies. Besides, machine-learning techniques have been combined with graph-based summarization as well. In [25], a linear Support Vector Machine (SVM) classifier was trained to work with the semantic graph structure to find the vertices that are useful for extracting summaries. Also, [27] incorporate graph representations of sentence relationships with deep neural networks using Graph Convolutional Networks (GCN). And in [21], a graph-based clustering method was used for document summarization, in which the system clusters the sentences to find how they relate to a specific topic using a document graph model.

The hypergraph-based model has been successfully used for text summarization task which is a generalization of the graph in which edges can link any number of vertices [15], [16]. This kind of models was proposed to solve the problem of traditional pairwise graph-based modeling, which is the incapability to capture complex associations among multiple sentences. Likewise, GRAPHSUM [12] summarizing system was proposed to solve the same problem of neglecting complex correlations among multiple terms. In which the system combines the graph-based approach with data mining techniques to create a correlation graph that can represent the correlations among multiple terms using association rules.

Other works also propose using untraditional graph-based structures. In [11], two kinds of links with different weighting schemes were used to connect sentences that belong to the same document (intra-link) and sentences from different documents (inter-link). This approach was proposed to distinguish between similar content within one document and repeated content across multiple documents. And instead of using a graph of one mode type where vertices are only sentences from the text, [14], [29] propose to use a graph of two-mode type (bipartite graph) that consist of two different sets of vertices, and the edges can connect only vertices from different sets.

III. PROPOSED APPROACH

Our work is based on two of the most important work on graph-based methods for sentence ranking; TextRank [4] and LexRank [3]. Graph-based methods for sentence ranking have shown to be successful for both single-document and multi-document summarization [1]. Such approaches do not involve any complex linguistic processing of the text other than identifying its sentences and words. They also have the advantage of being fully unsupervised and depend simply on the text to be summarized without the need for any training data. Moreover, graph-based methods rank the sentences based on information drawn from the entire text instead of depending only on local sentence information. And since similar sentences are linked together based on their words overlap, and then they vote for each other, the graph-based methods effectively benefit from input repetition, on both the word level and the sentence level.

To rank our sentences, we use a graph that we define as a weighted undirected graph $G(V; E; w)$ where V is the set of vertices representing the sentences to be ranked, E is the set of edges representing the relation between similar sentences, and w is the set of edge weights which represent the similarity scores.

The primary research method for our work is to experiment combining different weighting schemes and multiple ranking methods in one graph framework then analyze the findings. When combining multiple measures to compute the similarity between two sentences, our approach will be depending on multiple signals to decide if these sentences are similar or not. Based on that, it will only give the sentences a high similarity score if they have high similarity scores in all proposed measures, and we assume that this will improve the accuracy of calculating similarities in the graph. To show that our assumption is correct and to prove our point, we decide to experiment with combining four different similarity measures to produce an improved weighting scheme. We propose to combine the results of the following sentence similarity measures: Jaccard similarity coefficient, TF*IDF cosine similarity, Topic signatures similarity, and the Identity similarity measure. We choose to start with these measures as they easy to implement, yet they are strong enough and have shown good results in computing sentences similarity [36]. However, choosing those four measures is just meant for experimentation, and our idea is not limited to them. And in the future, we plan to add other sentence similarity measures to the combination process, such as Semantic similarity and Word2Vec cosine similarity. Furthermore, by continuing on the same principle, we propose to combine two graph-based ranking algorithms to produce an enhanced ranking technique that can help in sorting sentences according to their importance. To do that, we choose to combine the results of PageRank algorithm [5] and HITS algorithm [7] as they are popular in the community of document summarization and found to be very successful in many ranking applications.

A. WEIGHTING SCHEMES

Weighting schemes are measures that can define the weight of the edge between any two vertices by computing the similarity of their sentences. In this paper, we use four important weighting schemes to calculate the similarity between our sentences, which we discuss below.

1) JACCARD SIMILARITY COEFFICIENT

This weighting scheme is simple and commonly used to measure the content overlap, in which any two sentences are considered similar if they have terms in common. The similarity score can be determined simply as the number of words in common in the two sentences divided by the number of all unique words in those sentences. Formally, using the sentences S_1 and S_2, where each sentence is represented as a finite set of words, their Jaccard similarity is described as the size of the intersection of the words between S_1 and S_2 divided by the size of the union of the words in both sentences:

$$Sim_{(S_1,S_2)} = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \tag{1}$$

2) TF*IDF COSINE SIMILARITY

Cosine similarity is one of the most popular and successful similarity measures. It works based on a vector space model, where sentences are considered as finite dimensional vectors, and the weight of each term in a sentence is computed using the TF*IDF weighting scheme. It calculates the similarity between two sentences as the cosine of the angle between their corresponding vectors. Formally, for any two sentences S_1 and S_2, the cosine similarity between them is calculated as the dot product of these vectors divided by the product of their Euclidean lengths:

$$Sim_{(S_1,S_2)} = \frac{\sum\limits_{t\in S_1,S_2} (tf\text{-}idf_{t,S_1})(tf\text{-}idf_{t,S_2})}{\sqrt{\sum\limits_{t_1\in S_1} (tf\text{-}idf_{t_1,S_1})^2}\sqrt{\sum\limits_{t_2\in S_2} (tf\text{-}idf_{t_2,S_2})^2}} \tag{2}$$

In the above equation, $tf\text{-}idf_{t,S}$ is the weight of term t in sentence S, that is defined as the number of occurrences of term t in sentence S (term frequency $tf_{t,S}$) multiplied by the inverse document frequency of that term (idf_t); noticing that the TF*IDF weighting scheme will assign a weight to each term based on its number of occurrences in the whole document instead of only relying on its presence in the sentence.

