IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 15, 2020, accepted July 15, 2020, date of publication July 28, 2020, date of current version August 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012539

Extractive Document Summarization Based on Dynamic Feature Space Mapping

SAMIRA GHODRATNAMA[1], AMIN BEHESHTI[1], MEHRDAD ZAKERSHAHRAK[2], AND FARIBORZ SOBHANMANESH[1]

[1]Department of Computing, Macquarie University, Sydney, NSW 2113, Australia
[2]ASU School of Computing, Informatics, and Decision Systems Engineering (CIDSE), Arizona State University, Tempe, AZ 85281, USA

Corresponding author: Samira Ghodratnama (samira.ghodratnama@mq.edu.au)

This work was supported by the AI-Enabled Processes (AIP) Research Centre.

ABSTRACT The exponential growth of the Web documents has constituted the need for automatic document summarization. In this context, extractive document summarization, i.e., that task of extracting the most relevant information, removing redundancy and presenting the remained data in a coherent and cohesive structure, is a challenging task. In this article, we propose a novel intelligent approach, namely ExDoS, that harvests benefits of both supervised and unsupervised algorithms simultaneously. To the best of our knowledge, ExDoS is the first approach to combine both supervised and unsupervised algorithms in a single framework and an interpretable manner for document summarization purpose. ExDoS iteratively minimizes the error rate of the classifier in each cluster with the help of dynamic local feature weighting. Moreover, this approach specifies the contribution of features to discriminate each class, which is a challenging issue in the summarization task. Therefore, in addition to summarizing text, ExDoS is also able to measure the importance of each feature in the summarization process. We evaluate our model both automatically (in terms of ROUGE factor) and empirically (human analysis) on the benchmark datasets: the DUC2002 and CNN/DailyMail. Results show that our model obtains higher ROUGE scores comparing to most state-of-the-art models. The human evaluation also demonstrates that our model is capable of generating informative and readable summaries.

INDEX TERMS Automatic text summarization, extractive summarization, feature weighting, multi-document summarization.

I. INTRODUCTION

The expansion of Internet and Web applications, followed by the growing influence of smart-phones on every aspect of our lives, induced an everyday growth of textual information. This eruption in data generation not only makes it impossible for humans to summarize, but also even machines are unable to process this massive amount of generated data that is producing in different technologies, applications, and organizations. Analyzing large amounts of data that are mostly unstructured is quite challenging, if not impossible. This explosion of documents necessitates automated document summarization, which aims to create a shorter version of documents while preserving their essential information. Therefore, for any decision making and gaining insights from data, a summarization technique is essential. Hence, text summarization is becoming one of the most useful techniques in today's fast-growing world.

Summarization can help to gain insights from data and facilitate decision making. For instance, social media applications such as Twitter[1] and Facebook[2] are now being utilized for personal purposes as well as for marketing and political purposes. Nowadays, the majority of political campaigns all over the world use social media as their main apparatus to reach out to their supporters [1]. Thus, extracting textual information can be vital to successful marketing and political strategies. The real-world applications of automatic text summarization are not limited to marketing or political campaigns. For instance, text summarization is also employed for compressed descriptions of the search results in search engines as well as in keyword directed subscriptions to

The associate editor coordinating the review of this manuscript and approving it for publication was Hocine Cherifi.

[1]https://twitter.com/
[2]https://www.facebook.com/

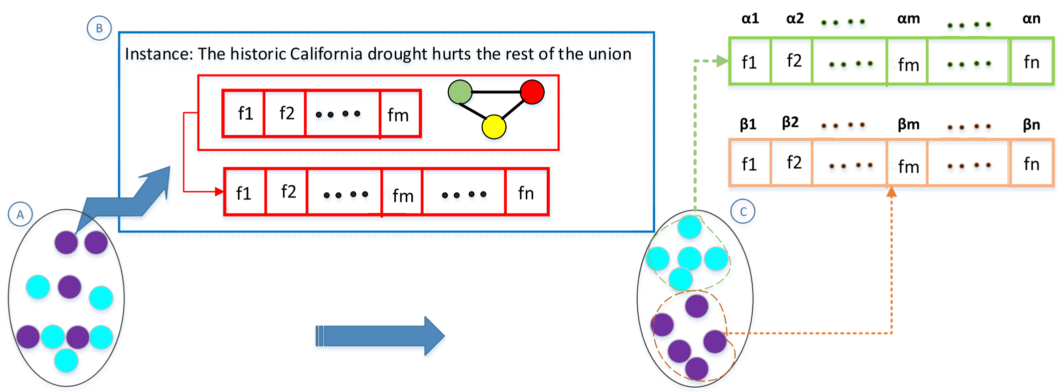

FIGURE 1. An overview of the proposed approach (ExDoS): (A) A simple dataset with two classes, (B) Each instance is a sentence. We combine both surface and linguistic features (extracted from the semantic graph) to make a unified feature set, and (C) The final output is groups of similar samples where features are locally weighted in each group.

news [2]. Furthermore, an effective text summarization of social media content can maintain the trust of users, based on user navigation between different contents [3].

Although text summarization is an old challenge and the first attempts return back to the 1950s, where they used features such as word and phrase frequency to extract important sentences, it still is a demand research field due to its applicability. A good summary should keep the main content while helping users to understand large volumes of information quickly. The first definition of text summarization is to extract the most informative parts as a compressed version for a particular user or task. A summary is also defined as a shorter version of a document that is generated by a machine to draw the most significant information in a shorter form and without any human assistance [4]. However, a more recent definition is given by Radev *et al.* [5] as "a text that is produced from one or more documents, which conveys the important information in the original text, and usually significantly less than that". According to this definition, three important issues should be considered: (i) summaries can be produced from single or multiple documents; (ii) summaries should preserve the important parts of the original text; and (iii) summaries have to reduce the original text by at least 50%.

There exist various categorizations for document summarization. However, automatic summarization models are mainly divided into two categories: extractive and abstractive. Abstractive approaches understand the text profoundly and then represent the text in another shorter structure. On the other hand, the goal of extractive methods is to select the most informative units. Since it is quite hard for the machine to produce a summary that is smooth and understandable by humans, in practice, extractive approaches are mostly used.

Among different categories for extractive summarization approaches, recently, machine learning approaches have been widely used in the extractive summarization problem. Extractive summarization can be either unsupervised or supervised. In unsupervised approaches, the goal is to find representative sentences while in supervised methods, the problem is considered as a binary classification task where classes defined as being/not being included in the summary [6], [7]. In this article, we present a novel intelligent approach for summarization, called ExDoS, which benefits from using both supervised and unsupervised algorithms simultaneously and in an interpretable manner. ExDoS has the following novel characteristics:

- We combine the clustering and classification algorithm into a single objective function. Clustering aims to discover the underlying structure of data and then feeds this information to the classification stage through a single objective function. Hence, it improves the performance of the summarization algorithm.
- Features are dynamically weighted through the optimization process in each cluster. These weights represent the role of each feature in discriminating each label individually while summarizing documents.
- Sentences are selected in a way that produced summaries are coherent and non-redundant. The most crucial sentence will be at the top, and then other sentences are chosen in a way that covers all critical information while not being redundant.

