Computer Speech and Language 71 (2022) 101267

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

Language-independent extractive automatic text summarization based on automatic keyword extraction

Ángel Hernández-Castañeda [a,b,*], René Arnulfo García-Hernández [b], Yulia Ledeneva [b], Christian Eduardo Millán-Hernández [b]

[a] *Cátedras CONACyT, Av. Insurgentes Sur 1582, Col. Crédito Constructor. C.P. 03940, Mexico*
[b] *Autonomous University of the State of Mexico, Instituto Literario 100. C.P. 50000 Col. Centro., Mexico State, Mexico*

ARTICLE INFO

Keywords:
Automatic summarization
Genetic algorithm
Topic modeling
Extractive summaries
Keywords

ABSTRACT

This study proposes a language and domain independent approach for automatic extractive text summarization (EATS) tasks, which is based on a clustering scheme supported by a genetic algorithm (GA), to find an optimal grouping of sentences. Furthermore, our approach includes a topic modeling algorithm to find the key sentences in clusters based on automatically generated keywords. Our experimental results show that our system outperforms previous methods through the application of two general steps: clustering, which helps to increase coverage, and the addition of semantic information to the model, which facilitates the detection of the key sentences in the clusters and improves precision.

1. Introduction

The automatic text summarization (ATS) task consists of automatically selecting the key ideas in a text that allow the reader to understand the target document. The ATS task is most frequently addressed by two types of methods: supervised and unsupervised.

In general, the supervised approach requires a certain amount of items with labels, indicating such features as key and secondary ideas as prior information. These labeled items constitute the training set for classifying new documents. In addition, the construction of the training set of documents requires considerable effort in this approach. Moreover, this set is very frequently not effective when it includes documents from different domains (Scanlon et al., 2020), such as scientific documents compared to news documents.

Recently, the ATS task has been addressed by using unsupervised approaches, because they do not require a set of pre-labeled items for training a classifier. Some scholars introduced unsupervised methods for ATS based on clustering schemes that search the best sentence grouping. Thus, the summaries are built based on the hypothesis that the centroids from each of the groups are the key sentences, whereas the objects around the centroids are secondary ideas (Akter et al., 2017; García-Hernández et al., 2008).

Supervised and unsupervised approaches can generate abstractive or extractive summaries. Extractive summaries are formed by considering only the information in the original texts at different levels, for example, words, sentences, and paragraphs. In contrast, abstractive summaries can contain new information (such as new sentences built by language models) generated from the original documents.

In this paper, we propose a new approach for automatic extractive text summarization (EATS) tasks, which is based on a clustering scheme supported by a genetic algorithm (GA) to find an optimal grouping of sentences. Furthermore, we include in our method a topic modeling algorithm (latent Dirichlet allocation (LDA)) to determine the key sentences in clusters based on

[*] Corresponding author.
E-mail addresses: anhernandezc@uaemex.mx (Á. Hernández-Castañeda), reagarciah@uaemex.mx (R.A. García-Hernández), nledenevay@uaemex.mx (Y. Ledeneva).

https://doi.org/10.1016/j.csl.2021.101267
Received 15 September 2019; Received in revised form 18 May 2021; Accepted 18 July 2021
Available online 27 July 2021

automatically generated keywords. To complement the semantic features, we also include a context-based model (Doc2vec) to build a vectorial space considering the context of words and their semantic links, because they have been shown to provide substantial information (Campr and Ježek, 2015) as compared to the features used in conventional methods, such as term frequency–inverse document frequency (TF–IDF) and n-grams.

The goal of this research was to design an approach that can produce summaries automatically, as far as possible, in the same manner as humans. Therefore, the challenging DUC02 dataset (Over and Liggett, 2002), which includes human-generated summaries (allowing the capability of the proposed EATS algorithm to be compared with human skills), was selected to measure the effectiveness of the proposed approach.