3) TOPIC SIGNATURES SIMILARITY

This weighting scheme depends on the most descriptive words in the input document to measure the similarity between sentences; these words are used to determine the importance of the sentences while other words are completely ignored in the calculations. These descriptive words are usually called "topic signatures" in the summarization literature [17], which are words that frequently appear in the input but are rare in other texts. The topic signatures can be statistically found using the log-likelihood ratio test which will separate all the input words into either descriptive or not by comparing their frequency of occurrence in the input document with their frequency in a large background corpus.

For any given word w in the input document D, the log likelihood ratio $\lambda(w)$ is computed via the binomial distribution formula as the ratio between observing the occurrence probability $P(w)$, in both the document to be summarized and the background corpus. Then, after obtaining the log likelihood ratio λ for each word w, it can be statistically classified into either descriptive or not if its likelihood statistic value $(-2log\lambda(w))$ is greater than a cutoff threshold with the value of (10.83) which is an indicator of high statistical significance and has a confidence level of (0.001) [17].

Finally, based on the log-likelihood ratio test, the similarity between any two sentences can be calculated as the cosine similarity of only the topic signatures in the two sentences and any other words will be completely ignored.

4) IDENTITY SIMILARITY

The identity measure [24] is a similarity measure that was initially developed for identifying the "co-derivative documents" which are documents that are derived from the same source such as plagiarized documents and documents with several versions. It has been shown to work well and to be very useful for these kinds of applications. Like the cosine similarity, this measure depends on using the TF*IDF scheme to give weight to each term in the document. However, the identity measure works under the concept of measuring the common contents in the documents; unlike the cosine similarity which is designed to measure how much the documents are different. Formally, for any two sentences S_1 and S_2, the identity measure is described as:

$$Sim_{(S_1,S_2)} = \frac{1}{1+|L_{S_1} - L_{S_2}|} \sum_{t\in S_1\cap S_2} \frac{idf_t}{1+|tf_{t,S_1} - tf_{t,S_2}|} \tag{3}$$

where idf_t is the inverse document frequency of term t, $tf_{t,S}$ is the number of occurrences of term t in sentence S, and L_S is the length or the total number of terms in S.

However, in this measure, the similarity results were not normalized originally. And for our experiments, all the similarity results must be normalized between $(0 - 1)$, so that the exactly similar sentences get the score 1 and the sentences that are non-similar get the score 0. So, to normalize our results, we need to do a simple modification on the Identity formula. Thus, we update the original formula by adding the *IDF* factor in the denominator, which will force all similarity scores to be normalized. Formally, for any two sentences S_1 and S_2, the modified identity similarity measure is described as:

$$Sim_{(S_1,S_2)} = \frac{1}{\sum\limits_{t\in S_1\cap S_2} idf_t + |L_{S_1} - L_{S_2}|} \sum_{t\in S_1\cap S_2} \frac{idf_t}{1+|tf_{t,S_1} - tf_{t,S_2}|} \tag{4}$$

B. GRAPH-BASED RANKING ALGORITHMS

Graph-based ranking algorithms are techniques that determine how important is a vertex within a graph based on information taken from the entire graph. In this paper, we use two graph-based ranking algorithms that are found to be successful in many ranking applications. These algorithms were originally designed for the directed graphs; however, they can be modified to work with undirected and weighted graphs.

1) PAGERANK ALGORITHM

PageRank [5] is one of the most successful ranking algorithms; it is an iterative link analysis algorithm that was introduced to rank Web pages. It computes the ranking score for each vertex in the graph based on the probability of being in that vertex at time t while making consecutive moves from one vertex to another random vertex (random walk).

In this work, we compute the ranking scores using a modified PageRank rule that can work with weighted undirected graphs. This rule starts by assigning arbitrary values to each vertex in the graph. It uses the links weights (similarity scores) to calculate the probability of transitioning from one vertex to another. Then as more and more transitions are made, the computation iterates until the probability of each vertex converges. This rule is defined as follows:

$$PR(V_i) = \frac{(1-d)}{N} + d * \sum_{V_j \in adj(V_i)} \frac{Sim(V_i, V_j)}{\sum_{V_z \in adj(V_j)} Sim(V_j, V_z)} PR(V_j) \quad (5)$$

where $PR(V_i)$ is the ranking score assigned to vertex V_i, N is the total number of vertices that used as a "normalization factor", $adj(V_i)$ is the set of neighboring vertices of V_i, and d is a "damping factor" which is the probability to teleport the random walk (we choose to set the damping factor value at 0.85 as the literature suggests [5]).

This way, while computing PageRank score for a sentence, the rule multiplies the PageRank scores of the linking sentences by the weights of the links. Also, to rank the weighted graph using PageRank all the links weights must be normalized to form a probability distribution (i.e. the weights of all links connected to one vertex sum up to one). By doing that, the graph becomes a Markov chain, and the links weights can be used as the probability of transitioning from one vertex to another. Finally, after reaching convergence, the vertices with higher probabilities will be considered more important within the graph.

