IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 14, 2020, accepted March 1, 2020, date of publication March 11, 2020, date of current version March 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2980226

Extractive Automatic Text Summarization Based on Lexical-Semantic Keywords

ÁNGEL HERNÁNDEZ-CASTAÑEDA, RENÉ ARNULFO GARCÍA-HERNÁNDEZ, YULIA LEDENEVA, AND CHRISTIAN EDUARDO MILLÁN-HERNÁNDEZ

Cátedras Conacyt, Autonomous University of Mexico State, Toluca 50000, Mexico

Corresponding author: René Arnulfo García-Hernández (renearnulfo@hotmail.com) and Yulia Ledeneva (yledeneva@yahoo.com)

This work was supported in part by the Consejo Nacional de Ciencia y Tecnología (CONACyT) under the cátedras program, and in part by the Programa para el Desarrollo Profesional Docente para el Tipo Superior (PRODEP).

ABSTRACT The automatic text summarization (ATS) task consists in automatically synthesizing a document to provide a condensed version of it. Creating a summary requires not only selecting the main topics of the sentences but also identifying the key relationships between these topics. Related works rank text units (mainly sentences) to select those that could form the summary. However, the resulting summaries may not include all the topics covered in the source text because important information may have been discarded. In addition, the semantic structure of documents has been barely explored in this field. Thus, this study proposes a new method for the ATS task that takes advantage of semantic information to improve keyword detection. This proposed method increases not only the coverage by clustering the sentences to identify the main topics in the source document but also the precision by detecting the keywords in the clusters. The experimental results of this work indicate that the proposed method outperformed previous methods with a standard collection.

INDEX TERMS Automatic text summarization, cluster validation indexes, genetic algorithm, extractive summaries, topic modelling.

I. INTRODUCTION

In recent years, a large amount of data are increasingly being stored digitally, enabling them to be accessed by a computer for analysis and interpretation. However, manually synthesizing the data through human efforts is an expensive task when the number of documents is considerably high. Therefore, various computerized methods have been proposed to automatically synthesize documents to provide the user with a summarized version. Simply stated, the automatic text summarization (ATS) task automatically selects the key ideas in a text to allow the reader to understand the target document.

In general, the ATS task attempts to synthesize a document by selecting (1) the main topics that make up the documents and (2) the relevant ideas of these topics. Therefore, existing methods try to improve their performance in identifying the key data in a document by considering all the themes found in it.

The main problem encountered by the ATS task is generalization; for example, summarizing a news story is a significantly different task from summarizing financial or medical reports. For this reason, many proposed methods have been applied to various specific problems of a specific domain.

For example, in the work of Hassan and Hill [1], automatic summarization techniques were used to provide comments for programming language statements. Their method establishes the basic concepts of a system that helps in understanding large codes, generally not commented on, written by other programmers. Thus, programmers can evaluate codes in a less time-consuming manner.

Another application of the ATS task was carried out by Cardinaels *et al.* [2]. The authors reported that humans tend to express personal interests in financial summaries. Therefore, they generated computer-based summaries of earnings releases, mimicking the human tendency to avoid including generally irrelevant or less objective information.

Various specific applications use ATS methods as the main mechanism to facilitate the analysis of large amounts of information; therefore, increasing the efficiency of these methods is crucial to improve the final applications.

There are two general techniques to automatically generate summaries: extractive and abstractive. Extractive techniques are based on a superficial analysis of the text that considers

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

only the syntactic level, where the output summary includes text units from the original text such as words, sentence segments, or complete sentences. In contrast, abstractive techniques perform a deeper analysis; for instance, they incorporate a semantic analysis, where the output summary may include new units not contained within the original text. Thus, the risk involved in abstractive summaries is that sentences may be reformulated with a different interpretation from that of the original author.

This study proposes an approach for automatic extractive text summarization (EATS) tasks. It is based on a clustering scheme supported by a genetic algorithm (GA) to identify the main topics in the document. Furthermore, the proposed method includes a topic modeling algorithm (latent Dirichlet allocation (LDA)) to determine the key sentences in clusters on the basis of the automatically generated keywords.

The clustering scheme requires a vectorial space, and therefore, different feature-generation methods have been proposed for mapping texts to numeric vectors: LDA and Doc2Vec [3]. In addition, conventional methods that have delivered good results according to the current standards, such as term frequency–inverse document frequency (TF–IDF) and *n*-grams, have been evaluated.

The goal of this study was to design an approach that can automatically produce summaries that are as close as possible to human-generated ones. Therefore, the challenging DUC02 dataset was selected to measure the effectiveness of the proposed approach. Moreover, this dataset includes human-generated summaries that could be used to compare the capability of the proposed EATS algorithm with human skills. In addition, the experimental results confirmed that our proposed approach can be applied in a multi-domain and multi-language framework by evaluating the TAC11 dataset.

Our experimental results indicate that our system outperformed previous methods owing to the two general steps that it applies: clustering, which helps in increasing the coverage by identifying the main topics in the source document, and the addition of semantic information to the model, which facilitates the detection of the key sentences in the clusters and improves precision.

The main contributions of this work are as follows:

- A language-independent system for EATS
- A domain-independent system for EATS
- An approach to identify key sentences that does not require prior information
- An EATS system that improves the detection of key sentences through an evolutionary and clustering approach
- The extraction of latent semantic information to locate keywords

The rest of the paper is organized as follows. In Section II, some approaches for the ATS task are discussed. In Section III, the basic concept applied in this study is explained in detail. The proposed approach, for the EATS task, is described in Section IV, and the experimental results are presented in Section V. Finally, the conclusions are drawn in Section VI.

II. RELATED WORKS

The general process of the EATS task involves the identification of relevant information from the text to build a new summarized document. Various strategies to automatically generate summaries, and thus allow the efficient processing of large numbers of documents, have been developed.

According to Gambhir and Gupta [4], depending on the linguistic level, ATS techniques can be classified as either extractive or abstractive.

Most research studies on EATS were focused on extractive summaries. For instance, they considered key sentences and their positions in the text [5], measured word frequencies [6], or assigned importance levels to the sentences [7].

At the lexical level, *n*-grams are frequently used to generate text models. For instance, in Ledeneva's method [8], the sequences of *n*-grams are extracted from the text by using a model of maximal frequent sequences. In contrast, Bando *et al.* [9] used *n*-grams to build paragraphs using the most representative terms in the document.

The features extracted from documents are evaluated by supervised and unsupervised methods to create models that allow the main components of the key ideas to be detected.

Supervised approaches have been widely explored [10], [11] to generate extractive and abstractive summaries. In Belkebir and Guessoum's method [12], each sentence in a document is labeled as ''1'' if it belongs to a summary, and the remaining sentences are labeled as ''0''. Then, the authors generated a variety of features, for instance, sentence position, sentence length, and similarity to title, by applying statistics- and linguistic-oriented procedures. The sentences are classified by using the AdaBoost algorithm.