As an improvement on our previous work (Hernández-Castañeda et al., 2020), this study presents a more in-depth analysis of the application of the proposed summary system to different languages using the TAC11 dataset, and also explores its performance in the widely used CNN/Daily mail dataset. In addition, different benchmarks were proposed, including human performance, to complement the analysis of the results obtained.

Our experimental results, supported by an test of statistical significance, show that our system outperforms or is comparable to previous methods by virtue of the two general steps that it applies: clustering, which helps to increase coverage, and the addition of semantic information to the model, which facilitates the detection of the key sentences in the clusters and improves precision.

2. Related work

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

According to Gambhir and Gupta (2017), depending on the linguistic level, automatic text summarization techniques can be classified as extractive and abstractive.

Extractive techniques are based on a superficial analysis of the text that considers only the syntactic level, where the output summary includes text units from the original text, such as words, sentence segments, or complete sentences. The sentence is considered the unit that represents the author's idea with its complete meaning. Therefore, in the EATS task sentences are commonly considered basic units for generating summaries. In contrast, abstractive techniques consider 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 an altered interpretation that differs from that of the original author.

Most EATS research studies have been focused on extractive summaries. For instance, they considered key sentences and their position in the text (Afsharizadeh et al., 2018), measured word frequencies (Sakhadeo and Srivastava, 2018), or assigned importance levels to the sentences (Narayan et al., 2018).

At the lexical level, n-grams are frequently used to generate text models. For instance, in Ledeneva's method (Ledeneva et al., 2014), the sequences of n-grams are extracted from the text by using a model of maximal frequent sequences. In contrast, Bando et al. (2007) used n-grams to build paragraphs using the most representative terms in the document.

Features extracted from documents have been 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 (Charitha et al., 2018; Sinha et al., 2018) to generate extractive and abstractive summaries. In Belkebir and Guessoum's method (Belkebir and Guessoum, 2015), each sentence in a document is labeled "1" if it belongs to a summary, and the remaining sentences are labeled "0". Then, the authors generate a variety of features, for instance, sentence position, sentence length, and similarity to title, by applying statistics-oriented and linguistic-oriented procedures. The sentences are classified by the AdaBoost algorithm.

Fattah and Ren (2008) proposed a method that is similar to that of Belkebir and Guessoum (2015) in that a summarizer that can be trained by using a variety of extracted features is applied. However, their method differs from Belkebir and Guessoum in that the relevance of the features is considered by assigning a weight to them. This assignment is provided by a GA (Rojas Simón et al., 2018) and a regression model (Vazquez Vazquez et al., 2019). These models obtain an appropriate set of weights by processing 50 manually summarized English documents.

The main problem of supervised approaches is that a set of labeled data is required. In addition, the domain of the training samples is frequently not sufficiently general for processing multi-domain new samples (Scanlon et al., 2020).

Recently, unsupervised machine learning approaches have been utilized by applying clustering algorithms (García-Hernández et al., 2008) to group sentences based on 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, the groups of sentences need to be evaluated. Two validation methods exist for evaluating the quality of the partitions: internal and external measures (Sarkar, 2018). The former do not consider external information about the dataset classes and the latter require class labels to be applied. Various authors have compared internal and external quality measures for clustering validation. They attempted to prove experimentally which of the two approaches can evaluate the optimal groups formed from a dataset. Several quality measures were tested based on the groups built by the clustering algorithms. The results prove that internal measures perform better than external measures, generating the best configurations of groups.

In most studies focused on unsupervised approaches, external quality measures were used to validate the model's performance, e.g., F-measure; however, cluster validation indexes, i.e., internal quality measures, have been little explored in the EATS task.

In their study, Soto and García-Hernández (2009) 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. The resultant vectors were grouped by using a K-means algorithm and the final clusters were evaluated by an external measure (F-score). Their experimental results show that the maximal frequent sequences provide relevant information to the model to improve its performance.