The proposed approach obviates the need for feature engineering in the summarization task. Although the most critical phase in machine learning algorithms is feature extraction, prior work mostly focused on the sentence selection process. Recently, some attempts have been made to find an optimum feature set for the summarization process. These approaches consider the relevance of each feature as a binary problem, i.e., whether a feature is included in the feature set or not [8]. The empirical results show that ExDoS can efficiently capture the important pattern of features. An overview of ExDoS is illustrated in Figure 1, where a sample is a sentence modeled as a vector of features. The final output is groups of similar samples where features are locally weighted in each group. The weights of features illustrate the importance of each feature in sub-spaces (clusters). In the summarization problem, these spaces represent being or not being in summary. Our main contributions can be summarized as follows:

- We introduce and formalize a theoretically grounded method, based on the idea of combining supervised and unsupervised learning. We employ this novel idea in the task of document summarization. This architecture allows us to develop clusters of sentences that can help to select summaries. Specifically, we designed a ranking measure that determines whether a document sentence matches a highlight and should be labeled with "1" (must be in summary) or "0" otherwise.
- ExDoS is capable of measuring the role of features in discriminating each class *individually* through the summarization process by making various feature spaces.
- We provide evidence in the form of experiments in which the model is trained to illustrate that ExDoS obtains state-of-the-art performance while the importance of all features for discriminating different classes on different datasets is reported.
- We perform a human evaluation experiment to assess summaries from the viewpoint of the human experts as ground truth. This proves that summaries created by ExDoS are less redundant and more informative than summaries created by competing approaches.
- In addition to being a state-of-the-art performer, ExDoS has an additional advantage, which is being very interpretable. The clearly separated terms in the optimization process allow us to track the output summary. Such visualization is especially useful to explain decisions made by the system to the end-user.

II. RESEARCH OBJECTIVE

The problem of document summarization has been widely studied before due to its real-world applications. In this problem, the input is a set of documents where the output is a human-readable summary. In this article, we propose an extractive multi-document summarization. The novelty of this article is to propose a general framework that benefits from both supervised and unsupervised techniques in a single algorithm to reveal the hidden structure of data. Our rationale behind this is to harvest the advantages of both classification and clustering algorithms simultaneously. While classification uses the knowledge of labels, clustering extracts the hidden information based on features. Therefore, combining these two approaches can lead to many advantages in different problems [9], [10]. ExDos approach iteratively minimizes the error rate of the classifier in each cluster with the help of dynamic local feature weighting in one objective function. In addition to summarizing the documents, ExDoS can measure the importance of different features with the help of local feature weighting. The local weights of features indicate how each feature contributes to making each cluster. We provide a detailed technical description of the proposed summarization system throughout the article and illustrate its functionality using a working example. Besides, we evaluate our model both automatically (in terms of ROUGE factor) and empirically (human analysis) on benchmark datasets: the DUC2002 and CNN/DailyMail dataset. The accuracy of the approach, the importance of features, the effect of local feature weighting, the complexity of the method, and the parameter are all evaluated in this article.

III. RELATED WORK

Producing a summary is a complicated task even for a domain-expert person who has the knowledge of words and concepts, and yet it can be even more difficult for machines. The machine should have the ability of natural language processing and producing a human-understandable summary, in addition to the background knowledge. There exist different categories for document summarization. For instance, one is based on the goal of the summarization task, which includes generic, domain-based (topic-focused) [11], or query-based summarization algorithm [12].

We also have other categories for document summarization which is based on the application of summarization such as article summarization [13], review summarization [14], news summarization [15], and also summarization for anomaly detection [16].

One main categorization is considering the process and output type of the summarization algorithm, which are extractive and abstractive approaches. Abstractive summaries are generated by interpreting the main concepts of a document and then stating those contents in a clear natural language [6]. Abstraction techniques are a substitute for the original documents rather than a part of it. Therefore, abstractive approaches require deep natural language processing such as semantic representation, and inference. However, they are challenging to produce and yet have not arrived at a mature stage. Today's systems or computing devices cannot provide semantic representation, inference, and natural language to such a level that is equivalent to humans. Abstractive summarization approaches are classified into two groups: structure-based and semantic-based approaches [17]. Structure-based approaches aim to find a schema that can describe the document, which includes template-based methods, rule-based methods, ontology-based, and tree-based procedures. Figure 2 illustrates the main categories of document summarization.

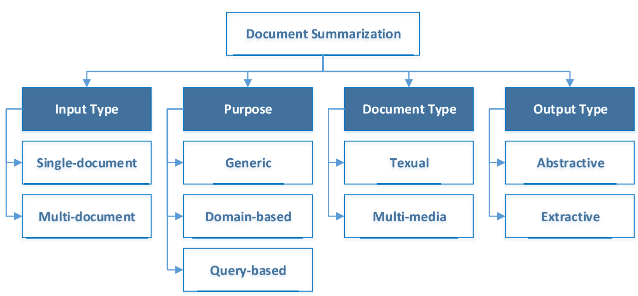

FIGURE 2. Categorization of document summarization techniques.

Extractive document summarization has been widely studied in the past. Since the proposed approach in this article is an extractive approach, we analyze the extractive methods in more detail. Extractive text summarization selects some sentences as the representative of the original documents. These sentences are then concatenated into a shorter text to produce a meaningful and coherent summary [18]. An extractive approach usually contains three steps: i) Representation of the original text document; ii) Sentence scoring; iii) Selecting high scores sentences in summary. Generally, based on these three steps, extractive approaches can be classified into three groups: surface or statistical methods, entity-level, and discourse-level methods. Surface-level approaches were the most popular methods since the 1950s, which process the documents based on shallow features such as frequency of words, sentence position, words in the title, or presence of cue phrases in the text. The entity-level methods focus on modeling semantic, syntactic, and logical relations between entities in the document. The relations between entities are based on similarity, proximity, and cohesion. Similarity refers to words in the text that have a similar stem (e.g., white and whiteness). In contrast, proximity refers to the distance between the text units in which the entity occurs. Furthermore, cohesion refers to the connection between relevant units of the text, which contain entities strongly connected into a semantic structure. Cohesion provides mechanical connections on the language level and ensures that the text makes sense as a whole. Connections between sections, sentences, and phrases are made with the grammatical and lexical links. Finally, the discourse-level methods refer to the modeling of the global text structure, which models the document's global structure and its relation to the communicative goals, taking into account the rhetorical structure of the text (argumentation and/or narrative structure) [2].

We also can divide the approaches based on the different representation models. The first category is the graph-based approaches, which are used for both single and multi-document summarization [4], [19]–[21]. In this category, each sentence is treated as a node. Two nodes or specifically, two sentences are connected with an edge if the two sentences have some similarities. The graph-based approaches depend on the sentence centrality and centroid [21]. Other approaches to find similarity measures between the nodes are discounting, cumulative sum method [19], and position weight [21].

The second category is clustering-based approaches that are suitable for both single document and multi-document summarization [19]. Similar documents and passages are clustered together so that the related information remains in the clusters. After clustering, sentences are ranked within each cluster, then their salience scores are calculated. Sentences with high scores in each cluster are extracted to make the summary.