Fattah and Ren [13] proposed a method that is similar to that of Belkebir and Guessoum [12] in that a summarizer that can be trained by using a variety of extracted features is applied. However, their method differs from that of Belkebir and Guessoum in that the relevance of a feature is considered by assigning a weight to it. This assignment is provided by a GA [14] and a regression model [15]. These models obtain an appropriate set of weights by processing 50 manually summarized English language documents.

The main problem with supervised approaches is that they require a set of labeled data. In addition, the domain of the training samples is often not sufficiently general for processing new multi-domain samples.

Recently, unsupervised machine-learning approaches have been utilized by applying clustering algorithms [16] to group sentences on the basis of the structure and frequency of the words. The most representative sentences of the formed groups are used to generate the summary.

In clustering approaches, to guarantee good-quality summaries, one needs to evaluate the groups of sentences. Two validation methods exist for evaluating the quality of the partitions: internal and external measures [17]. The former do not consider external information of the dataset classes, whereas the latter require class labels to be applied. Various authors have compared internal and external quality measures

for clustering validation. They attempted to experimentally determine which approach among the two approaches can evaluate the optimal groups formed from a dataset. Several quality measures have been tested on the basis of the groups built by the clustering algorithms. The results confirmed that internal measures outperform external measures by generating the best configurations of the groups.

In most studies focusing on unsupervised approaches, external quality measures such as the F-measure were used to validate the model performance; in contrast, internal quality measures such as cluster validation indexes have been rarely explored in the EATS task.

In their study, Soto *et al.* [18] developed an automatic summarization system that uses unsupervised learning. The authors used three text models to build numeric vectors: bag-of-words, *n*-grams, and maximal frequent sequences. They grouped the resultant vectors by using a *K*-means algorithm, and the final clusters were evaluated by using an external measure (F-score). Their experimental results indicated that the maximal frequent sequences provide relevant information to the model to improve its performance.

In general, the goal of the EATS task is to separate the key ideas in documents from those that are secondary. Previously proposed methods consider only the external factors of the documents, such as the sentence length or position; however, they do not consider the structure of the document. Therefore, in this study, an evolutionary clustering scheme based on a generative model (LDA) and on a context-based model (Doc2vec) that provides substantial information of the latent semantic links among words is proposed.

III. METHODOLOGY

In this section, we describe the basic concepts of the proposed method. The general steps of the proposed approach where the methodology is applied are presented in Figure 1.

Following this flowchart, the first step to begin the proposed summarization process is the conversion of the texts to numerical vectors by applying different methods to generate the features (discussed in detail in Section III-A); therefore, given that the basic unit selected is the sentence, each sentence in a document is represented by a numerical vector.

These vectors are then grouped according to the proximity measure (see Section III-B), and the quality of the clustering is evaluated using the Silhouette index (see Section III-C), which, together with a GA, helps in selecting the best approximate number of clusters (process discussed in detail in Section IV-B).

The above steps provide a clustering representation (described in Section IV-A) where the key sentences are selected from each cluster formed as follows: an LDA model is used to obtain the word distribution in the document to be summarized. This distribution establishes a link between the word and the probability of its occurrence in the document. Therefore, the inference is that the words with a high probability of occurrence are very relevant words in the document; therefore, such probabilities help in ranking the sentences

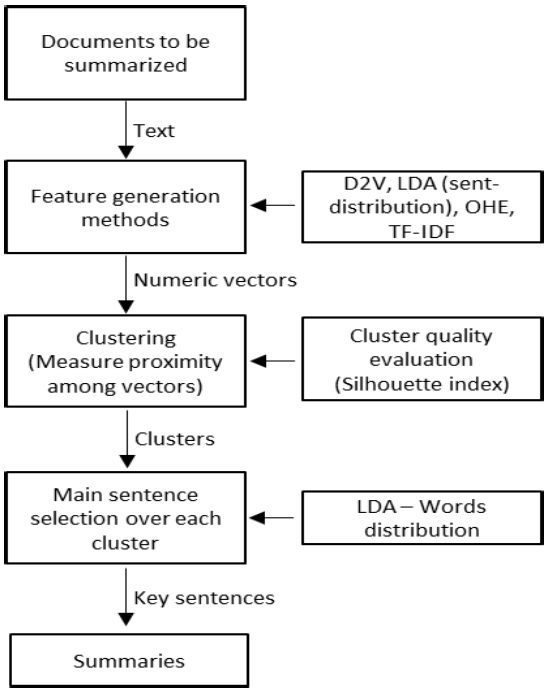


FIGURE 1. General scheme of the proposed approach.

to obtain those that will make up the summarized document (process discussed in detail in Section IV-C).

A. FEATURE GENERATION METHODS

We specifically focused on two different sources of features (i.e., TF–IDF and one-hot encoding (OHE)) with the aim of comparing and combining the mapping methods, i.e., Doc2Vec and LDA, applied in our proposed approach.

We specifically focused on four different sources of features: latent Dirichlet allocation (LDA), Doc2Vec (D2V), TF–IDF and one-hot encoding (OHE). These methods were chosen, on the one hand, because they have shown competitive results in the current research and, on the other hand, because they cover different levels of language; for example, the simple representation of OHE provides lexical features, while LDA provides semantic features, D2V provides semantic features considering the context of the words, and TF–IDF provides features related to the importance of words in a text collection.

In most of the current benchmark studies (Section II), unigrams were used as the basis for adding new features to achieve a better performance. Instead, we chose to use OHE (Section III-A.2) because it delivers a similar performance and its representation is simpler.

However, the main disadvantage of the bag-of-words method is that context information is lost. Therefore, we opted to use unsupervised algorithms to generate the semantic relations: a method based on context (Doc2Vec; see Section III-A.4) and a probabilistic generative model (LDA; see Section III-A.3). These methods automatically create a vector space where words having opposite meanings are at

a distance from each other. Furthermore, in Doc2Vec and LDA, the set of words can change according to the dataset, suggesting that the generated categories are specific to the document collection, and thus, the features may be more informative.

1) FEATURES BASED ON TERM FREQUENCY–INVERSE DOCUMENT FREQUENCY

TF–IDF reflects the importance of a word in a document and, in turn, in a dataset. This feature may be useful in the information-retrieval task of searching for similar documents; however, in the proposed framework, the relevance of the words in the document can be useful for determining whether the sentence is relevant.

2) ONE-HOT ENCODING

To build one-hot vectors, we simply **obtain an OHE representation**, in which a list of all the words W_1, W_2, …, W_n in the dataset is made. Then, we analyze each document to determine whether W_n exists in the current text. If so, feature n (F_n) is set to 1 or to 0 otherwise.

3) LATENT DIRICHLET ALLOCATION

LDA [19] is a probabilistic generative model for discrete data collections such as text collections. It represents documents as a mixture of different topics, where each topic consists of a set of words that have a link between them. Words, in turn, are chosen on the basis of probability. The process of selecting topics and words is repeated to generate a document or a set of documents. As a result, each generated document is based on different topics.

Simply stated, the generation process assumed by the LDA consists of the following steps.