The use of neural networks (NN) for the automatic summarization task has been popular in recent years as they have been shown to improve results in various types of tasks in NLP. However, the NN requires a large amount of data to obtain competitive results. See et al. (2017) generate abstract summaries from the CNN/Daily Mail dataset. The proposed approach applies two general steps to generate summaries. First, a pointer–generator network and a sequence-by-sequence model are combined to copy words from source texts and suggest new words from a vocabulary; second, a coverage mechanism is applied to avoid repetition of words in abstracts.

In the study of Narayan et al. (2018), extractive summaries are generated by ranking sentences in the source document. The authors' proposal is based on three general components: a sentence and document encoder and a sentence extractor. To encode the sentences, an encoder is used to map them into a continuous representation. The document encoder then obtains a representation of the document using a recurrent neural network. Therefore, the encoder–decoder model ranks the sentences and summaries are generated according to them. A reward mechanism compares the candidate summary with the gold standard to provide feedback to the reinforce mechanism for updating the model.

The graph-based algorithms have also shown competitive results. For instance, in the study of Zheng and Lapata (2019), a graph-based ranking algorithm is applied to find the most relevant sentences in the source document. Thus, a BERT representation is used to map sentences to a continuous space; the authors then measure proximity between sentences to create the graph. Unlike the common approaches that only consider the local structure of the graph, the authors propose a new centrality formula that considers the global structure of it to improve the selection of relevant sentences.

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 documents' structure. Therefore, in this paper an evolutionary clustering scheme based on a generative model (LDA) and a context-based model (Doc2vec) that provides substantial information about the latent semantic links among words are proposed.

3. Methodology

In the following sections, the proposed EATS system is described in detail. In this section, we describe the basic concept. Because the proposed approach is based on a clustering scheme, the methods for building the vectorial space are described in Section 3.1. Then, in Section 3.2 the proximity measures applied to calculate distances between objects of the vectorial space are presented. Finally, the validation indexes for evaluating groups are addressed in Section 3.3.

3.1. *Feature generation methods*

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

In most of the current benchmark studies (Section 2), unigrams were used as the basis for adding new features to achieve a better performance. Instead, we chose to use OHE (Section 3.1.2), because it shows a similar performance and its representation is simpler.

However, the main disadvantage of bag-of-words methods is that context information is lost. Therefore, we opted to use unsupervised algorithms to generate semantic relations: a method based on context (Doc2Vec; see Section 3.1.4) and a probabilistic generative model (LDA; see Section 3.1.3). These methods automatically create a vector space where words having an opposite meaning 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, features may be more informative.

3.1.1. *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 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.

3.1.2. *One-hot encoding*

To build one-hot vectors, we simply obtain an OHE representation, in which a list of all the words $W_1, W_2, \ldots, W_n$ in the dataset is formed. 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; otherwise, to 0.

3.1.3. Latent Dirichlet allocation

LDA (Blei et al., 2003) 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 based on 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 deals with 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 the Poisson distribution.
2. Choose a mix of topics for the document out of a fixed set of K topics according to 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 of which a document may consist.

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.

3.1.4. Doc2Vec

In diverse research studies on machine learning, the authors have searched for numeric representations of studied objects. Thus, Mikolov et al. (2013) 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 occurrence of other words.

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

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

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

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

$$p(w_t \mid w_t - k, \dots, w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_{e^{y_i}}}. \tag{2}$$

Each y_i in the formula above is calculated as

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

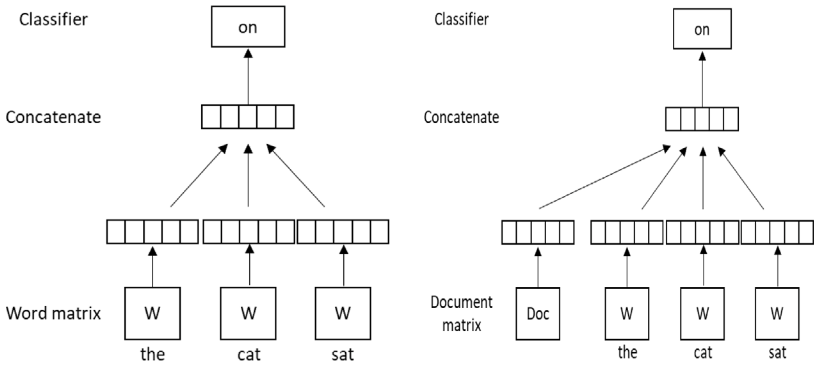
where h is constructed by the concatenation of vectors in the word matrix W and U, b are the softmax parameters. A hierarchical softmax is used, because it offers fast training. Then, a neural network is used as the classifier, trained by stochastic gradient descent, where the gradient is obtained by backpropagation. When the algorithm converges, the words with similar meanings should be as close as possible in the vector space, unlike 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 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 Eq. (3) is constructed by W and D.