2) HITS ALGORITHM

Hyperlink Induced Topic Search [7] (also known as hubs and authorities) is another iterative link analysis algorithm that was introduced to rank Web pages. In this paper, we also use a modified version of the HITS algorithm that can take into account edge weights when computing the ranking scores. Usually, this algorithm defines two values for each vertex: The Authority value (value of the incoming links) and The Hub value (value of the outgoing links). It computes the values of the Authority and the Hub in a mutual recursion based on each other. Formally, let w_{ij} be the edge weight that connects the two vertices V_i and V_j, then the Authority and the Hub formulas can be expressed as follows:

$$HITS_A^W(V_i) = \sum_{V_j \in adj(V_i)} w_{ji}\, HITS_H^W(V_j) \quad (6)$$

$$HITS_H^W(V_i) = \sum_{V_j \in adj(V_i)} w_{ij}\, HITS_A^W(V_j) \quad (7)$$

The algorithm starts by giving each vertex a Hub and Authority values of 1. Then it updates the Authority scores using the $HITS_A^W(V_i)$ formula as well as the Hub scores using the $HITS_H^W(V_i)$ formula and normalizes the results. This process will be repeated until the scores have come to consistent values (convergence). Finally, after reaching convergence, the algorithm returns two sets of scores as an output; the Authority scores set and the Hub scores set. However, in the case of undirected graphs, the Authority and the Hub results will be exactly the same. Therefore, we can use only one of them to rank the graph vertices.

C. ALGORITHM DESIGN

In this work, we develop two algorithms to improve and enhance the automatic multi-document summarization results using graph-based methods. For the first algorithm (**Algorithm-1**), instead of using only one similarity measure, we suggest combining four effective and well-known weighting schemes by taking the average of their results for each pair of sentences using the arithmetic mean and once again using the harmonic mean. And for the second algorithm (**Algorithm-2**), we propose to enhance the sentence ranking by combining two of the most important graph-based ranking methods, also by taking the average of their results for each vertex in the graph using the harmonic mean.

IV. EXPERIMENTS AND RESULTS

This section presents and explains the experiments performed in this work. It also shows an evaluation of our approach and compares the accuracy of our generated extractive summaries with two baselines and different state-of-the-art systems.

A. DATASET AND PREPROCESSING

For our experiments, we use task number 2 of both (DUC 2003 & DUC 2004) datasets,[1] which is a generic multi-document summarization task that was created of news articles in the English language. DUC 2003 consist of 30 clusters and DUC 2004 consist of 50 clusters of news documents. Each cluster in both datasets comes with 3-4 golden human reference summaries; those summaries are usually used to compare with the researchers' system results. To make a fair evaluation of this comparison, it is important to set a limit

[1] Created by *NIST*, https://duc.nist.gov/

Algorithm 1 Computing Sentences Scores Using Average Weighting

```
Input  : An array $S$ of $n$ sentences
output: An array of sentences' scores
Array $AvgWeightMatrix[n][n]$;
Array $Scores[n]$;
for i ← 1 to n do
    for j ← 1 to n do
        jc = jaccard-similarity($S[i]$,$S[j]$);
        cs=tf*idf-cosine-similarity($S[i]$,$S[j]$);
        ts = topic-signatures-similarity ($S[i]$,$S[j]$);
        id = identity-similarity($S[i]$,$S[j]$);
        $AvgWeightMatrix[i][j]$ = AvgValue(jc,cs, ts,id);
    end
end
$Scores$ = Graph-based-ranking($AvgWeightMatrix$);
return $Scores$;
```

Algorithm 2 Computing Sentences Scores Using Average Ranking

```
Input  : An array $S$ of $n$ sentences
output: An array of sentences' scores
Array $SimMatrix[n][n]$;
Array $PageRank[n]$; Array $HITS[n]$; Array $Scores[n]$;
for i ← 1 to n do
    for j ← 1 to n do
        $SimMatrix[i][j]$ = MeasureSimilarity ($S[i]$,$S[j]$);
    end
end
$PageRank$ = PageRank( $SimMatrix$);
$HITS$ = Hyperlinked-Induced-Topic-Search
($SimMatrix$);
for i ← 1 to n do
    $Scores[i]$ = AverageValue( $PageRank[i]$, $HITS[i]$);
end
return $Scores$;
```

on the length of the extracted summaries. For that, we set the length of each summary in the DUC 2003 clusters to 100 words, and in the DUC 2004 clusters to 665 bytes due to the choice of the DUC 2003 & DUC 2004 organizers for gold summaries.

Besides, in the graph-based extractive summarization, the preprocessing stage is essential as it has a significant effect on the accuracy of the similarity scores calculations. Therefore, we perform some suitable preprocessing steps to all text documents in each dataset. Originally all text documents in both datasets were tagged to identify the document source information from the textual components to be processed. So, as a first step in the preprocessing stage we remove all the informational tags like (<DOC>, <TEXT>, <p>, ...etc.) and we extract only the text we need to process from the documents. After that, we perform some general and essential preprocessing steps which are:

1) Split each document into a list of sentences.
2) Split each sentence into a list of words.
3) Convert all capitalized words to lower case.
4) Remove all the stop words.
5) Remove all the punctuations.
6) Convert each word into its corresponding stem.

B. EVALUATION METRIC

To measure the performance of our experiments, we use a very important and effective evaluation toolkit called ROUGE-N [9], which is a recall-based metric that relies on N-gram statistics and used with fixed length summaries. It measures the efficiency of the automatically generated summaries by comparing it with the golden summaries made by humans and finding the number of the n-grams overlapping between them. The scores produced by ROUGE-N measure change based on the number of successive terms used for comparison. For our experiments, we use the ROUGE-1 and ROUGE-2 measures which use one term and two terms for comparison respectively. We selected these two measures due to their common use in other works.

C. EXPERIMENTAL SETUP

After the preprocessing step, our proposed approach extracts the summaries through three main steps: sentences similarity, sentences ranking, and sentences selection (as shown in Fig.1).