The third category is the lexical chaining approaches. Lexical chains are basically defined as semantically related words spread over the entire document. Each chain of words shows the semantically related cluster of words [22], [23]. Words from the document are grouped into meaningful clusters to identify various themes within a document. Then, these clusters are arranged systematically to form a binary tree structure.

The fourth category is frequent-term approaches [24]. These methods check for the terms which are frequent and semantically similar. For the calculation of semantic similarity, they check the length of the path linking the terms, the position of the terms, the difference of information content, and the similarity between the terms. After this, summarizer filters the sentences having the most frequent, semantically related terms and extracts them for a summary.

The fifth category is the information retrieval approaches [19] that is very similar to the graph-based and lexical chaining approaches. These approaches are an enhancement of the two graphical methods, LexRank (threshold) and LexRank (continuous). In these methods, the main feature is logical closeness, i.e., how two sentences are logically related to each other rather than just the topical closeness. Besides, sentences must be coherent in a sense. Finally, more related sentences are picked up in a chain to produce a logical summary.

The sixth category is machine-learning approaches. Different machine learning approaches are used for this purpose, such as naive-bayes, decision trees, log-linear, Hidden Markov Model, neural networks, and reinforcement learning [4], [20]. Recently, the focus is mainly on neural network-based and deep reinforcement learning methods, which could demonstrate promising results. They employ word embedding [25] to represent words at the input level. Then, feed this information to the network to gain the output summary. These models mainly use a convolutional neural network [6], [26], a recurrent neural network [27], [28] or a combination of these two [29], [30]. Although these approaches could gain outstanding results in terms of performance, they are not efficient and interpretable. None of them could estimate the role of each feature in the summarization task and for each class separately. A few approaches predicted the *general* importance of features using various combinations of features [8], [31]. It is clear that as the number of features increases, these approaches are not practical. In other attempts, authors considered a summarization approach as a combinational optimization problem. Although, in these approaches, features are weighted, the goal is to select sentences that optimize the objective functions. Therefore, these weights aim to measure the importance of sentences, not features [32], [33]. The leading methods in each group are considered as baselines for evaluating ExDoS.

IV. THE PROPOSED APPROACH (ExDoS)

To tackle the summarization problem, previous approaches used supervised or unsupervised methods [34]. Although supervised approaches used to gain better results when data are labeled, nowadays, unsupervised methods could outperform supervised methods in some cases [35]. Besides, extracted features play a significant role in both categories. ExDoS considers all these properties to improve the accuracy of the summaries while measuring the function of each feature in the output summary.

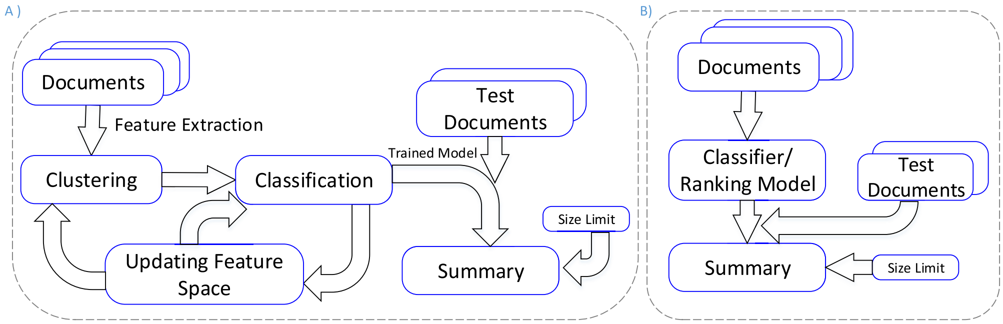

FIGURE 3. A) The architecture of ExDoS: the weights of features in each cluster are updated iteratively in a way that brings similar samples closer to each other in the new feature spaces by minimizing the error of classifiers in clusters. B) The architecture of state-of-the-art approaches.

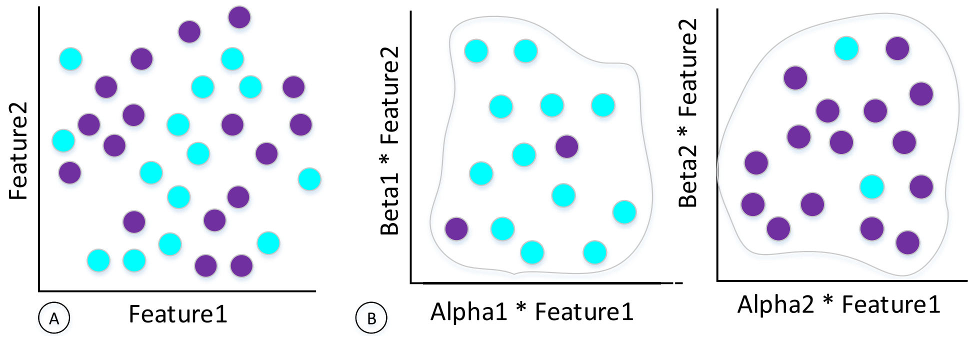

FIGURE 4. A)The distribution of synthetic data (2-class). B) The output of the ExDoS: two new feature spaces where the weights (Alpha and Beta) are updated in a way that similar samples are close to each other. The details of this transformation are illustrated in Figure 5.

ExDoS benefits from using both supervised and unsupervised approaches. A clustering algorithm is used to understand the underlying structure of data. Then, in an iterative process, the classification error in each cluster is minimized. To reach this goal, ExDoS transforms the feature space into a new feature space by weighting features locally in each cluster. This feature-weighting process aims to close up the same-label samples and push different-label samples further. Since the algorithm performs in an iterative process using gradient descent, the simplest clustering (k-means) and classification (KNN) algorithms are used to support efficiency. However, K-means is one of the most reliable and most widely used clustering algorithms. In addition, the K-nearest neighbor (NN) classifier has been successfully used in many pattern-recognition applications. It has been statistically proven that when $K = 1$ (1NN), the probable error of 1NN would be less than twice the Bayes classifier error. This proof states that 1NN is capable of generating near-optimal results. Figure 3 shows the architecture of ExDoS and how supervised and unsupervised approaches are combined to make new feature spaces. As from the picture can be seen, weights of features in each cluster are updated iteratively in a way that brings similar samples more close to each other in the new feature spaces by minimizing the error of classifier in clusters. To explain this process, a synthetic example of the ExDoS is illustrated in Figure 4 and 5. In Figure 4, the distribution of a synthetic two-dimensional data (2-class) is depicted. The output of the ExDoS is two new feature spaces where the weights (Alpha and Beta) are updated in a way that similar samples are close to each other. The details of this transformation and how the output is derived are illustrated step by step in Figure 5.

1) PROBLEM STATEMENT

In this section, we formally define the summarization tasks considered in this article. The input is a set of documents $\{D_1, D_2, \ldots, D_n\}$ while each document consists of a sequence of sentences $S = [s_1, s_2, \ldots, s_N]$. Each sentence $s_i \in R^d$ is a sample vector corresponding to the i-th sentence and d is the number of features. $Y = [Y_1, Y_2]$ is the class labels with two possible values of "1" (being in the summary) and "0" (not being in the summary). K is the number of clusters, and cluster centroids are denoted by C where C_k is the center of k-th cluster. The sample $s_=$ is the closest sample with the same class label, and the sample $s_{\neq}$ is the closest sample with a different class label, and d_w is the weighted Euclidean distance. Then, the goal is to learn a function $f : S \rightarrow Y$ which is defined on a given dataset $\{(s_1, y_1), (s_2, y_2), \ldots, (s_m, y_m)\}$.