In summary, Doc2Vec (Le and Mikolov, 2014) is an unsupervised algorithm that generates fixed-length numeric vectors by processing a document; it was inspired by Word2Vec (Mikolov et al., 2013). 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 Fig. 1(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 Fig. 1(b)). Thus, Doc2Vec infers new words with the word vector and a vector paragraph, which serve as memory of context; that is, it establishes the topic of the document to predict the next word better.

In contrast to the bag-of-words approach, Doc2Vec can take into account the ordering and semantics of the words. In addition, this algorithm avoids sparsity and high dimensionality, in contrast to OHE.



(a) Word2Vec (b) Doc2Vec

Fig. 1. Word representations schemes.

3.2. Proximity measures

A cluster is usually defined as a group of objects that are similar to each other; the objects in different clusters are not similar. Thus, the 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 (Huang, 2008). In this study, Euclidean and cosine distances were selected and combined, because they have been proven to be highly correlated with the sentence relevance (Templeton and Kalita, 2018).

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

$$CS = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4}$$

where A_i and B_i are attributes of vector 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, $D(x_i, x_j) \geq 0$ *for all* 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 hyperspherical clusters. Furthermore, it is invariant to translations and rotations. The distance between two points is described as

$$d_E(P, Q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{5}$$

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

3.3. 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 their performance 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 shows the degree of homogeneity of the objects in the same group. The separation measure shows the degree of separation of the groups from other groups.

Properties that each index meets at a higher or lower degree have been proposed for determining the index quality. Liu et al. (2010a) explored the use of five validation properties: monotonicity, noise, density, subclusters, and skewed distributions. Synthetic

Table 1
Basic dataset statistics.

Corpus	Source docs.		Summaries	
	docs.	Avg. length	docs.	Avg. length
DUC02	567	649	1,112	114
TAC11	60	4,391	30	265
CNN/Daily Mail	11,490	778	11,490	58

datasets allow the performance of each property for different indexes to be determined. Similarly, Rendón et al. (2011) 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, that is, in a distinct organization of objects. The conclusion of both these studies (Liu et al., 2010a; Rendón et al., 2011) was that the performance of the Silhouette index is better than that of others. Therefore, we tested this index in this study, and it is briefly described in the following section.

The **Silhouette coefficient** (Rousseeuw, 1987) measures the closeness of each centroid in the cluster to each other object in the neighbor clusters. Thus, for each object i the average proximity a_i between i and all other objects in the cluster to which 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

$$s(i) = \frac{b(i) - a(i)}{Max\{a(i), b(i)\}} \tag{6}$$

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

4. 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 4.3, by means of a GA (see Section 4.4). The Silhouette index was applied as a fitness function in the GA to evaluate the quality of the groups.

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

The proposed approach is briefly described as follows: First, each document is separated into sentences that 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 clusters. Finally, the key sentences of the clustering are selected, based on the LDA topics, as part of the summary. This process is repeated for each document in the collection.

Our system has the advantages of being language- and domain-independent, because it needs no a priori information.

4.1. Datasets

Several datasets were selected to validate our proposed approach and each one is briefly explained below:

We selected the DUC02 dataset where every news item was written by two expert humans; this allowed us to compare the summaries generated by the system with those created by humans.