1) COMPUTE SENTENCES SIMILARITY

For our graph, we calculate the weights of the edges by combining four different similarity measures which are: Jaccard similarity coefficient, TF*IDF cosine similarity, Topic signatures similarity, and the Identity similarity measure. So, for every pair of sentences, we compute four similarity scores and stored their results so that we can combine them afterwards. To compute these scores, we use all four measures as explained in the previous section. However, in the Topic signatures similarity, we need to compare the occurrence probability of the words in the input cluster against some background corpus to find the most descriptive words. To do that, for every input cluster, we use the rest of all clusters from the same dataset as a background corpus. For example, if we want to summarize the first cluster in DUC 2003 dataset, we will use as a background all 29 remaining clusters in this dataset.

Based on the scores of those four measures, we conduct two experiments to find our new combined weighting schemes. In these experiments, we adopt two ways of the Pythagorean means to get an "average" value of the similarity scores. In the first experiment, we use the Arithmetic mean, where we simply find the average value by adding all proposed similarity scores, then dividing the result by their number. We choose to use the arithmetic mean as it is the most common measure to find the average value, although it is greatly influenced by outliers. In the second experiment, we use the Harmonic mean, which can be described as the multiplicative inverse of the arithmetic mean of the multiplicative inverses of the dataset. Choosing to use the harmonic mean in this part of our experiment is due to its

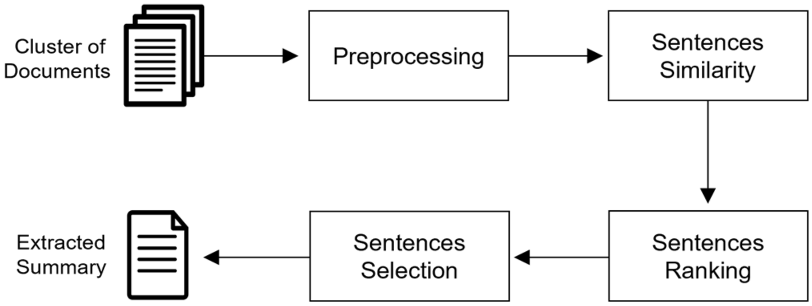

FIGURE 1. Sentence extraction process.

tendency to be more conservative than the arithmetic mean, and thus reducing the impact of large outliers. Moreover, in each experiment of those, at first, we compute the average values using all proposed similarity measures. Then to study the effect of each similarity measure, we repeat the experiment several times where we remove one measure at a time.

2) COMPUTE RANKING SCORES

After we calculate all the similarity scores, we need to rank our sentences so we can extract the most central ones. To do that, we first convert each cluster of documents to a weighted undirected graph where sentences are vertices, and weighted edges are formed by connecting sentences using the similarity scores. Then, we use two important graph-based ranking algorithms to rank the vertices of the graph which are the PageRank algorithm [5] and the HITS algorithm [7].

Based on that, we perform three experiments to rank our sentences. In the first and the second experiments, we compute the ranking scores using the modified methods of PageRank and HITS respectively; both methods are used as explained in the previous section. In the third and final experiment, we try to enhance the sentence ranking by combining these two methods by taking the average of their results for each vertex in the graph using the harmonic mean approach.

3) SENTENCES SELECTION

After computing the ranking scores for all sentences, we sort the sentences in descending order then we extract the most central sentences that have the highest scores and include them into the summary until we reach the required limit of the summary length. However, since our work is a multi-document summarization, it is important to ensure that the extracted sentences do not have redundant information. So, to reduce the redundancy, we prevent any candidate sentence to be included in the summary if the cosine similarity score between it and any one of the previously extracted summary sentences is more than a pre-defined threshold (as shown in **Algorithm-3**). In our experiments, we set the threshold value at (0.7) based on a previous study [15].

Algorithm 3 Selecting Best Sentences for a Summary

```
Input  : A list $S$ of $n$ ranked sentences
output: An extracted summary
$Summary = []$;
$Sorted\_List$ = sort the sentences in $S$ based
                on their ranking scores;
while (length($Summary$) $<$ $limit$) do
    $next\_sen$ = remove the highest ranked
                 sentence from $Sorted\_List$;
    for ($sen \in Summary$) do
        if( $CosineSimilarity(sen, next\_sen) > threshold$):
            break;
        else:
            $Summary = Summary \cup \{next\_sen\}$;
    end
end
return $Summary$;
```

D. EXPERIMENTAL RESULTS

In the first phase of our experiments, we study the effect of combining multiple similarity measures on the summarization results. We test two methods to combine the similarity measures which are: arithmetic mean, and harmonic mean. For each method, we use the PageRank algorithm to rank the sentences and compare the results with two baselines (TextRank [4] and LexRank [3]).

The results in Table.1 & Table.2 show that the arithmetic mean approach outperformed the TextRank baseline in both datasets. For ROUGE-1, it showed an improvement that ranges from 0.43% to 1.14% in DUC 2003 and from 0.57% to 1.50% in DUC 2004. However, this approach did not show any improvement over LexRank in DUC 2003, and in DUC 2004 it showed a slight improvement that ranges from 0.15% to 0.88%.

Nevertheless, the harmonic mean approach showed much improvement and outperformed all baselines in both datasets. For ROUGE-1, when we use all proposed similarity measures, this approach has a 1.82% improvement over TextRank and 0.59% over LexRank with DUC 2003. Also, with DUC 2004, it has a 1.53% improvement over TextRank and

TABLE 1. Summarization results of (DUC 2003 dataset) using *PageRank*.