2) FEATURE EXTRACTION

We explore a broad range of features that are commonly used for summarization. Two feature sets are defined to represent documents: *surface-level* and *linguistic-level*. The first sets

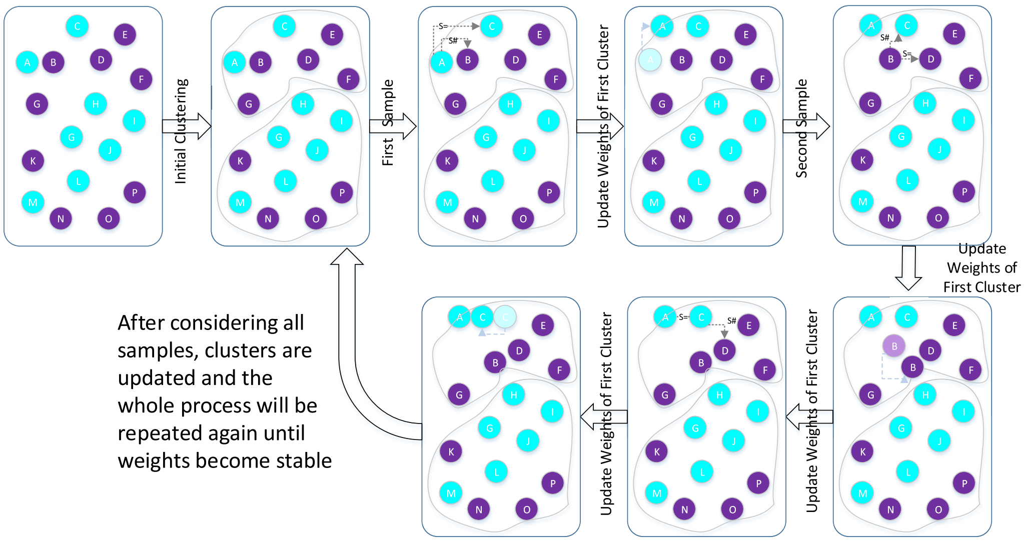

FIGURE 5. A single iteration of ExDoS: All samples are considered in each iteration, and weights are updated to bring the nearest same-label sample($s_=$) closer and push different-label samples ($s_{\neq}$) further. Then local weights of features are updated.

are extracted directly from the document. However, for the latter, the document is transformed into a semantic graph which is described below:

- **Surface Features**: It contains frequency-based features (TF-IDF, RIDF, gain and word co-occurrence), word-based features (uppercase word and signature words), similarity-based features (Word2Vec and Jaccard measure), sentence-level features (position, length cut off and length) and Named Entity. The detail structure of these features is presented in [8].
- **Linguistic Features**: It is based on the semantic graph as described below: *a) Semantic graph*: For each sentence, a parse tree is constructed using the NLP Stanford library tool [36] Each sentence is summarized in a sub-graph, which is a triple form. To make the triples, we used an algorithm that extracts triples in the form of subject, predicate, and object [37]. Sub-graphs are connected to each other, where edges are annotated with similarity weights. Similar or synonymous verbs (using Wordnet) are merged, and subjects are concatenated. Then weights update as the average weights of two merged sentences. *b) Linguistic Features*: Linguistic features are made of the average weights of connected edges, the merge status of a sentence as a binary feature, the number of sentences merged with a sentence, and finally, the number of sentences connected with a sentence.

3) THE PROPOSED METHODOLOGY

ExDoS aims to discover the underlying structure of the data in the clustering phase and then feed this information to the classification stage, in an iterative manner. Therefore, a continuous objective function is defined for analytically optimizing both clustering and classification stages by incorporating a new local feature weighting technique. The error rate of the nearest neighbor classifier is minimized using weighted distance, which overcomes the deficiency of popular Euclidean distance. Moreover, the captured space for decision making (in 1NN) by the Euclidean distance is a hyper-sphere. The overall objective function is defined as:

$$J(W, C) = J_1(W, C) + J_2(W) \tag{1}$$

where the first term (J_1) is the estimation error of clustering, and the second term (J_2) is the summation of the classification errors over the K clusters expanded below, and N_k is the number of samples in k-th cluster.

$$J(W, C) = \sum_{k=1}^{K}\sum_{i=1}^{|N_k|} d_w^2(s_i, C_k) + \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{|N_k|} S_\beta\left(\frac{d_w(s_i, s =)}{d_w(s_i, s_{\neq})}\right) \tag{2}$$

To estimate the error of 1NN, the following approximation function is used [38]:

$$\frac{1}{N}\sum_{s\in S} S_\beta\left(\frac{d_w(s, s_=)}{d_w(s, s_{\neq})}\right) \tag{3}$$

The sample $s_=$ is the nearest same-class sample, and the sample $s_{\neq}$ is the nearest different-class sample to the input sample s. Respectively d_w is the weighted Euclidean distance, and S_β is the Sigmoid function. Two parameters are optimized in this objective function. The feature-dependent weights associated with the sample s are trained to make the $s_=$ closer to s while making the sample $s_{\neq}$ further. Then, cluster centers update using the learned weighted distance. Since this function is differentiable, we can analytically use gradient descent, guaranteeing convergence, for estimating the matrix W as well as centers. The iterative optimization of learning parameters are given below, where α and γ are learning parameters:

$$W^{t+1} = W^t - \alpha(\frac{J(W,C)}{\delta(W)}) \tag{4}$$

$$C^{t+1} = C^t - \gamma(\frac{J(W,C)}{\delta(C)}) \tag{5}$$

To simplify the formula, the function $R(x)$ is defined as:

$$R(s_i) = (\frac{d_w(s_i, s_{i,=})}{d_w(s_i, s_{i,\neq})}) \tag{6}$$

The partial derivative of $J(W, C)$ with respect to W is calculated by:

$$\frac{\delta J(W,C)}{\delta W_k} \cong \sum_{i=1}^{|N_k|} 2W_k \odot (x_i - C_k)^2 + \frac{1}{N}\sum_{i=1}^{|N_k|} S'_\beta(R(s_i))\frac{\delta R(s_i)}{\delta W_k} \tag{7}$$

where $\odot$ is the inner product and $\frac{\delta R(x_i)}{\delta W_k}$ is:

$$\frac{\delta R(s_i)}{\delta W_k} = \frac{1}{d^2_{W_k}(s_i, s_{i,\neq})}(\frac{1}{R(s_i)}W_k \odot (xs_i - s_{i,=})^2 - R(s_i)W_k \odot (s_i - s_{i,\neq})^2) \tag{8}$$

The derivative of $S_\beta(z)$ is defined as:

$$S_\beta(z)' = \frac{\delta S_\beta(z)}{\delta z} = \frac{\beta e^{\beta(1-z)}}{(1+e^{\beta(1-z)})^2} \tag{9}$$

And the partial derivative of $J(W, C)$ with respect to C is calculated as:

$$\frac{J(W,C)}{\delta C_k} \cong \sum_{i=1}^{|N_k|} -2W_k^2 \odot (x_i - C_k) \tag{10}$$

Since we need to optimize the weight of features for cluster samples along with the center of clusters, we first update W in each cluster, and then we update centers (C).