The TAC11 dataset (Giannakopoulos et al., 2011) is a multi-lingual corpus on news texts that covers seven different languages: Arabic, Czech, English, French, Greek, Hebrew, Hindi. This dataset aims to evaluate the application of language independent summarization algorithms on a variety of languages. In addition, this dataset contains ten documents order to build the summary based on a multi-document scheme.

Finally, the widely used CNN/Daily Mail dataset (Hermann et al., 2015) was selected to measure the performance of our proposal relative to other supervised and unsupervised approaches.

Table 1 details the basic statistics of the source documents and abstracts. The latter were only used to calculate the Rouge measure (Lin, 2004) after our model generated the candidate summaries, and not to provide feedback in any way to the model. Therefore, our proposal can be applied to real problems where there are commonly no gold standard summaries.

4.2. Rouge measure

Lin (2004) is a proposed measure to automatically determine the quality of a summary by comparing it to ideal summaries written by humans. This measure has different versions that count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries.

4.2.1. Rouge-N

This measure is an n-gram recall between a candidate summary and a set of reference summaries. Rouge-n is calculated as follows:

$$RougeN = \frac{\sum_{S\in ReferenceSummaries} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S\in ReferenceSummaries} \sum_{gram_n \in S} Count(gram_n)} \tag{7}$$

Rouge-N is a recall-related measure because the denominator of the equations is the total sum of the number of n-grams occurring at the reference summary side. Thus, the number of n-grams in the denominator increases as more references are added and, at the same time, the space of alternative summaries are expanded. It is worth noting that by controlling what types of references are added to the reference pool, evaluations can focus on different aspects of summarization.

On the other hand, the numerator sums over all reference summaries. This gives more weight to matching n-grams occurring in multiple references. Therefore a candidate summary that contains words shared by more references is favored by Rouge-N.

4.2.2. Rouge-L

This measure views a summary sentence as a sequence of words looking for the Longest Common Subsequence (LCS). The intuition is that the longer the LCS of two summary sentences is, the more similar the two summaries are. The authors propose the LCS-based F-measure to estimate the similarity between two summaries X of length m and Y of length n, assuming X is a reference summary sentence and Y is a candidate summary sentence, as follows:

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{8}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \tag{9}$$

$$F_{lcs} = \frac{(1+\beta)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{10}$$

Where $LCS(X,Y)$ is the length of a longest common subsequence of X and Y, and $\beta = P_{lcs}/R_{lcs}$.

4.2.3. Rouge-SU: Skip-Bigram

Skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps. Skip-bigram coocurrence statistics measure the overlap of skip-bigrams between a candidate summary and a set of reference summaries. The skip-bigram-based F-measure is calculated as follows:

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)} \tag{11}$$

$$P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)} \tag{12}$$

$$F_{skip2} = \frac{(1+\beta)R_{lcs}P_{lcs}}{R_{skip2} + \beta^2 P_{skip2}} \tag{13}$$

Where $SKIP2(X,Y)$ is the number of skip-bigram matches between X and Y, β controlling the relative importance of P_{skip2} and R_{skip2}, and C is the combination function. In addition, Rouge-SU add unigram as counting unit to avoid not giving credit to a candidate sentence if the sentence does not have any word coexisting with its reference.

4.3. Partitional clustering representation

Following the human behavior where people create summaries by choosing the most important sentences in a document, we attempted to capture the key sentences by considering that they are surrounded by other similar ideas, as a centroid is surrounded by attracted patterns.

Two common methods for mapping texts to numeric vectors were used: TF–IDF and OHE. In addition, we propose building LDA and Doc2Vec models to add semantic information to the feature vectors.

To obtain proximity between objects two measures were combined: Euclidean and cosine. Because the cosine represents similarity and the Euclidean represents the distance between objects, we turn the Euclidean distance measure into a similarity measure by the following adequacy: $modifyEuclidean = \frac{1}{Euclidean+1}$; *similarityEuclidean* obtains values in the range $(0,1]$, where 1 means that objects are the same and values close to 0 means that 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 $modifiedEuclidean * modifiedCosine$.