Methods	ROUGE-1	ROUGE-2
LexRank	0.36812	0.08554
TextRank	0.35579	0.08394
Proposed Methods with "*Arithmetic Mean*"		
All Similarity Measures Included	0.36502	0.08775
Identity-Similarity Removed	0.36010	0.08641
TopicSignatures-Similarity Removed	0.36718	0.08688
Jaccard-Similarity Removed	0.36576	0.08716
Cosine-Similarity Removed	0.36378	0.08679
Proposed Methods with "*Harmonic Mean*"		
All Similarity Measures Included	**0.37399**	0.09360
Identity-Similarity Removed	0.36497	0.08795
TopicSignatures-Similarity Removed	0.37047	0.08973
Jaccard-Similarity Removed	**0.37428**	0.09353
Cosine-Similarity Removed	0.37051	0.09424

TABLE 2. Summarization results of (DUC 2004 dataset) using *PageRank*.

Methods	ROUGE-1	ROUGE-2
LexRank	0.38208	0.09423
TextRank	0.37581	0.09022
Proposed Methods with "*Arithmetic Mean*"		
All Similarity Measures Included	0.38896	0.08903
Identity-Similarity Removed	0.39085	0.09641
TopicSignatures-Similarity Removed	0.38149	0.08440
Jaccard-Similarity Removed	0.38801	0.08722
Cosine-Similarity Removed	0.38354	0.08447
Proposed Methods with "*Harmonic Mean*"		
All Similarity Measures Included	**0.39107**	0.09609
Identity-Similarity Removed	0.38964	0.09675
TopicSignatures-Similarity Removed	0.38942	0.09588
Jaccard-Similarity Removed	**0.39300**	0.09983
Cosine-Similarity Removed	0.38472	0.09235

0.90% over LexRank. However, our experiments show that the harmonic mean approach gives its best results if we remove the Jaccard similarity from the proposed combination. This method showed an improvement of 1.85% and 0.62% over TextRank and LexRank respectively in DUC 2003, as well as 1.72% over TextRank and 1.09% over LexRank in DUC 2004. On the other hand, the results show that removing any other similarity measure from the proposed combination will not give as much improvement as removing the Jaccard similarity.

Moreover, we can also see in the ROUGE-2 scores that the harmonic mean approach performed the best and outperformed both baselines. With TextRank it showed an improvement that ranges from 0.40% to 1.03% in DUC 2003 and from 0.21% to 0.96% in DUC 2004. It also showed an improvement over LexRank that ranges from 0.24% to 0.87% in DUC 2003 and from 0.17% to 0.56% in DUC 2004.

In the second phase of our experiments, we study the effect of combining two graph-based ranking methods which are: PageRank and HITS algorithms. To combine the results of those methods we use the harmonic mean approach. And we use as a graph weighting schemes all the similarity measures that we proposed before. The results in Table.3 & Table.4 show that the proposed average ranking approach did not give that much improvement compared to the PageRank approach in both datasets. For ROUGE-1, it showed a slight improvement that ranges from 0.03% to 1.0% with only four weighting schemes in DUC 2003, and from 0.03% to 0.16% with six weighting schemes in DUC 2004.

At the end of our experiments, we compare our best result on DUC 2004 dataset with the results of many state-of-the-art systems that involve: optimization model [13], supervised regression model [44], deep neural network models [39], [40] and graph-based neural network model [27]. In Table.5, we can see that the *BestCombination* of our approach has shown a very competitive performance as it obtained comparable results to *Lin&Bilmes* [13] and *SRSum* [40], and outperformed *GRU+GCN* [27] with 1.07%, *REGSUM* [44] with 0.73% and *R2N2-ILP* [39] with 0.52% in terms of ROUGE-1. We further discuss and analyze our results in the following section.

E. EXPERIMENTAL RESULTS DISCUSSION

Based on the results we presented in the previous section we can see that our proposed approach of combining different weighting schemes in one graph framework showed some improvement over both baselines. Generally, we can say that using either arithmetic or harmonic mean in combining the similarity measures has enhanced the summarization results. The reason for such improvement is that when we combine multiple successful weighting schemes, we capture their strengths and avoid the weaknesses that could come from each one them, and hence the ranking algorithm becomes more accurate.

However, our experiments show that using the harmonic mean approach gave the best results and outperformed the arithmetic mean approach. That is because the harmonic mean is a more conservative approach than the arithmetic mean and can handle the outliers much better. Fig.2 demonstrates that the similarity scores cannot always be in a normal distribution, and sometimes the scores appear to be very divergent from each other because of the outliers. Using the arithmetic mean when there is a significant outlier within the scores may skew the results and might give the edge a higher weight than it should have since it takes the middle value of all scores including the outlier. On the other hand, the harmonic mean can yield better results for the intended purposes. If the scores of the sentence are divergent from each other where

TABLE 3. Summarization results of (DUC 2003 dataset) using *PageRank*, *HITS_Rank*, and *AverageRanking*.

Methods	PageRank		HITS		AverageRanking	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
LexRank	0.36812	0.08554	0.35800	0.08796	0.36634	0.08602
TextRank	0.35579	0.08394	0.35820	0.08695	0.35470	0.08394
Proposed Methods with "*Arithmetic Mean*"						
All Similarity Measures Included	0.36502	0.08775	0.35790	0.08472	0.36453	0.08707
Identity-Similarity Removed	0.36010	0.08641	0.35480	0.08600	0.36095	0.08682
TopicSignatures-Similarity Removed	0.36718	0.08688	0.35865	0.08326	0.36360	0.08557
Jaccard-Similarity Removed	0.36576	0.08716	0.36083	0.08678	0.36677	0.08780
Cosine-Similarity Removed	0.36378	0.08679	0.35275	0.08159	0.36263	0.08696
Proposed Methods with "*Harmonic Mean*"						
All Similarity Measures Included	0.37399	0.09360	0.35529	0.08642	0.36980	0.09256
Identity-Similarity Removed	0.36497	0.08795	0.35535	0.08685	0.36342	0.08880
TopicSignatures-Similarity Removed	0.37047	0.08973	0.35789	0.08698	0.37088	0.09231
Jaccard-Similarity Removed	0.37428	0.09353	0.35720	0.08720	0.37461	0.09425
Cosine-Similarity Removed	0.37051	0.09424	0.34675	0.08439	0.36795	0.09472

TABLE 4. Summarization results of (DUC 2004 dataset) using *PageRank*, *HITS_Rank*, and *AverageRanking*.