1) Determine the number N of words in the document according to Poisson distribution.
2) Choose a mix of topics for the document from a fixed set of K topics according to the Dirichlet distribution.
3) Generate each word in the document as follows:
   a) Choose a topic;
   b) Choose a word in this topic.

Assuming this generative model, LDA analyzes the set of documents to reverse engineer this process by finding the most likely set of topics that make up the document.

Accordingly, given a fixed number of topics, LDA can infer the likelihood that each topic (set of words) appears in a specific document of a collection. For example, in a collection of documents and three latent topics generated using the LDA algorithm, each document would have different distributions of three likely topics. This also means that vectors of three features would be created.

4) DOC2VEC

In serveral studies on machine learning, the authors have searched for numeric representations of the studied objects. Thus, Mikolov *et al.* [20] offered a distributed representation of words to build a vector that represents the semantic meaning of each word in a set of documents, considering the context. The goal is to predict a word, given the occurrences of other words.

The process is briefly defined as follows. A matrix of words is generated by mapping all the words in the vocabulary, i.e., each column of the matrix is a word representation, where the concatenation or sum can be used as a feature to predict the next word.

Thus, given a sequence of training words, the objective is to maximize the average log probability given by

$$\sum_{t=k}^{T-k} logp(w_t \mid w_{t-k}, \ldots, w_{t+k}), \tag{1}$$

and the prediction task is provided via a multiclass classifier (softmax), following the formula

$$p(w_t \mid w_t - k, \ldots, w_t + k) = \frac{e^{y_{w_t}}}{\sum e^{y_i}}. \tag{2}$$

Each y_i in the formula above is calculated as

$$y = b + Uh(w_{t-k}, \ldots, w_{t+k}; W), \tag{3}$$

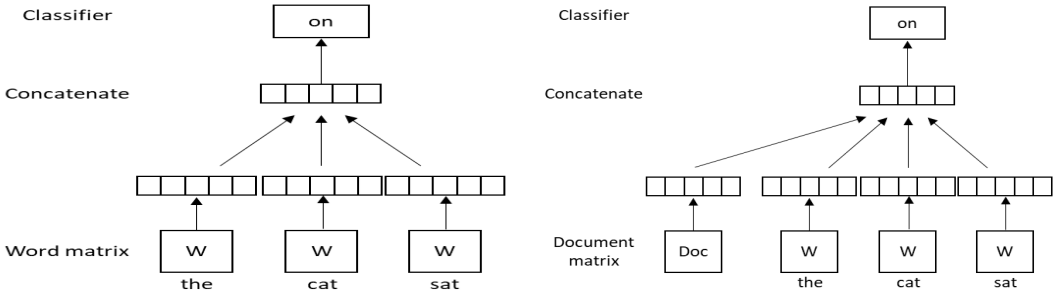
where h is constructed by the concatenation of the vectors in the word matrix W, and U and b are the softmax parameters. A hierarchical softmax is used because it offers fast training. Then, a neural network is used as the classifier, trained by stochastic gradient descent, where the gradient is obtained by backpropagation. When the algorithm converges, the words with similar meanings must be as close as possible in the vector space, unlike the opposite words, such as ''good'' and ''bad.''

The distributed representation of documents is inspired by the distribution of words. As words are predicted by the occurrence of other words, in this case, paragraphs or documents are considered in the word prediction. The paragraph vectors are mapped to the columns of matrix D, and the word vectors are mapped to matrix W. In this framework, the paragraph and word vectors are concatenated to infer the next word. Thus, the unique change is that h of Equation 3 is constructed by W and D.

In summary, **Doc2Vec** [21] is an unsupervised algorithm that generates fixed-length numeric vectors by processing a document; it was inspired by Word2Vec [20]. The difference between the two algorithms is that the former builds a fixed-length vector representation of a variable-length text, whereas the latter builds a vector for each word in the text.

As can be seen in Figure 2(a), Word2Vec generates a word matrix for predicting any next word; in contrast, Doc2Vec supplies the word matrix with paragraphs, which provide many sampled contexts (see Figure 2(b)). Thus, Doc2Vec infers new words with the word vector and a vector paragraph, which serve as a memory of the context; it establishes the topic of the document to better predict the next word.

In contrast to the bag-of-words approach, Doc2Vec can consider the ordering and semantics of the words.



(a) Word2Vec (b) Doc2Vec

FIGURE 2. Word representation schemes.

In addition, this algorithm avoids sparsity and high dimensionality, in contrast to OHE.

B. PROXIMITY MEASURES

A cluster is typically defined as a group of objects that are similar to each other; the objects in different clusters are not similar. Thus, determination of the closeness of objects is a very important process toward obtaining good-quality clusters. Different measures have been proposed to calculate the proximity between objects in a partition [22]. In this study, Euclidean and cosine proximity measures were selected and combined because they have been proven to be highly correlated with the sentence relevance [23].

Cosine similarity is frequently used to numerically represent the distance between two patterns represented as feature vectors. If two vectors consist of the same terms, the cosine value is 1; however, this value may decrease to -1. Cosine similarity is defined as

$$CS = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

where A_i and B_i are the attributes of vectors A and B, respectively.

Euclidean distance is a standard metric that represents the ordinary distance between two points. This measure is widely used in clustering problems. A true metric meets the following properties:

- Symmetry: $D(x_i, x_j) = D(x_j, xi)$
- Positivity: $D(x_i, x_j) \geq 0\, forall x_i, x_j$
- Triangle inequality: $D(x_i, x_j) \leq D(x_i, x_k) + D(x_k, x_j)$ $\forall x_i, x_j$ and x_k
- Reflexivity: $D(x_i, x_j) = 0$, if $x_i = xj$.

Euclidean distance tends to form hyper-spherical clusters. Furthermore, it is invariant to translations and rotations. The distance between two points is defined as

$$d_E(P, Q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{4}$$

where P and Q are two points of the n-dimensional space.

C. CLUSTER VALIDATION INDEXES

In a clustering problem, a measure must be chosen to validate the quality of the clustering. In the literature, various internal cluster validation indexes have been presented. Because each index has advantages and disadvantages for different datasets, we decided to select our measures according to their properties and performances on different synthetic datasets.

The goal of clustering is to build groups where the objects in the same group are similar, whereas the objects in different groups are as different as possible. Therefore, internal measures evaluate two aspects of the clusters: compactness and separation. The compactness measure indicates the degree of homogeneity of the objects in the same group. In contrast, the separation measure indicates the degree of separation of the groups from other groups.

Properties wherein each index meets at a higher or lower degree have been proposed for determining the index quality. Liu *et al.* [24] explored the use of five validation properties: monotonicity, noise, density, subclusters, and skewed distributions. Synthetic datasets allow the performance of each property for different indexes to be determined. Similarly, Rendón *et al.* [25] evaluated internal quality indexes on 12 synthetic datasets. In their study, although the property to be measured was not labeled, each dataset was built to measure the clustering index performance in different scenarios, i.e., in a distinct organization of objects. The conclusion of both of these studies [24], [25] was that the performance of the Silhouette index is better than that of the other indices. Therefore, this index was tested in this study and it is briefly discussed below.