To calculate all the distances among objects, a proximity matrix is created. In the framework of this research, 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 ith and jth sentences.

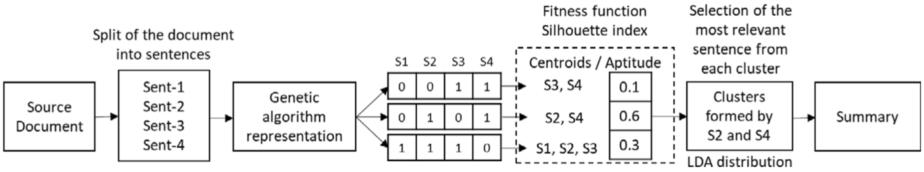

Source Document
Split of the document into sentences
Sent-1
Sent-2
Sent-3
Sent-4
Genetic algorithm representation
S1 S2 S3 S4
0 0 1 1
0 1 0 1
1 1 1 0
Fitness function Silhouette index
Centroids / Aptitude
S3, S4 0.1
S2, S4 0.6
S1, S2, S3 0.3
Selection of the most relevant sentence from each cluster
Clusters formed by S2 and S4
LDA distribution
Summary

Fig. 2. Summary generation example using a genetic algorithm representation.

To generate groups of similar objects, the basics of partitional clustering algorithms are used; however, the 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 $x_i \in \mathcal{R}^d, i = 1, \dots, N$, 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 (Xu and Wunsch, 2008), as suggested in

$$P(N,K) = \frac{1}{K!}\sum_{m=1}^{K}(-1)^{K-m}C_K^m m^N \tag{14}$$

where N is the number of attributes (dimensions) of the patterns, K is the number of clusters in the partition, and C_k stands for a particular cluster.

The possible solution for grouping 30 sentences into 3 clusters is 2×10^{14}. Therefore, we decided to use a heuristic, as described in detail in Section 4.4, to provide the best approximate solutions.

4.4. Generating partitions using a genetic algorithm

A GA representation is proposed to find the best combination of sentences to provide good-quality summaries (see Fig. 2). Therefore, the individuals are configured as follows: The number of chromosomes 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 chromosome may be set to 1 or 0, where 1 means that the sentence is a centroid and 0 means that it is not.

The initial population is generated by assigning a random value to each gene. That is, given the individual $P = \{g_1, g_2, \dots, g_n\}$, 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 not exceed a maximum limit established per dataset, 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 the maximum limit of 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 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 the bad solutions.

To generate offspring, a recombination operator is proposed, because common recombination operators can generate individuals with a large number of activated genes, that is, candidate abstracts could contain many sentences which is undesirable for the next generation. Therefore, random genes in the parent individuals are selected to be part of the new individual, taking into account that only genes with a value of 1 are considered. The minimum number of words, which 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 mutation will occur; however, it plays an important role in the diversification of solutions. The standard mutation operator inverts the binary value of a selected gene. However, in this paper we propose applying 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.

4.5. Using latent Dirichlet allocation to select representative sentences

The 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, while 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, that is, sentences with few relevant words, because centroids should meet the separation property. Given this premise, the selection of key sentences could be incorrect.

Table 2
Representative document words obtained by 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

Therefore, in this paper we suggest creating a vectorial space model by adding semantic information obtained using an LDA model. Thus, it is possible to create groups with sentences about different topics. 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.

In view of the above, the clustering scheme helps to obtain wide coverage. That is, each group built in 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.

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 LDA model was configured to generate three topics, because this configuration was empirically proven to yield the best results. Thus, the 10 most representative words of 3 topics were selected as keywords.

The selection of key sentences for each cluster was conducted as follows. Given each probability p_i associated with each word w_i in the keywords, each word w_s in the candidate sentence was compared to w_i. If w_i was equal to w_s, p_i was accumulated in $pTotal$. The sentence that reached the maximum value of $pTotal$ was selected for generating the summary.