Methods	PageRank		HITS		AverageRanking	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
LexRank	0.38208	0.09423	0.36856	0.08912	0.38041	0.09305
TextRank	0.37581	0.09022	0.37601	0.09205	0.37609	0.09055
Proposed Methods with "*Arithmetic Mean*"						
All Similarity Measures Included	0.38896	0.08903	0.38624	0.09119	0.38834	0.08924
Identity-Similarity Removed	0.39085	0.09641	0.38405	0.09269	0.38976	0.09609
TopicSignatures-Similarity Removed	0.38149	0.08440	0.38025	0.08879	0.38256	0.08526
Jaccard-Similarity Removed	0.38801	0.08722	0.38496	0.09091	0.38841	0.08833
Cosine-Similarity Removed	0.38354	0.08447	0.38665	0.09053	0.38452	0.08505
Proposed Methods with "*Harmonic Mean*"						
All Similarity Measures Included	0.39107	0.09609	0.38424	0.09372	0.39194	0.09726
Identity-Similarity Removed	0.38964	0.09675	0.38031	0.09315	0.38853	0.0964
TopicSignatures-Similarity Removed	0.38942	0.09588	0.38745	0.09447	0.38742	0.09524
Jaccard-Similarity Removed	0.39300	0.09983	0.37695	0.09106	0.39139	0.09917
Cosine-Similarity Removed	0.38472	0.09235	0.38585	0.09235	0.38629	0.09236

TABLE 5. State-of-the-art results (%) on DUC 2004 dataset.
[Results of the systems marked with the * symbol are taken from their corresponding references, and "*BestCombination*" is the proposed combination that gave us the best results using the harmonic mean when "Jaccard Similarity" removed.].

Methods		Model	Year	ROUGE-1
PEER-65		The best performing participants at DUC 2004	2004	37.88
Lin&Bilmes*	[13]	Optimization model	2011	**39.35**
REGSUM*	[44]	Supervised regression model	2014	38.57
R2N2-ILP*	[39]	Deep neural network based model	2015	38.78
SRSum*	[40]		2018	**39.29**
GRU+GCN*	[27]	Graph-based neural network model	2017	38.23
BestCombination				**39.30**

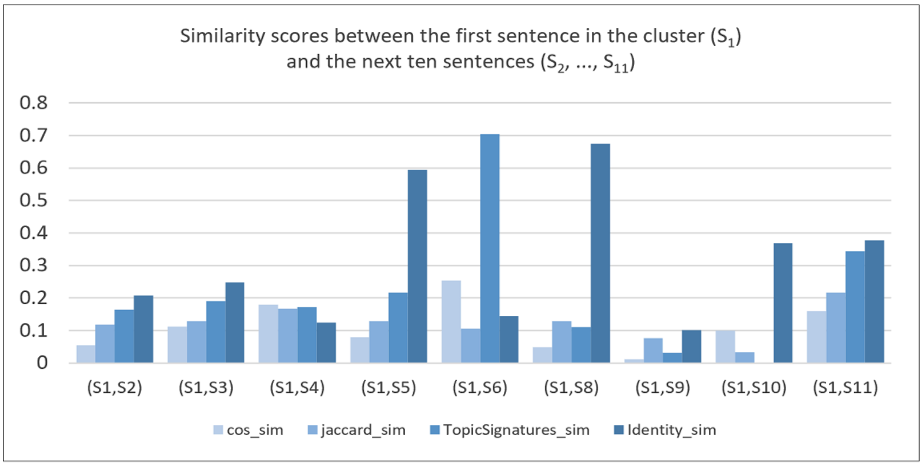

FIGURE 2. Similarity scores example from the first cluster in DUC-2004.

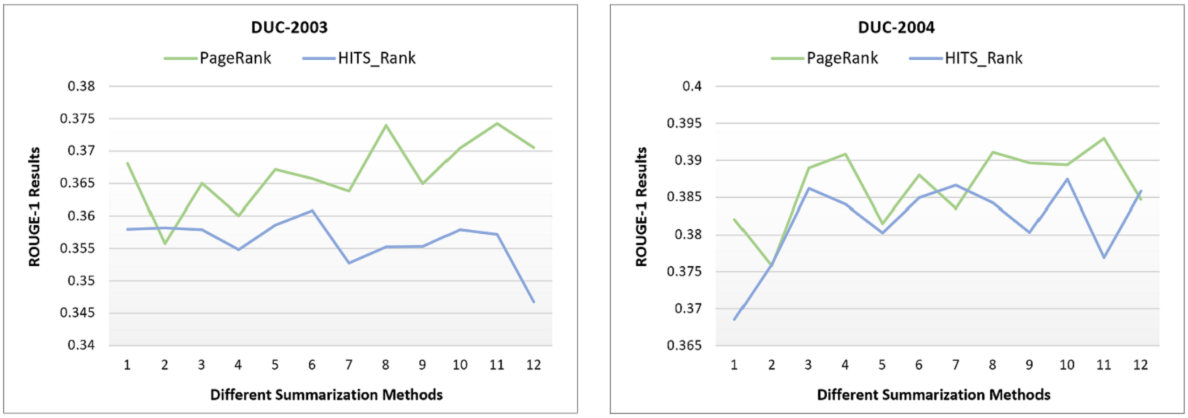

FIGURE 3. Comparing ROUGE-1 summarization results of *PageRank* and *HITS*.

some are very high, and some are very low, then it will give a lower weight to this sentence. This way, the harmonic mean will not give a higher weight to a sentence unless it has high scores in all measures.