The **Silhouette coefficient** [26] measures the closeness of each centroid in the cluster to each other object in the neighboring clusters. Thus, for each object i, the average proximity a_i between i and all other objects in the cluster where i belongs is computed. Then, for the remaining clusters c, the average proximity $d(i, c)$ to all objects in c is calculated. The smallest value of $d(i, c)$ is defined as $b_i = min_c d(i, c)$. The coefficient is defined as

$$s(i) = \frac{b(i) - a(i)}{Max\{a(i), b(i)\}} \tag{5}$$

where $SC = \frac{1}{c}\sum_{i=1}^{c} s(i)$ represents the coefficient for the complete partition.

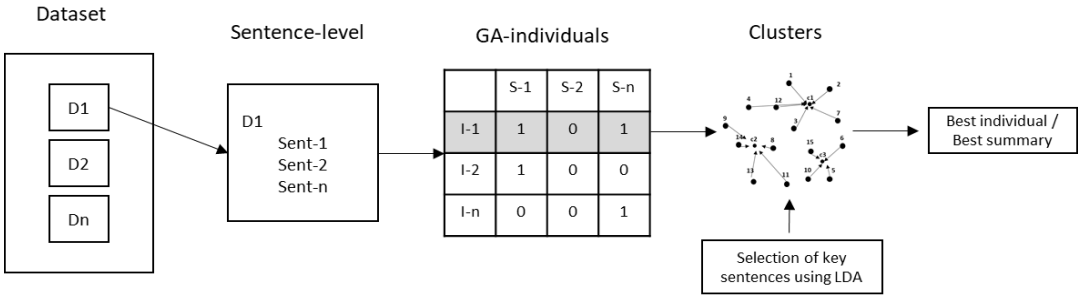


FIGURE 3. Processing steps for a document in the dataset.

IV. PROPOSED APPROACH FOR AUTOMATIC TEXT SUMMARIZATION

There are several approaches for automatically generating summaries; however, they require prior knowledge of the language, characteristics, or domain of the documents (supervised approach). Some approaches use unsupervised methods, but they apply external measures that require class labels. This type of information is typically not available in a real-world problem.

In this study, automatic summarization was tackled by clustering sentences, as described in detail in Section IV-A, by using a GA (see Section IV-B). The Silhouette index was applied as a fitness function in the GA to evaluate the quality of the groups.

For a better understanding of the proposed approach, Figure 3 shows the steps to summarize one document. First, each document is divided into sentences, which are considered the document's basic units. Next, the binary individuals of the GA represent the sentences of a certain document, where the algorithm provides the best tentative solutions of the clusters. Finally, the key sentences of the clustering are selected, on the basis of the LDA topics, as part of the summary. This process is repeated for each document in the collection.

In addition, an LDA model is incorporated into our approach, not only to build a vectorial space model but also to find the most representative sentence in each cluster formed.

To test our proposed approach, we selected two datasets: DUC02 and TAC11.

The DUC02 dataset consists of 567 news items written in English. Every news item was written by two human experts; this allowed us to compare the summaries generated by the system with those created by humans.

The TAC11 [27] dataset contains texts in different languages: Arabic, Czech, French, Greek, Hebrew, and Hindi. Each language has a compilation of 100 documents, which deal with 10 different topics, and, in turn, each topic contains 10 documents with some shared event sequences. Unlike the DUC02 task, the TAC11 task is multi-document and multi-domain as the goal is to generate a summary from 10 documents and, in addition, the documents come from different topics.

A. PARTITIONAL CLUSTERING REPRESENTATION

Following the human behavior wherein people create summaries by choosing the most important sentences in a document, we attempted to capture the key sentences, in the source document, by considering that they are surrounded by other similar ideas, just as a centroid is surrounded by attracted patterns. Therefore, this clustering representation involves two aspects: (1) the generation of the word space model (WSM) and (2) the selection of proximity measures.

1) Two common methods for mapping texts to numeric vectors were used to obtain the WSM representation: TF–IDF and OHE. In addition, we propose building LDA and Doc2Vec models to add semantic information to the feature vectors. Thus, the next step is to measure the distance between vectors. Then,
2) To obtain the proximity between objects, we combined two measures, namely, Euclidean and cosine, as these combined measures were proven to outperform other measures in clustering problems [28] and empirically proven also to obtain better results in this study.
   Because the cosine measure represents similarity and the Euclidean measure represents the distance between objects, we turn the Euclidean distance measure into a similarity measure by using the following adequacy: $modifyEuclidean = \frac{1}{Euclidean+1}$; $similarityEuclidean$ obtains values in the range (0, 1], where 1 means that the objects are the same and values close to 0 means that the objects are highly dissimilar. The cosine measure was modified by simply adding a unit to obtain only positive values: $modifiedCosine = cosine + 1$ in the range [1, 2]. Finally, the similarity between two objects is given by $modifiedEuclidean * modifyCosine$.

To calculate all the distances among objects, we created a proximity matrix. In the framework of this study, the objects are the sentences in the document to be summarized. Thus, for N sentences, we define an $N \times N$ symmetric matrix, where the intersection of i and j represents the similarity between the i^{th} and the j^{th} sentence.

To generate the groups of similar objects, we use the basics of partitional clustering algorithms, i.e., assigning each object (sentence) to the closer centroid. Therefore, if there are n-centroids, then n-groups should be created.

Formally stated, given a set of objects $X = x_1, \ldots, x_N$, where $x_j = (x_{j1}, \ldots, x_{jd}) \in R_d$, with each measure x_{ji} called a feature: partitional clustering attempts to seek a k-partition of X, $C = C_1, \ldots, C_K$ with $K \leq N$, such that:

- $C_i \neq \emptyset, i = 1, \ldots, K$;
- $\cup_{i=1}^{K} C_i = X$;
- $C_i \cap C_j = \emptyset, i, j = 1, \ldots, K$ and $i \neq j$.

However, determination of the number of groups to be generated to find the best solution becomes a combinatorial problem; that is, partitional algorithms may organize a set of sentences into K clusters. Therefore, given a set of sentences $x_i \in \mathcal{R}^d, i = 1, \ldots, N$, it is possible to enumerate all possibilities to determine the best solution. However, this brute-force approach is infeasible because it becomes a problem that is extremely expensive computationally, as suggested in [29].

$$P(N, K) = \frac{1}{K!} \sum_{m=1}^{K} (-1)^{K-m} C_K^m m^N \tag{6}$$

The possible solution for grouping 30 sentences into three clusters is 2×10^{14}. Therefore, we decided to use a heuristic, as described in detail in Section IV-B, to provide the best approximate solutions.

B. GENERATING PARTITIONS USING A GENETIC ALGORITHM

A GA representation is proposed to find the best combination of sentences to provide good-quality summaries. Therefore, the individuals are configured as follows: The number of genes in each individual is equal to the number of sentences in the document to be summarized. In turn, the individual codification is binary, and, thus, each gene may be set to 1 or 0, where 1 means that the sentence is a centroid and 0 means otherwise.