A. GENERATING SUMMARY

Having the model trained, we define three measures, including coverage, coherence, and redundancy, to generate summaries:

1) COVERAGE

The coverage of sentences based on the proposed architecture is defined as:

$$Cov(s_i) = | d_w(c_+, s_i) - d_w(c_-, s_i) | \tag{11}$$

For each sentence, its weighted distance to cluster centers is estimated. The coverage is defined as the difference between data points and two cluster centers. c_+ is the cluster where the majority of samples belong to the positive class (being in summary) and (c_-) is the cluster where samples mostly belong to the negative class (not being in summary). This concept can be generalized when we have a multi-class problem. The coverage of a summary is the sum of coverage of all sentences in the summary.

2) COHERENCE

A critical aspect of a good summary is the coherence and the order of a summary. For this purpose, we use G-Flow [39][3] which is a graph model for selection and ordering that balance coverage and coherence. It relies on the approximate discourse graph (ADG) where each node is a sentence, and edges indicate whether a sentence follows the other in a coherent way. The indicators include co-reference, discourse cues, deverbal nouns, and more. The coherence is defined as the sum of the edge weights between successive summary sentences defined as below:

$$Coh(s_i) = w_{G+}(s_i, s_{i+1}) + w_{G-}(s_i, s_{i+1}) \tag{12}$$

where w_{G+} represents positive edges and w_{G-} represents negative edges. Since it considers the coherence between adjacent sentences, the produced summary may lack topic coherence compared to human-generated summaries. However, the outcome of the experiments does not indicate this problem. The coherence of a summary is the sum of the coherence of all sentences in summary.

3) REDUNDANCY

The redundancy measure of a sentence is defined as the combination of embedding similarity of a sentence with the previously selected sentences. The overall score of each sentence is defined as:

$$Red(s_i) = \sum_{s \in Summary} sim(s_i, s) \tag{13}$$

4) OBJECTIVE FUNCTION

We make our objective function to balance these criteria by having all coverage, coherence, and redundancy and the limit size (B).

$$\begin{aligned} &maximize : Score(S) \triangleq Cov(S) + \lambda Coh(S) - \phi Red(S) \\ &s.t. \sum_{s \in Summary} length(s) < B \end{aligned} \tag{14}$$

For solving this objective function, we need to use a local search to reach an approximation of the optimum value. For this purpose, we use a hill-climbing algorithm with a random start [40]. Adding, removing, or replacing a sentence is permitted in each step, and the parameters are trained in the process.

V. EXPERIMENTS AND EVALUATION

In this section, we present our experimental setup for assessing the performance of our summarization model. We discuss the dataset, give implementation details, and explain how system output was evaluated.

A. DATASET

To compare the performance of ExDoS with the existing leading approaches, experiments on two benchmark datasets,

[3]http://knowitall.cs.washington.edu/gflow/

DUC2002[4] and CNN/Daily Mail [41][5] are performed. We set the experimental setting similar to the state-of-the-art approaches [42]. In this setting, 75% of the data in DUC2002 is used for training, and the rest for the evaluation. In CNN/DailyMail, 286,722 documents are set for training and 11,480 for testing.

B. EVALUATION METHODS

1) AUTOMATIC EVALUATION

We evaluate the quality of summaries using ROUGE measure [43][6] defined below.

$$ROUGE_n = \frac{\sum_{S\in\{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \tag{15}$$

The three variants of ROUGE (ROUGE-1, ROUGE-2, and ROUGE-L) are used. ROUGE-1 and ROUGE-2 are used to evaluate informativeness, and ROUGE-L (longest common subsequence) is used to assess the fluency. We used the limited length ROUGE recall-only evaluation (75 words) for comparison of DUC to avoid being biased. Besides, the full-length F1 score is used for the evaluation of the CNN/DailyMail dataset. We used this measure to compare the produced summary with state-of-the-art approaches and to analyze the effect of local feature weighting in the same approach.

2) HUMAN EVALUATION

While ROUGE serves as a rough measure of coverage, it only compares the n-gram units. Therefore, we conducted a human experiment to evaluate the model based on other criteria such as informativeness, redundancy, and overall quality using twenty randomly samples DUC2002 test documents. Twenty-five participants attended the task, without any specific prior background, using Amazon Mechanical Turk. Participants were presented with a news article and summaries generated by different approaches. The output of these systems was shown to them, and they were asked to rank them based on the mentioned criteria.

C. BASELINES

We compared ExDoS to various previously published models known to their significant performance on the datasets. These approaches are briefly explained below:

- Two significant approaches among early approaches include *Leading sentences (Lead-3)*, which selects the three leading sentences as the summary and *Phrase-based ILP model* [44], which is based on a linear programming formulation that learns to combine phrases considering features such as coverage, length, and grammar constraints. Although these approaches rely on shallow features, these two approaches still report promising results.
- *TGRAPH* [45], and *URANK* [46] are significant approaches in graph-based approaches that use graph model and ranking scores for each sentence obtained in a unified ranking process.
- Among neural network models *NN-SE* [27] is a neural network model composed of a hierarchical document encoder and an attention-based extractor. *SummerRuNNer* [28] is recurrent neural network model (RNN). *HSSAS* [47] is a neural network model with a hierarchical structured self-attention mechanism to create the sentence and document embedding. *BANDITSUM* [42] is a neural network model that considers summarization as a contextual bandit (CB) problem. It receives a document and chooses a sequence of sentences to include in the summary. The policy here is to maximize the ROUGE score.

[4]Produced by the National Institute of Standards and Technology (https://duc.nist.gov/)

[5]https://github.com/abisee/cnn-dailymail

[6]We run ROUGE 1.5.5: http://www.berouge.com/Pages/defailt.aspx with parameters -n 2 -m -u -c 95 -r 1000 -f A -p 0.5 -t 0

D. RESULTS

We evaluated the proposed method from various perspectives, including automatic accuracy evaluation of the results, human preference evaluation, the effect of local feature weighting, parameter analysis, and efficiency analysis of the proposed approach, as described below. The initial number of clusters is set to the best value estimated by the silhouette approach [48].

TABLE 1. ROUGE score (%) comparison on DUC-2002 dataset.

Model	Rouge-1 Score	Rouge-2 Score	Rouge-L Score
Lead-3	43.6	21.0	40.2
ILP	45.4	21.3	40.3
TGRAPH [7]	48.1	24.3	N/A
URANK	48.5	21.5	N/A
NN-SE	47.4	23.0	43.5
SummaRuNNer	46.6	23.1	43.03
HSSAS	52.1	24.5	48.8
ExDoS	52.5	24.7	48.8

1) AUTOMATIC EVALUATION

The ROUGE results are illustrated in Table 1 and Table 2. According to Table 1 (DUC2002 dataset), ExDoS outperforms most state-of-the-art approaches, while competes with HSSAS. Results on CNN/Daily dataset follow the same trend as DUC2002. Note that the score is generally lower compared to DUC2002. This is due to the fact that the gold standards summaries are included paraphrasing. *HSSAS* [47] is a neural network model that has a hierarchical structured self-attention mechanism to create the sentence and document embedding; and *BANDITSUM* [42] is a neural network model that considers summarization as a contextual bandit (CB) problem. It receives a document and chooses a sequence of sentences to include in the summary where the policy is to maximize the ROUGE score. Our model is a simple, efficient model

[7]Rouge-L results for TGRAPH and URANK are not reported.