Moreover, as we perform several experiments to study the effect of each one of the proposed weighting schemes on the results, we found that the harmonic mean approach gives its best results if we remove the Jaccard similarity from the proposed combination; whereas removing any other similarity measure will not give as much improvement. The reason why removing the Jaccard similarity gives the best performance is that the Jaccard similarity does not take term frequency into account, it only takes a unique set of terms for each sentence and does not consider how many times each term occurs in that sentence. Meaning that this measure treats all the words the same way and does not give special weight to the term based on its frequency. On the other hand, the three remaining measures worked well together because they all consider the term frequency and assign a weight to each term in the sentence using the same weighting method (TF*IDF).

Finally, we study the effect of combining PageRank and HITS ranking algorithms on the summarization results. And we found that our proposed approach of taking the average ranking scores using the harmonic mean did not give the desired improvement. That is because the PageRank algorithm gave better results than the HITS ranking algorithm in almost every method in both datasets. Thus, the HITS low results will hold back the desired improvement of combining the two algorithms. So, the proposed average ranking method did not perform well because there was no variation in the results that the combination could benefit from since PageRank is almost always better than HITS as shown in Fig.3.

In this research, our main interest was in finding out if combining different weighting schemes and multiple ranking methods in one graph framework will improve the results of multi-document summarization or not. For that, we did not do any industrial analysis of our work. However, we can say that the improvement on the summarization results that showed by our approach could help in improving the industry of some NLP applications that involve text summarization like products recommendations, news summarization, and search engines results, etc.

V. CONCLUSION AND FUTURE WORKS

In this work, we proposed an improved graph-based ranking approach to enhance the results of extractive, generic, multi-document summarization. We conclude our main contributions as follows: (1) We produced an improved weighting scheme by combining multiple important measures that calculate the similarity between two sentences, which are: Jaccard similarity coefficient, TF*IDF cosine similarity, Topic signatures similarity, and the Identity similarity measure. To combine these measures, we have experimented two different ways of results averaging, which are: the arithmetic mean and the harmonic mean. (2) We also developed a new ranking technique to rank our graph vertices in which we used the harmonic mean to combine the results of two of the most important graph-based ranking methods which are: PageRank algorithm [5] and HITS algorithm [7]. (3) In addition, we have built a straightforward approach that extracts the summaries through simple steps that do not require complex linguistic processing or labeled training data.

To evaluate our proposed approach, we used the DUC 2003 & DUC 2004 benchmark dataset, and we measured its performance using the ROUGE evaluation toolkit [9]. Our experiments showed that using the harmonic mean in combining weighting schemes outperform the arithmetic mean and show a good improvement over the two baselines and many state-of-the-art systems. Nevertheless, the results showed that our proposal of taking the average ranking scores using the harmonic mean obtained comparable results to the PageRank and did not give the desired improvement.

In the future we plan to increase the number of the participated weighting schemes and ranking methods, then investigate their role in the combination process. Also, we plan to experiment with some more advanced methods for combining the scores, like using machine learning techniques to learn what is the best score that can be used among all proposed scores.

REFERENCES

[1] A. Nenkova and K. McKeown, "Automatic Summarization," *Found. Trends. Inf. Retr.*, vol. 5, no. 2, pp. 103–233, 2011.

[2] I. Mani, *Automatic Summarization*. Amsterdam, The Netherlands: John Benjamins Publishing Company, 2001.

[3] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," *J. Artif. Intell. Res.*, vol. 22, pp. 457–479, Dec. 2004.

[4] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2004, pp. 404–411.

[5] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Comput. Netw. ISDN Syst.*, vol. 30, nos. 1–7, pp. 107–117, Apr. 1998. doi: 10.1016/S0169-7552(98)00110-X.

[6] R. Mihalcea, "Graph-based ranking algorithms for sentence extraction, applied to text summarization," in *Proc. Interact. Poster Demonstration Sessions*, 2004, p. 20.

[7] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM* , vol. 46, no. 5, pp. 604–632, 1999.

[8] P. Herings, G. van der Laan, and D. Talman, "Measuring the power of nodes in digraphs," *SSRN Electron. J.*, Nov. 2001. doi: 10.2139/ssrn.288088.

[9] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using N-gram co-occurrence statistics," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics Human Lang. Technol.*, vol. 1. 2003, pp. 71–78.

[10] Y. Chali and S. Joty, "Improving the performance of the random walk model for answering complex questions," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2008, pp. 9–12.

[11] X. Wan and J. Yang, "Improved affinity graph based multi-document summarization," in *Proc. Hum. Lang. Technol. Conf.*, 2006, pp. 181–184.

[12] E. Baralis, L. Cagliero, N. Mahoto, and A. Fiori, "GraphSum: Discovering correlations among multiple terms for graph-based summarization," *Inf. Sci.*, vol. 249, pp. 96–109, Nov. 2013.

[13] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2011, pp. 510–520.

[14] D. Parveen, H. Ramsl, and M. Strube, "Topical coherence for graph-based extractive summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1949–1954.

[15] W. Wang, F. Wei, W. Li, and S. Li, "HyperSum: Hypergraph based semi-supervised sentence ranking for query-oriented summarization," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, Nov. 2009, pp. 1855–1858.

[16] A. Bellaachia and M. Al-Dhelaan, "Multi-document hyperedge-based ranking for text summarization," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, Nov. 2014, pp. 1919–1922.