The initial population is generated by assigning a random value to each gene. That is, given the individual $P = \{g_1, g_2, \ldots, g_n\}$, where n is the total number of sentences in the document, each $g_1 = Random[0, 1]$. The sole constraint is that the generated summaries should consist of around 100 words, so that the results are comparable with those of the current benchmark studies; thus, it is possible to add sentences to the individual, i.e., the summary, until a maximum of 100 words is reached.

The activated genes ($g_n = 1$) act as attractors to the closer sentences. Thus, an individual formed of n-centroids would form n-clusters. Finally, the centroids of the groups are considered the main topics of the document, whereas the sentences attracted by the centroid are considered ideas that are close to the main topic.

The selection process over the populations was addressed by selecting the Silhouette index as a fitness function. This index considers a range of real values between 1 and -1, where the values closest to 1 represent a better clustering; therefore, those fitness values closest to 1 represent the best individuals of the population.

The principle of evolution suggests that the recombination of good solutions tends to provide outperforming solutions. However, their diversity is also important. Thus, the parents' selection process is performed by using a roulette operator that provides a high likelihood that the best solutions are selected; however, it does not completely discriminate against the bad solutions. In this study, other selection methods were also applied, such as random, rank and tournament selection; however, they proved to get inferior performance.

To generate the offspring, we propose a recombination (cross over) operator because the frequently used methods are not suitable for the summarization process. Therefore, random genes in the parent individuals are selected to be part of the new individual and only the genes with a value of 1 are considered. The minimum number of words that form the summary is verified each time a gene is selected to be part of the son chromosome.

According to the evolution scheme, there is a low probability that a mutation will occur; however, it plays an important role in the diversification of the solutions. The standard mutation operator inverts the binary value of a selected gene. However, in this study, we propose to apply this operator in the first instance to genes with a value of 1 and then to those with a value of 0. The purpose is to control the number of words in the summaries; as in the recombination process, the summary length is revised after each mutation is applied.

C. RANKING SENTENCES USING LATENT DIRICHLET ALLOCATION

Sentence selection can be performed by selecting the centroids of the formed clusters because the inference is that the centroid sentences are the main ideas of the document, whereas the remaining sentences are secondary ideas; however, this assumption is not quite true.

For example, if the clustering is built using TF–IDF as the mapping method, then the best configuration will guarantee that the centroids represent the sentences that are dissimilar, among them, with respect to the word relevance in the document. This representation could provide centroid sentences with relevant words, but also the opposite, i.e., sentences with few relevant words, because the centroids should meet the separation property. Given this premise, the selection of key sentences could be incorrect.

Therefore, in this study, we propose the creation of a vectorial space model by adding the semantic information obtained using an LDA model. So, the specific purpose of using of LDA is to provide information of the latent semantic links among words. An example of the sentence distribution obtained with the LDA model is shown in Figure 4; as can be seen in the figure, LDA reports the distribution at the word and sentence levels. That is, given a sentence, Topic 1 has a 0.58 probability of being part of it, and, in turn, the word "hurricane" has a 0.02321 probability of being selected into Topic 1.

The main reason why the LDA algorithm was used instead of Latent Semantic Indexing (LSI) is that the approach of

TABLE 1. Results of the automatic text summarization using different feature generation methods. The results are shown in terms of precision, recall, and F-measure based on Rouge-1.

Metric	Average R-Rouge	Average P-Rouge	Average F-Rouge
TF–IDF	0.48383	0.46627	0.47294
OHE	0.48670	**0.47090**	0.47679
LDA	0.56677	0.42616	0.48515.
Doc2Vec	0.48989	0.47018	0.47785
LDA+TF–IDF	0.56680	0.42622	0.48519
Doc2Vec+TF–IDF	0.48061	0.46437	0.47039
LDA+OHE	**0.56687**	0.42625	0.48524
Doc2Vec+OHE	0.47816	0.46568	0.47031
LDA+Doc2Vec	0.56681	0.42619	0.48518
LDA+Doc2Vec+TF–IDF	0.55497	0.43496	**0.48681**

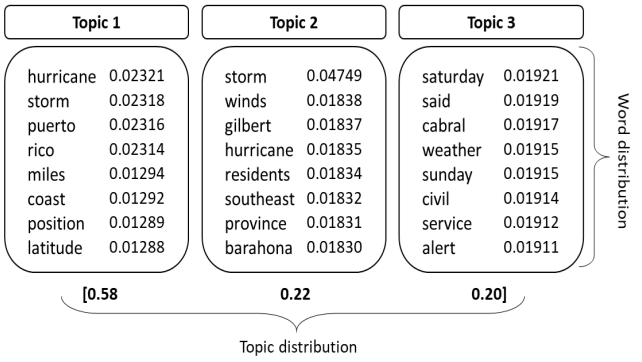


FIGURE 4. Example of the LDA distribution of a sentence.

the proposal requires the distribution of words and topics over the collection; the later can not be obtained using LSI. In addition, LDA can assign the same word to different topics to better handle polysemy. As a result, to use those words at different rates shall result in a more accurate topic distributions for each document.

The LDA model is configured to generate three topics in the experiments of this study because this configuration has been empirically proven to yield the best results. This model is applied in two steps of the proposed approach (see Figure 1): (1) in the process of mapping texts to numeric vectors and (2) in the selection of the key sentences. Both procedures are discussed in detail below.

1) Sentence distribution, represented by the topic distribution (see Figure 4), is used to complement the numeric vectors. As a result, when the clustering step is applied, the sentences are grouped by considering the themes that they contain. The clustering scheme helps to obtain a wide coverage. That is, each group built into the clustering process addresses different aspects of the main topic in question. For example, when the document is about a hurricane, one group may contain sentences discussing the location of the natural disaster, whereas a different cluster may contain sentences about the people affected. This clustering model is more appropriate for the generation of a text that describes an object, phenomenon, or fact with respect to its different aspects.

2) The clustering of sentences does not yet provide information about the key sentences in the document. Therefore, with the aim of identifying these sentences, the 10 most representative words of three topics were selected as keywords. Thus, the selection of key sentences for each cluster was conducted as follows. Given each probability pT_i associated with each topic T_i, and each probability p_i associated with each word w_i in the keywords, each word w_s in the candidate sentence was compared with w_i. If w_i was equal to w_s, $p_i * pT_i$ was accumulated in $pTotal$. The sentence that reached the maximum value of $pTotal$ was selected for generating the summary.

Table 5 shows an example of the keywords obtained by the LDA model of the original document. In addition, its corresponding human-generated and system-generated summary (reference summary) are shown below.