Topic: *How historic California drought affects rest of nation, often for the worse*	*Summary*
It's more than just one state's internal problem. The historic California drought hurts the rest of the union, too. That's because California is a breadbasket to the nation. California is growing more than a third of its vegetables and nearly two-thirds of its fruits and nuts. Here's why we should heed the ongoing drought in the most populous state. It is a slowly expanding natural disaster now in its fourth year that this week prompted Gov. Jerry Brown to announce a mandatory 25% cutback in water consumption in all cities. Prices rose last year for these items on your kitchen table: Broccoli by 11 cents per pound to $1.89.Grapes by 64 cents a pound to $3.06	A) The historic California drought hurts the rest of the union, too. B) The historic California drought hurts the rest of the union, too. California is growing more than a third of its vegetables and nearly two-thirds of its fruits and nuts. C) The historic California drought hurts the rest of the union, too. California is growing more than a third of its vegetables and nearly two-thirds of its fruits and nuts. Jerry Brown to announce a mandatory 25% cutback in water consumption in all cities.

FIGURE 6. Visualization of the summarization process for a CNN article about "California Drought"[8]. The left side is the original text, while the right one is the summarization process (three iterations). To visualize the sentence score, we divide the ranks of an iteration into four portions, each colored differently (darkest color shows the most important one). However, it should be noticed that ranks are changed in each iteration.

TABLE 2. ROUGE score (%) comparison on CNN/DailyMail using F1 variant of ROUGE.

Model	Rouge-1 Score	Rouge-2 Score	Rouge-L Score
LEAD-3	39.2	15.7	35.5
NN-SE	35.4	13.3	32.6
SummaRuNNer	39.9	16.3	35.1
HSSAS	42.3	17.8	37.6
BANDITSUM	41.5	18.7	37.6
ExDoS	42.1	18.9	37.7

that achieves better results in terms of ROUGE score in most cases as well as other benefits such as interpretability, which are discussed in the following. We performed an ANOVA test to evaluate the significant supremacy of our approach statistically. Results show ExDoS outperforms the baselines, including ILP, TGRAPH, URANK, NN-SE, with a significant margin ($p < 0.01$), while competing with HSSAS and BANDITSUM.

TABLE 3. Human evaluation result.

Model	Informativeness	Non-redundancy	Overall
LEAD-3	13%	21%	20%
SummaRuNNer	17%	19%	16%
HSSAS	20%	16%	21%
BANDITSUM	23%	22%	18%
ExDoS	27%	22%	25%

2) HUMAN EVALUATION

Human results reported in Table 3 represent the voting percentage of participants for each approach (ties are not allowed). ExDoS performs better than most state-of-the-art methods in all measures. However, in the redundancy aspect, ExDoS competes with BANDITSUM. Overall, ExDoS achieved significant performance. This is an interesting result showing that ExDoS performs well while using only the information of clustering without sophisticated constraint optimization (ILP, TGRAPH) or the complex architecture of the neural network (HSSAS and BANDITSUM).

[8]http://www.cnn.com/2015/04/03/us/california-drought/

TABLE 4. Estimation of features importance.

	Feature set	Freq based	Word based	Similarity based	Position based	Linguistic based
DUC	Sum	0.39	0.06	0.35	0.51	0.22
	NotSum	0.21	0.09	0.25	0.42	0.20
CNN	Sum	0.33	0.04	0.46	0.31	0.29
	NotSum	0.24	0.01	0.37	0.38	0.44

3) FEATURE IMPORTANCE

In addition to being state-of-the-art, ExDoS is also capable of learning the relevance of features, separately for each class, which are reported in Table 4. The reported weights are the average weights in each feature set. Based on observations, we concluded that in DUC2002, the position-based features play a major role in selecting summaries. Then frequency-based features for the class of *summary* and similarity feature for the class of *not summary* are the most important features. In CNN/DailyMail the similarity-based feature has a major impact on discriminating both classes, probably due to the paraphrased standard summaries.

TABLE 5. The effects of dynamic local feature weighting.

	DUC2002-ROUGE1	DUC2002-ROUGE2	CNN/Mail-ROUGE1	CNN/Mail-ROUGE2
ExDoS + Weighting	51.7	24.7	41.1	18.5
ExDoS - Weighting	43.3	20.1	38.7	14.3

4) EFFECT OF LOCAL FEATURE WEIGHTING

To evaluate the effect of local feature weighting, we conduct an ablation study to compare results to the global weighting of the same procedure, and the results are reported in Table 5. Table 5 shows that local feature weighting significantly affects the summarization result in both datasets.

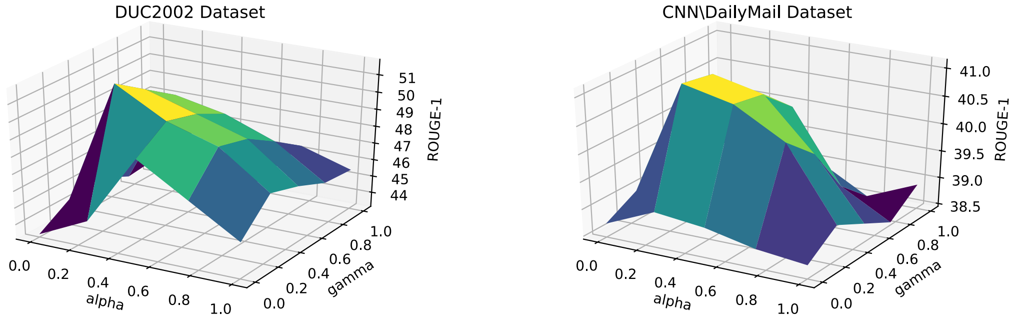

FIGURE 7. The image shows learning parameters(α, γ) and the corresponding ROUGE-1.

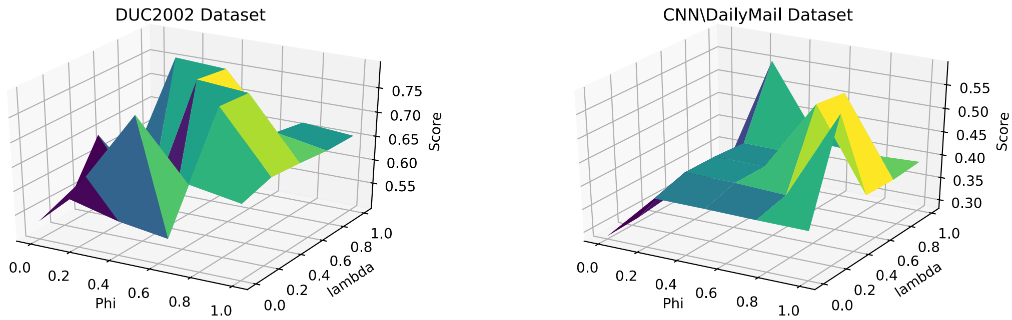

FIGURE 8. The image shows learning parameters(ϕ, λ) and the corresponding *score* value.