[17] C. Lin and E. Hovy, "The automated acquisition of topic signatures for text summarization," in *Proc. Int. Conf. Comput. Linguistic*, 2000, pp. 495–501.

[18] X. Wan and J. Yang, "Multi-document summarization using cluster-based link analysis," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2008, pp. 299–306.

[19] M. Mendoza, S. Bonilla, C. Noguera, C. Cobos, and E. León, "Extractive single-document summarization based on genetic operators and guided local search," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4158–4169, Jul. 2014.

[20] P. Mehta, "From extractive to abstractive summarization: A journey," in *Proc. ACL Student Res. Workshop*, 2016, pp. 100–106.

[21] R. Ferreira, L. de Souza Cabral, F. Freitas, R. D. Lins, G. de França Silva, S. J. Simske, and L. Favaro, "A multi-document summarization system based on statistics and linguistic treatment," *Expert Syst. Appl.*, vol. 41, no. 13, pp. 5780–5787, 2014.

[22] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proc. 18th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1995, pp. 68–73.

[23] X. Wan, "An exploration of document impact on graph-based multi-document summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2008, pp. 755–762.

[24] T. C. Hoad and J. Zobel, "Methods for identifying versioned and plagiarized documents," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 54, no. 3, pp. 203–215, 2003.

[25] J. Leskovec, N. Milic-frayling, and M. Grobelnik, "Impact of linguistic analysis on the semantic graph coverage and learning of document extracts," in *Proc. AAAI*, 2005, pp. 1069–1074.

[26] J. Conroy and D. O'leary, "Text summarization via hidden Markov models," in *Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2001, pp. 406–407.

[27] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. Radev, "Graph-based neural multi-document summarization," 2017, *arXiv:1706.06681*. [Online]. Available: https://arxiv.org/abs/1706.06681

[28] M. A. Fattah, "A hybrid machine learning model for multi-document summarization," *Appl. Intell.*, vol. 40, no. 4, pp. 592–600, Jun. 2014.

[29] D. Parveen and M. Strube, "Integrating importance, non-redundancy and coherence in graph-based extractive summarization," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 1–9.

[30] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," 2016, *arXiv:1603.07252*. [Online]. Available: https://arxiv.org/abs/1603.07252

[31] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–12.

[32] J. L. Neto, A. D. Santos, C. A. Kaestner, and A. A. Freitas, "Document clustering and text summarization," in *Proc. 4th Int. Conf. Practical Appl. Knowl. Discovery Data Mining*, 2000, pp. 41–55.

[33] V. Hatzivassiloglou, J. L. Klavans, M. L. Holcombe, R. Barzilay, M. Y. Kan, and K. McKeown, "Simfinder: A flexible clustering tool for summarization," in *Proc. Workshop Summarization*, 2001, pp. 1–20.

[34] L. Yang, X. Cai, Y. Zhang, and P. Shi, ''Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization,'' *Inf. Sci.*, vol. 260, pp. 37–50, Mar. 2014.

[35] C. Li, X. Qian, and Y. Liu, ''Using supervised bigram-based ILP for extractive summarization,'' in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2013, pp. 1004–1013.

[36] P. Achananuparp, H. Xiaohua, and S. Xiajiong, ''The evaluation of sentence similarity measures,'' in *Proc. Int. Conf. Data Warehousing Knowl. Discovery*, 2008, pp. 305–316.

[37] K. Hong, M. Marcus, and A. Nenkova, ''System combination for multi-document summarization,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 107–117.

[38] P. Li, Z. Wang, W. Lam, Z. Ren, and L. Bing, ''Salience estimation via variational auto-encoders for multi-document summarization,'' in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–9.

[39] Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou, ''Ranking with recursive neural networks and its application to multi-document summarization,'' in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 1–9.

[40] P. Ren, Z. Chen, Z. Ren, F. Wei, L. Nie, J. Ma, and M. De Rijke, ''Sentence relations for extractive summarization with deep neural networks,'' *ACM Trans. Inf. Syst.*, vol. 36, no. 4, p. 39, 2018.

[41] Z. Cao, F. Wei, S. Li, W. Li, M. Zhou, and W. A. N. G. Houfeng, ''Learning summary prior representation for extractive summarization,'' in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, Jul. 2015, pp. 829–833.

[42] P. Ren, F. Wei, Z. Chen, J. Ma, and M. Zhou, ''A redundancy-aware sentence regression framework for extractive summarization,'' in *Proc. 26th Int. Conf. Comput. Linguistics Tech. Papers*, 2016, pp. 33–43.

[43] Z. Cao, W. Li, S. Li, and F. Wei, ''Retrieve, Rerank and rewrite: Soft template based neural summarization,'' in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 152–161.

[44] K. Hong and A. Nenkova, ''Improving the estimation of word importance for news multi-document summarization,'' in *Proc. 14th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2014, pp. 712–721.

ABEER ALZUHAIR received the B.S. degree in computer science from Imam Mohammed Ibn Saud University, Riyadh, Saudi Arabia, in 2011, and the M.S. degree in computer science from King Saud University, Riyadh, Saudi Arabia, in 2019.

MOHAMMED AL-DHELAAN received the M.S. and Ph.D. degrees in computer science from The George Washington University, USA, in 2009 and 2014, respectively. He then moved on to become an Assistant Professor with the Computer Science Department, King Saud University, Saudi Arabia. He has supervised M.S. students and taught several courses at both the bachelor's and graduate levels. His research interests include natural language processing and data mining, specifically he is focused on graph-based ranking, extracting keyphrases, statistical topic models, and text summarization. In addition, he has participated as a PC Member at many conferences, where he reviewed several research articles.