1) HUMAN-GENERATED SUMMARY

In retaliation against U.N. imposed economic and military ***sanctions***, ***Iraq*** *today rounded up hundreds of* ***foreign*** *nationals in* ***Kuwait***. *Some were taken to* ***Iraq***. *Britain* ***said*** *Baghdad gave no* ***reason*** *for detaining 366 people, most of them passengers from a* ***British*** *Airways flight stranded in* ***Kuwait*** *by the invasion. World* ***oil prices*** *soared to their highest level in four years as the* ***sanctions*** *effectively cut off* ***Iraqi oil*** *to world markets. President Hussein* ***warned*** *his nation to be on the alert for possible* ***U.S.*** *attacks. The United States* ***warned*** ***Iraq*** *against attacking* ***Saudi*** *Arabia and President Bush said all* ***U.S.*** *options remain open.*

2) SYSTEM-GENERATED SUMMARY

Iraq struck back against the ***West*** *today, rounding up hundreds of* ***foreign*** *nationals in* ***Kuwait***, *as the U.N. Security Council overwhelming approved sweeping trade and military* ***sanctions*** *against Iraq to punish it for invading the emirate. The* ***Iraqi*** *invaders scoured* ***hotels*** *looking for some of the thousands of Westerners based in* ***Kuwait*** *or caught by the invasion. More than 1 million foreigners live and work in* ***Kuwait***. *Several hundred Britons, Americans and* ***West*** *Germans were grabbed in the* ***hotels***, *and some were taken to Iraq, the* ***West*** *German and* ***British foreign*** *ministries* ***said***.

TABLE 2. Results of the automatic text summarization using different feature generation methods. The results are shown in terms of precision, recall, and F-measure based on Rouge-2.

Metric	Average R-Rouge	Average P-Rouge	Average F-Rouge
TF–IDF	0.22693	0.21775	0.22133
OHE	0.23057	0.22270	0.22568
LDA	0.27269	0.20521	0.23350
Doc2Vec	0.23344	**0.22339**	0.22737
LDA+TF–IDF	0.27266	0.20519	0.23347
Doc2Vec+TF–IDF	0.22379	0.21542	0.21861
LDA+OHE	**0.27273**	0.20524	**0.23353**
Doc2Vec+OHE	0.22379	0.21542	0.21861
LDA+Doc2Vec	0.27262	0.20511	0.23341
LDA+Doc2Vec+TF–IDF	0.26605	0.20849	0.23334

TABLE 3. Results of the automatic text summarization using different feature generation methods. The results are shown in terms of precision, recall, and F-measure based on Rouge-SU.

Metric	Average R-Rouge	Average P-Rouge	Average F-Rouge
TF–IDF	0.24487	0.23504	0.23884
OHE	0.24771	0.23917	0.24238
LDA	0.29219	0.21870	0.24942
Doc2Vec	0.25009	**0.23929**	0.24353
LDA+TF–IDF	0.29218	0.21871	0.24941
Doc2Vec+TF–IDF	0.24203	0.23301	0.23642
LDA+OHE	**0.29223**	0.21874	0.24946
Doc2Vec+OHE	0.24203	0.23301	0.23642
LDA+Doc2Vec	0.29216	0.21865	0.24937
LDA+Doc2Vec+TF–IDF	0.28523	0.22255	**0.24954**

TABLE 4. Comparison of the results of the proposed approach with those of other approaches. In addition, the statistical significance is shown (SS).

Approach	Rouge-1	Rouge-2	Rouge-SU	Average	p-value/SS
This work	**0.48681(1)**	**0.23334(1)**	**0.24954**	0.36007	- / -
FEOM [30]	0.46575(6)	0.12490(4)	-	0.29532	0.0101 / yes
GA approach [31]	0.48270(4)	-	-	0.24135	0 / yes
UnifiedRank [32]	0.48478(2)	0.21462(3)	-	0.34970	0.3575 / no
SFR [33]	0.48423(3)	0.22471(2)		0.35447	0.4420 / no
DE [34]	0.46694(5)	0.12368(5)	-	0.29531	0.01 / yes
NetSum [35]	0.44963(7)	0.11167(6)	-	0.28065	0.0021 / yes
CRF [36]	0.44006(8)	0.10924(7)	-	0.27465	0.001 / yes

TABLE 5. Representative document words obtained by the latent Dirichlet allocation model.

Topic	Words obtained from the original text to be summarized
Topic 1	oil, today, said, kuwait, iraqi, kuwaiti, prices, one, reasons
Topic 2	foreign, west, warned, british, sanctions, hotels, heaping
Topic 3	turkey, saudi, states, united, turkish, moving, border

The words shown in Table 5 are statistically the principal components of the original document, i.e., the essence of the document; therefore, they provide a guide to the words that should be contained in the summary.

V. RESULTS

Although the proposed approach is language and domain independent, a measure of the quality of the automatically generated summaries as compared with human-generated ones is required. The Rouge external measure [37] was used to measure the performance of the approach. Rouge measures the precision and recall to calculate the F-score of a summary that is automatically generated with respect to n-references (usually human-generated summaries). The F-score is calculated on the basis of the n-grams, and, thus, Rouge-1 is calculated on the basis of the unigrams; Rouge-2, the bigrams; and Rouge-SU, the skip-grams.

In the initial experiments, the DUC02 documents were summarized. Table 1, Table 2, and Table 3 show the results of Rouge-1, Rouge-2, and Rouge-su, respectively. It is worth noting that the LDA information increased the F-score in all cases, which indicates that the generated summary covers most words in the original document. That is, the content of the generated summary tends to be more similar to that of the original document. In addition, the statistical significance (SS) among other approaches and the proposal of this work is provided by applying a t-test. The SS was calculated taking into account the average of the results obtained of each system for Rouge-1 and Rouge-2. A confidence interval of 95% was considered.

It can be also seen that LDA adds relevant information to other methods because its performance increases when its features are provided. For example, the combination of LDA

TABLE 6. Comparison of the Rouge-1 results of the proposed approach with those of other approaches on different languages. In addition, the statistical significance is shown (SS).

Approach	Arabic	Czech	French	Greek	Hebrew	Hindi	Average	p-value/SS
This work	**0.33913** (1)	0.43643 (5)	**0.49841** (1)	**0.32770** (1)	0.30576 (3)	**0.11351** (1)	**0.33682**	- / -
CIST1	0.23190 (5)	0.46863 (3)	0.46702 (4)	0.24764 (5)	0.21566 (6)	0.02883 (5)	0.27661	0 / yes
CLASSY1	0.29188 (3)	0.48287 (2)	0.48789 (3)	0.32589 (2)	0.30154 (4)	0.06895 (3)	0.32650	0.079 / no
JRC1	0.29987 (2)	**0.48610** (1)	0.49427 (2)	0.25711 (4)	0.31205 (2)	0.10998 (2)	0.32656	0.079 / no
LIF1	0.26279 (4)	0.44620 (4)	0.46006 (5)	0.31683 (3)	**0.34731** (1)	0.02225 (6)	0.30924	0.0002 / yes
SIEL_IIITH1	-	-	0.40200	-	-	0.06850	0.07841	0 / yes
TALN_UPF1	0.27630	-	0.42045	-	-	0.09276	0.13158	0 / yes
UBSummarizer1	0.22376 (6)	0.43427 (6)	0.45914 (6)	0.17291 (6)	0.26446 (5)	0.04352 (4)	0.26634	0 / yes
UoEssex1	0.26786	-	-	-	-	-	0.04464	0 / yes
Baseline	0.23097	0.43543	0.43120	0.25228	0.29842	0.08098	0.28821	0 / yes
Topline	0.30786	0.57950	0.54315	0.35508	0.42783	0.04924	0.37711	- / -

and OHE is slightly better than that of OHE and D2V; on the other hand, the LDA features combined with D2V are slightly better than the D2V features combined with OHE. Therefore, the distribution of topics in a sentence proved to be more relevant information for detecting key sentences.