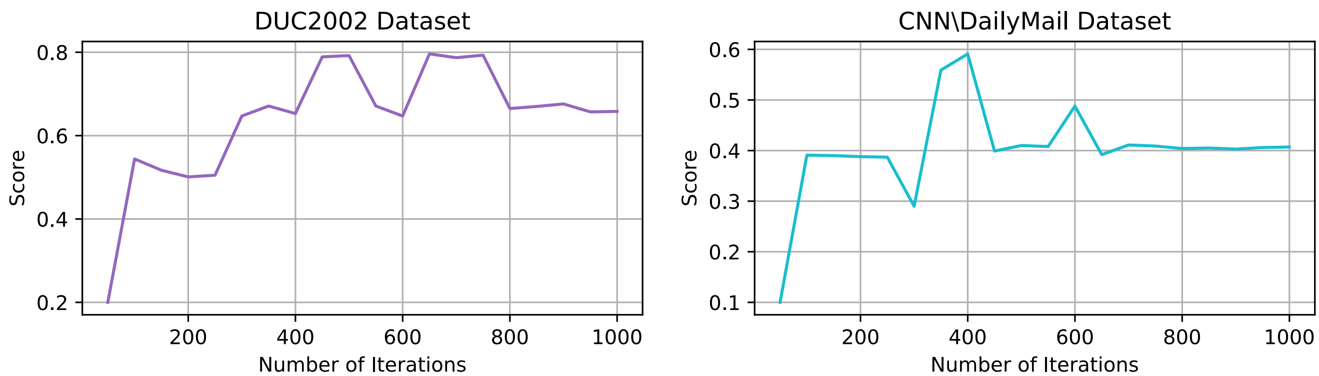

FIGURE 9. The image shows number of iterations versus score in both datasets.

5) EFFICIENCY

ExDoS is an efficient approach in terms of complexity. The computational complexity of ExDoS is determined as $O(K \times N_k \times I)$, where K is the number of clusters, N_k is the number of samples in the most populated cluster (Max $= N$), and I represents the maximum number of iterations where $I \ll N_K$. In Figure 9, the numbers of iterations versus *score* value are reported to illustrate the efficiency of ExDoS based on the number of iterations needed to converge the algorithm.

6) PARAMETER ANALYSIS

As in other parametric models, ExDoS has some hyper-parameters that are needed to be tuned. The learning-rate parameters of weights and centers (α, γ) control the speed of convergence in the gradient-descent algorithm. When the learning-rate is sufficiently small, the algorithm achieves linear convergence. A large value of the learning rate decreases the probability of converging to a suitable local optimum. β is another key hyper-parameter which regulates the slope of the Sigmoid function, where $S_\beta(R(s)) = 1/(1 + e^{(\beta(1-R(s)))})$. For small values of β, the Sigmoid derivative is almost constant. On the other hand, for large values of β, learning happens when the distance ratio $(R(s))$ is close to 1. Two other parameters are ϕ and λ, which control the coherency and redundancy in producing the summaries.

To find the best parameters, we have tested different combinations of learning rates (α, γ). These combinations and also the corresponding evaluation metric (ROUGE-1) are reported in Figure 7. It is noteworthy to say that the gradient-descent based learning schemes always converge to a local optimum. By running the algorithm, we empirically observed that it has a convincing convergence rate. The two other parameters (ϕ, λ) have also been tested by different values and reported in Figure 8. Since these two parameters control the coherency and redundancy, they do not have a significant effect on ROUGE. Therefore, the combination of these variables is reported in terms of *score* value.

VI. CONCLUSION AND FUTURE WORK

In this article, a general-purpose extractive approach for summarizing documents is proposed. The algorithm achieved better results than most state-of-the-art methods in terms of efficiency and performance. Besides, the post-trained weights represent the importance of each feature in discriminating against each class. To understand the role of local feature weighting and new feature spaces, we consider the performance of ExDoS by locally weighing and without weighting. Learning the importance of features is a start for summarization task. As our future work, we aim to use the knowledge of the crowd to update the weights in addition to the weights learned by the system. In this way, we consider the preference of users in making summaries implicitly.

REFERENCES

[1] C. Shirky, "The political power of social media: Technology, the public sphere, and political change," *Foreign Affairs*, pp. 28–41, Jan. 2011.
[2] M. Vlainić and N. M. Preradović, "A comparative study of automatic text summarization system performance," in *Proc. Eur. Comput. Conf. (ECC)*. Dubrovnik, Croatie, 2013, pp. 1–6.
[3] M. Z. Shahrak and S. Mohammadi, "Middle east user navigation in online social networks and interactions in e-commerce, an analogy," *Adv. Comput. Sci., Int. J.*, vol. 3, no. 2, pp. 32–36, 2014.
[4] D. Das and A. F. Martins, "A survey on automatic text summarization," *Literature Surv. Lang. Statist. II Course CMU*, vol. 4, nos. 192–195, p. 57, 2007.
[5] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," *Comput. Linguistics*, vol. 28, no. 4, pp. 399–408, Dec. 2002.
[6] Z. Li, Z. Peng, S. Tang, C. Zhang, and H. Ma, "Text summarization method based on double attention pointer network," *IEEE Access*, vol. 8, pp. 11279–11288, 2020.
[7] R. Rautray, R. Dash, and R. Dash, "Performance analysis of modified shuffled frog leaping algorithm for multi-document summarization problem," *Informatica*, vol. 43, no. 3, pp. 373–380, Sep. 2019.
[8] Y. K. Meena, P. Dewaliya, and D. Gopalani, "Optimal features set for extractive automatic text summarization," in *Proc. 5th Int. Conf. Adv. Comput. Commun. Technol.*, Feb. 2015, pp. 35–40.
[9] S. Ghodratnama and R. Boostani, "An efficient strategy to handle complex datasets having multimodal distribution," in *Proc. Interdiscipl. Symp. Complex Syst. (ISCS)*. Cham, Switzerland: Springer, 2015, pp. 153–163.
[10] S. Ghodratnama and H. A. Moghaddam, "Content-based image retrieval using feature weighting and C-means clustering in a multi-label classification framework," *Pattern Anal. Appl.*, pp. 1–10, Jun. 2020.
[11] J. Codina-Filbà, N. Bouayad-Agha, A. Burga, G. Casamayor, S. Mille, A. Müller, H. Saggion, and L. Wanner, "Using genre-specific features for patent summaries," *Inf. Process. Manage.*, vol. 53, no. 1, pp. 151–174, Jan. 2017.
[12] S. Xiong and D. Ji, "Query-focused multi-document summarization using hypergraph-based ranking," *Inf. Process. Manage.*, vol. 52, no. 4, pp. 670–681, Jul. 2016.
[13] H. Xu, Z. Wang, and X. Weng, "Scientific literature summarization using document structure and hierarchical attention model," *IEEE Access*, vol. 7, pp. 185290–185300, 2019.
[14] Y.-H. Hu, Y.-L. Chen, and H.-L. Chou, "Opinion mining from online hotel reviews—A text summarization approach," *Inf. Process. Manage.*, vol. 53, no. 2, pp. 436–449, Mar. 2017.
[15] P. Yang, W. Li, and G. Zhao, "Language model-driven topic clustering and summarization for news articles," *IEEE Access*, vol. 7, pp. 185506–185519, 2019.
[16] M. Ahmed, "Intelligent big data summarization for rare anomaly detection," *IEEE Access*, vol. 7, pp. 68669–68677, 2019.
[17] J.-U. Heu, I. Qasim, and D.-H. Lee, "FoDoSu: Multi-document summarization exploiting semantic analysis based on social folksonomy," *Inf. Process. Manage.*, vol. 51, no. 1, pp. 212–225, Jan. 2015.
[18] P. Mehta and P. Majumder, "Effective aggregation of various summarization techniques," *Inf. Process. Manage.*, vol. 54, no. 2, pp. 145–158, Mar. 2018.
[19] S. Hariharan and R. Srinivasan, "Studies on graph based approaches for singleand multi document summarizations," *Int. J. Comput. Theory Eng.*, vol. 1, pp. 1793–8201, Dec. 2009.
[20] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," *J. Emerg. Technol. Web Intell.*, vol. 2, no. 3, pp. 258–268, Aug. 2010.
[21] S. Hariharan, T. Ramkumar, and R. Srinivasan, "Enhanced graph based approach for multi-document summarization," *Int. Arab J. Inf. Technol.*, vol. 10, pp. 334–341, Jul. 2013.
[22] C. Santos, "Alexia-acquisition of lexical chains for text summarization," Ph.D. dissertation, Dept. Comput. Sci., Univ. Beira Interior, Covilhã, Portugal, 2006, p. 109.
[23] A. Hernandez-Castaneda, R. A. Garcia-Hernandez, Y. Ledeneva, and C. E. Millan-Hernandez, "Extractive automatic text summarization based on lexical-semantic keywords," *IEEE Access*, vol. 8, pp. 49896–49907, 2020.
[24] N. Kumar Nagwani and S. Verma, "A frequent term and semantic similarity based single document text summarization algorithm," *Int. J. Comput. Appl.*, vol. 17, no. 2, pp. 36–40, Mar. 2011.
[25] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[26] Z. Cao, F. Wei, S. Li, W. Li, M. Zhou, and H. Wang, "Learning summary prior representation for extractive summarization," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. (Short Papers)*, vol. 2, 2015, pp. 829–833.
[27] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2016, pp. 484–494.
[28] 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. 3075–3081.
[29] Y. Wu and B. Hu, "Learning to extract coherent summary via deep reinforcement learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, p. 5602.
[30] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., (Long Papers)*, vol. 1, 2018, pp. 1747–1759.
[31] R. Ferreira, L. de Souza Cabral, R. D. Lins, G. Pereira e Silva, F. Freitas, G. D. C. Cavalcanti, R. Lima, S. J. Simske, and L. Favaro, "Assessing sentence scoring techniques for extractive text summarization," *Expert Syst. Appl.*, vol. 40, no. 14, pp. 5755–5764, Oct. 2013.
[32] R. A. García-Hernández and Y. Ledeneva, "Single extractive text summarization based on a genetic algorithm," in *Proc. Mex. Conf. Pattern Recognit.* Berlin, Germany: Springer, 2013, pp. 374–383.
[33] 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.
[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] K.-F. Wong, M. Wu, and W. Li, "Extractive summarization using supervised and semi-supervised learning," in *Proc. 22nd Int. Conf. Comput. Linguistics COLING*, 2008, pp. 985–992.
[36] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2014, pp. 55–60.
[37] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenic, "Triplet extraction from sentences," in *Proc. 10th Int. Multiconf. Inf. Soc.-IS*, vol. 10, 2007, pp. 8–12.
[38] R. Paredes and E. Vidal, "Learning weighted metrics to minimize nearest-neighbor classification error," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1100–1110, Jul. 2006.
[39] M. J. Christensen, S. Soderland, and O. Etzioni, "Towards coherent multi-document summarization," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol. (NAACL)*, 2013, pp. 1163–1173.

[40] R. Stuart and P. Norvig, "Artificial intelligence: A modern approach," Upper Saddle River, NJ, USA, Tech. Rep., 2003.

[41] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1693–1701.

[42] Y. Dong, Y. Shen, E. Crawford, H. van Hoof, and J. C. K. Cheung, "BanditSum: Extractive summarization as a contextual bandit," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3739–3748.

[43] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, 2004.

[44] K. Woodsend and M. Lapata, "Automatic generation of story highlights," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 565–574.

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

[46] X. Wan, "Towards a unified approach to simultaneous single-document and multi-document summarizations," in *Proc. 23rd Int. Conf. Comput. linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 1137–1145.

[47] K. Al-Sabahi, Z. Zuping, and M. Nadher, "A hierarchical structured self-attentive model for extractive document summarization (HSSAS)," *IEEE Access*, vol. 6, pp. 24205–24212, 2018.

[48] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

SAMIRA GHODRATNAMA received the M.S. degree (Hons.) in computer science (artificial intelligence) from Shiraz University, Shiraz, Iran, in 2015. She is currently pursuing the Ph.D. degree in computer science with the Data Analytics Research Lab, Department of Computing, Macquarie University, Sydney, NSW, Australia. Her main research interests include machine learning and natural language processing, where she has published several articles.

AMIN BEHESHTI received the bachelor's and master's degrees (Hons.) in computer science and the Ph.D. degree in computer science and engineering from UNSW Sydney. He is currently the Director of the AI-Enabled Processes (AIP) Research Centre and the Head of the Data Analytics Research Lab, Department of Computing, Macquarie University. He is also a Senior Lecturer in data science with Macquarie University and an Adjunct Academic in computer science with UNSW Sydney. He is the leading author of the book entitled *Process Analytics*, coauthored with other high-profile researchers in UNSW and IBM research, recently published by Springer.

MEHRDAD ZAKERSHAHRAK received the M.S. degree in computer science from the University of Nebraska–Lincoln (UNL), Lincoln, NE, USA, in 2016, and the second M.S. degree in information technology from the K. N. Toosi University of Technology, Tehran, Iran. He is currently pursuing the Ph.D. degree in computer science with Arizona State University, Tempe, AZ, USA. He is a Research Data Scientist. He is also a Fulton Schools of Engineering Research Fellow with Arizona State University. His main research interests include machine learning and human–robot interaction specializing in reinforcement learning.

FARIBORZ SOBHANMANESH received the Ph.D. degree in electronics engineering from UNSW Sydney, Australia, in 2006. He is currently a Research Data Scientist with the AI-Enabled Processes (AIP) Research Centre, Department of Computing, Macquarie University, Sydney, NSW, Australia. His research area was initially on signal processing and speech recognition techniques when he was an Assistant Professor with Shiraz University, Iran, but gradually shifted toward machine learning and data science techniques in knowledge discovery of big data sets, including the genomic datasets, when he was working as a Research Scientist at CSIRO, Australia, business datasets and text mining techniques. He has published his research findings in high-reputation journals and conferences.