The main advantage of LDA is that it allows the latent structure of a document to be obtained; that is, we can obtain a distribution of topics and, in turn, a distribution of words. Therefore, the probable representative words of a document can be obtained for each topic distribution. In contrast, Doc2Vec provides context-based semantic information in an n-dimensional vectorial space; however, there is no information about the vector building process because it is based on a neural network.

The results showed that the best combination for achieving a high recall value is LDA+OHE, although the Doc2Vec method provides the best results in terms of precision. However, for achieving a high harmonic average (F-score), the best combination of methods is partially LDA+Doc2Vec+TF–IDF. This combination of three methods provides the best result for Rouge-1 and Rouge-SU, but does not obtain the best result for Rouge-2. This behavior is due to the the Rouge measure evaluates the final summaries through different representations of the text; Rouge-1, for example, evaluates the occurrence of unique words (unigrams), instead, Rouge-2 evaluates the occurrence of two-word sequences (bigrams). Therefore, the combination of methods for generating features tends to produce slightly different results depending on the representation of the text to be compared.

Table 4 shows a comparison between the results obtained in this study and those obtained by other approaches. As can be seen in the table, our approach outperformed the previous methods.

The results obtained show that the proposed approach has a good performance for the English language; however, this approach was designed to be language independent. Therefore, the TAC11 dataset was selected to prove that our methods can be applied to different languages.

For these experiments, our proposed approach was configured in the same way as it was for the DUC02 task. In addition, the best combination of features discovered in previous experiments was selected (LDA+Doc2Vec+TF–IDF).

TABLE 7. Global evaluation of our proposed approach with respect to the TAC11 task.

Method	Partial ranking						Global ranking
	1	2	3	4	5	6	
This work	4	0	1	0	0	0	5.00
CIST1	0	0	1	1	3	1	2.33
CLASSY1	0	2	3	1	0	0	4.33
JRC1	1	4	0	1	0	0	4.83
LIF1	1	1	0	2	1	1	3.33
UBSummarizer1	0	0	0	1	1	4	1.50

Table 6 compares the Rouge-1 results of the proposed approach with those of other approaches for the TAC11 task (ranking only the approach that evaluated the six languages). In addition, the statistical significance (SS) between other approaches and the proposal of this work is provided by applying a t-test. The SS was calculated taking into account the average of the obtained results of each system for all languages.

To show the final ranking between the results of the TAC11 task and those of our proposed approach, we used the equation (Equation 7) proposed by Aliguliyev [38]:

$$rank(method) = \sum_{s=1}^{m} \frac{(m - s + 1)r_s}{m} \tag{7}$$

where r_s stands for the number of times that the method appears in the s rank and m stands for the number of methods included in the ranking.

It can be seen in Table 7 that our proposed approach provides a competitive result on the basis of this global ranking.

VI. CONCLUSION

In this study, an approach for automatic text summarization that incorporates a vectorial space generated by different feature-generation methods was proposed. In our approach, the vectorial space is the basis of a GA that searches the best clustering of sentences. This clustering process allows the sentences of a document to be organized on the basis of certain semantic and lexical features.

The semantic features were obtained using two methods: Doc2vec and LDA. The research findings indicate that LDA provides the most relevant information for generating

good-quality summaries, as compared with the other methods used in this study. Thus, the keyword-selection process allows a more accurate detection of the representative sentences of the documents because these words tend to be contained in the key sentences.

The results on the DUC02 dataset indicate that our system outperformed previous methods, according to the evaluation results with the unigrams (Rouge-1), bigrams (Rouge-2), and skip-grams (Rouge-SU). This means that the generated summaries not only showed matches of unique words but also included context by matching the adjacent words.

None of the procedures introduced in this study require a priori information to generate the vectors. The mapping methods, namely, LDA, Doc2Vec, TF–IDF, and OHE, generate representations by processing the content of the documents themselves; in addition, the evolutionary clustering process uses the Silhouette index as a fitness function, and, therefore, knowledge about classes is not required. Thus, the proposed EATS system is language and domain independent. This assertion was proven by summarizing documents in different languages and domains from the TAC11 dataset. The results on the TAC11 dataset exhibit good performance in various languages, and our proposed approach outperformed other systems in the global rankings.

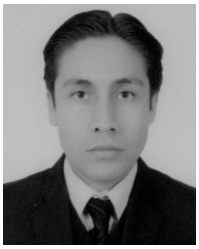
ACKNOWLEDGMENT

The authors would like to thank the Mexican Government (Cátedras CONACYT, SNI, PRODEP, Universidad Autónoma del Estado de México) for its support.

REFERENCES

[1] M. Hassan and E. Hill, "Toward automatic summarization of arbitrary java statements for novice programmers," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Sep. 2018, pp. 539–543.
[2] E. Cardinaels, S. Hollander, and B. J. White, "Automatic summarization of earnings releases: Attributes and effects on investors' judgments," *Rev. Accounting Stud.*, vol. 24, no. 3, pp. 860–890, Sep. 2019.
[3] M. Campr and K. Ježek, "Comparing semantic models for evaluating automatic document summarization," in *Proc. Int. Conf. Text, Speech, Dialogue*. Cham, Germany: Springer, 2015, pp. 252–260.
[4] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: A survey," *Artif. Intell. Rev.*, vol. 47, no. 1, pp. 1–66, Jan. 2017.
[5] M. Afsharizadeh, H. Ebrahimpour-Komleh, and A. Bagheri, "Query-oriented text summarization using sentence extraction technique," in *Proc. 4th Int. Conf. Web Res. (ICWR)*, Apr. 2018, pp. 128–132.
[6] A. Sakhadeo and N. Srivastava, "Effective extractive summarization using frequency-filtered entity relationship graphs," 2018, *arXiv:1810.10419*. [Online]. Available: http://arxiv.org/abs/1810.10419
[7] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," 2018, *arXiv:1802.08636*. [Online]. Available: http://arxiv.org/abs/1802.08636
[8] Y. Ledeneva, R. A. García-Hernández, and A. Gelbukh, "Graph ranking on maximal frequent sequences for single extractive text summarization," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*. Berlin, Germany: Springer, 2014, pp. 466–480.
[9] L. L. Bando, K. R. Lopez, M. T. Vidal, D. V. Ayala, and B. B. Martinez, "Comparing four methods to select keywords that use n-Grams to generate summaries," in *Proc. Electron., Robot. Automot. Mech. Conf. (CERMA)*, Sep. 2007, pp. 724–728.
[10] S. Charitha, N. B. Chittaragi, and S. G. Koolagudi, "Extractive document summarization using a supervised learning approach," in *Proc. IEEE Distrib. Comput., VLSI, Electr. Circuits Robot. (DISCOVER)*, Aug. 2018, pp. 1–6.
[11] A. Sinha, A. Yadav, and A. Gahlot, "Extractive text summarization using neural networks," 2018, *arXiv:1802.10137*. [Online]. Available: http://arxiv.org/abs/1802.10137
[12] R. Belkebir and A. Guessoum, "A supervised approach to arabic text summarization using AdaBoost," in *New Contributions in Information Systems and Technologies*. Cham, Germany: Springer, 2015, pp. 227–236.
[13] M. A. Fattah and F. Ren, "Automatic text summarization," *World Acad. Sci., Eng. Technol.*, vol. 37, no. 2, p. 192, 2008.
[14] J. Rojas Simón, Y. Ledeneva, and R. A. García Hernández, "Calculating the upper bounds for multi-document summarization using genetic algorithms," *Computación Sistemas*, vol. 22, no. 1, pp. 11–26, 2018.
[15] E. Vazquez Vazquez, Y. Ledeneva, and R. A. García Hernández, "Learning relevant models using symbolic regression for automatic text summarization," *Computación Sistemas*, vol. 23, no. 1, p. 127, 2019.
[16] R. A. García-Hernández, R. Montiel, Y. Ledeneva, E. Rendón, A. Gelbukh, and R. Cruz, "Text summarization by sentence extraction using unsupervised learning," in *Proc. Mex. Int. Conf. Artif. Intell.* Berlin, Germany: Springer, 2008, pp. 133–143.
[17] K. Sarkar, "Automatic text summarization using intenal and extemal information," in *Proc. 5th Int. Conf. Emerg. Appl. Inf. Technol. (EAIT)*, Jan. 2018, pp. 1–4.
[18] R. Montiel Soto, Y. Ledeneva, R. A. García-Hernández, and R. C. Reyes, "Comparación de tres modelos de texto para la generación automática de resúmenes," *Procesamiento Lenguaje Natural*, vol. 43, no. 43, pp. 303–311, Sep. 2009.
[19] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.
[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[21] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.
[22] A. Huang, "Similarity measures for text document clustering," in *Proc. 6th New Zealand Comput. Sci. Res. Student Conf. (NZCSRSC)*, Christchurch, New Zealand, 2008, pp. 49–56.
[23] A. Templeton and J. Kalita, "Exploring sentence vector spaces through automatic summarization," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 55–60.
[24] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 911–916.
[25] E. Rendón, I. Abundez, A. Arizmendi, and E. M. Quiroz, "Internal versus external cluster validation indexes," *Int. J. Comput. Commun.*, vol. 5, no. 1, pp. 27–34, 2011.
[26] 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.
[27] G. Giannakopoulos, M. El-Haj, B. Favre, M. Litvak, J. Steinberger, and V. Varma, "TAC 2011 multiling pilot overview," in *Proc. Text Anal. Conf.*, 2011, pp. 1–22.
[28] X. Gu, P. P. Angelov, D. Kangin, and J. C. Principe, "A new type of distance metric and its use for clustering," *Evolving Syst.*, vol. 8, no. 3, pp. 167–177, Sep. 2017.
[29] R. Xu and D. Wunsch, *Clustering*, vol. 10. Hoboken, NJ, USA: Wiley, 2008.
[30] W. Song, L. C. Choi, S. C. Park, and X. F. Ding, "Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 9112–9121, Aug. 2011.
[31] 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.
[32] 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.
[33] E. Vázquez, R. A. García-Hernández, and Y. Ledeneva, "Sentence features relevance for extractive text summarization using genetic algorithms," *J. Intell. Fuzzy Syst.*, vol. 35, no. 1, pp. 353–365, Jul. 2018.
[34] R. M. Alguliyev, R. M. Aliguliyev, N. R. Isazade, A. Abdi, and N. Idris, "COSUM: Text summarization based on clustering and optimization," *Expert Syst.*, vol. 36, no. 1, Feb. 2019, Art. no. e12340.

[35] K. Svore, L. Vanderwende, and C. Burges, "Enhancing single-document summarization by combining ranknet and third-party sources," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn. (EMNLP-CoNLL)*, 2007, pp. 448–457.
[36] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," in *Proc. IJCAI*, vol. 7, 2007, pp. 2862–2867.
[37] C.-Y. Lin and F. Och, "Looking for a few good metrics: Rouge and its evaluation," in *Proc. NTCIR Workshop*, 2004, pp. 1–8.
[38] R. M. Aliguliyev, "Performance evaluation of density-based clustering methods," *Inf. Sci.*, vol. 179, no. 20, pp. 3583–3602, Sep. 2009.



ÁNGEL HERNÁNDEZ-CASTAÑEDA received the M.Sc. and Ph.D. degrees in computer science (Hons.) from the Centre for Computing Research (CIC), National Polytechnic Institute (IPN), in 2013 and 2017, respectively. He is currently a Research Professor with the Autonomous University of Mexico State and a member of the National System of Researchers (SNI) of Mexico. His research interests include natural language processing, data mining, and pattern recognition.

RENÉ ARNULFO GARCÍA-HERNÁNDEZ received the B.E. degree in computer systems engineering from the Toluca Institute of Technology, Mexico, in 2001, the M.S. degree in computer science from the National Centre of Research and Technology Development (Cenidet), Mexico, in 2003, and the Ph.D. degree in computer science from the National Institute of Astrophysics, Optics, and Electronics (INAOE), Mexico, in 2017. He is currently a Full Research Professor with the School of Software Engineering and the Postgraduate School, Autonomous University of the State of Mexico (UAEM). He has authored over 70 articles in top journals and international conferences, and three books. He is an Adviser of 34 theses. He is recognized as a Second-Level National Researcher, a higher level. His research interests mainly include pattern recognition, evolutionary computation, text mining, and natural language processing. He is a member of the Mexican Association for the Natural Language Processing. He received the First place in the Entrepreneur competition in regional phase in UAEM, the Second place in the Entrepreneur competition in final phase in UAEM, and the First and Second place in the SOCO competition for the source code re-use detection in language C.

YULIA LEDENEVA received the B.Sc. and M.Sc. degrees in engineering from the Peoples' Friendship University of Russia, in 2002 and 2004, respectively, the M.Sc. degree in computer science from the National Institute for Astrophysics, Optics, and Electronics, Mexico, in 2006, and the Ph.D. degree in computer science from the Centre for Computing Research, IPN, Mexico. She is currently a Research Professor with the Autonomous University of the State of Mexico and a member of the National System of Researchers (SNI) of Mexico. She is the author of more than 70 publications. Her main research interests include computational linguistics, natural language processing, text mining, graph, and genetic algorithms. She received the Presea Lázaro Cárdenas from the hands of the President of Mexico, in 2009.

CHRISTIAN EDUARDO MILLÁN-HERNÁNDEZ received the master's and Ph.D. degrees in computer science from the Autonomous University of State of Mexico, in 2016 and 2020, respectively. His research interests are evolutionary computation, pattern recognition, machine learning, computer vision, and natural language processing.