In Table 2, an example of the keywords obtained by the LDA model of the original document is shown. In addition, its corresponding human-generated summary (reference summary) is shown below.

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.*

The words shown in Table 2 are statistically the principal components of the original document, i.e., those that make the document make sense; therefore, they provide a guide to the words that should be contained in the summary.

4.6. Experimental setup

In this study, several parameters of different algorithms, such as: the GA, Doc2Vec and LDA, were adjusted. This was carried out based on a set of experiments to achieve good-quality summaries according to the ROUGE measure. The DUC01 (Paul, 2001) dataset was used for hyper-parameter tuning since, such as DUC02, consist of pairs of human-made summaries. Candidate summaries were generated tuning hyper-parameters and comparing them with its respective references (human-made summary). Therefore, the better the Rouge result, the better the fit of the hyper-parameters.

In addition, the Python *gensim* library was used to obtain the LDA and Doc2Vec models. These models were built on each dataset separately and created using only the information available in those datasets. Thus, each model is created using the source documents and the numeric vectors adjusted in the building process represent sentences from those documents.

On the one hand, according to the performance of the GA algorithm, and also following the suggestions of the literature, a crossover and mutation rate of 0.7 and 0.3, respectively, was selected. In addition, the number of generations was set at 50 since fitness function did not show significant changes from that point on.

On the other hand, the LDA hyperparameters were selected as follows. Since a low value of alpha means that the source document consists of few topics and a low value of beta means that each topic consists of few words, the selection of these hyperparameters was as follows. First, the alpha value was set to 0.5 because a summary document typically consists of only a few topics. Second, the beta value was set to 0.001 to obtain few word on each topic, as only the top ten words are useful for key ideas selection.

In the case of Doc2Vec, the standard hyperparameters of *gensim* were used and the vector size was set to 100.

5. Results

Although the proposed approach is language- and domain-independent, a measure of the quality of the automatically generated summaries as compared to human-generated ones was required. The ROUGE external measure (Section 4.2) was used to measure the performance of the approach by running ROUGE-1.5.5.[1] with the stemming but not removal of stopwords[2]

[1] https://github.com/nisargjhaveri/ROUGE-1.5.5-unicode.git

[2] ROUGE-1.5.5.pl -a -d -n 2 -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0

Table 3
Automatic text summarization results on the DUC02 dataset using different feature generation methods. The results are shown in terms of precision, recall and F-measure based on Rouge-1.

Metric	Avg-R-Rouge	Avg-P-Rouge	Avg-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**

Table 4
Automatic text summarization results on the DUC02 dataset using different feature generation methods. The results are shown in terms of precision, recall and F-measure based on Rouge-2.

Metric	Avg-R-Rouge	Avg-P-Rouge	Avg-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 5
Automatic text summarization results on the DUC02 dataset using different feature generation methods. The results are shown in terms of precision, recall and F-measure based on Rouge-SU.

Metric	Avg-R-Rouge	Avg-P-Rouge	Avg-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**

In addition, the statistical significance (SS) is provided among other approaches and the proposal of this work for each dataset analyzed (see Tables 6 to 8) by applying a t-test. This test of significance was selected because it allows comparison between two machine learning models; therefore, the ROUGE script was not used for this process because it lacks this feature. The SS was calculated taking into account the average of the Rouge results obtained of each system. A confidence interval of 95% was considered, which means that a system will reach statistical significance if the p-value obtained is lower than 0.05.

5.1. DUC02 dataset results

In Table 3, Table 4, and Table 5 the results of ROUGE-1, ROUGE-2 and ROUGE-SU, respectively, are shown. It is noteworthy that the LDA information increases 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 the original document.

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.

On the one hand, the results show 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 LDA+Doc2Vec+TF–IDF.

Table 6
Comparison of the results of the proposed approach with those of other approaches for the DUC02 dataset. The number in brackets represents a ranking among the proposed systems. In addition, statistical significance (SS) is shown with a 95% confidence interval, that is, a *p*-value less than 0.05 is statistically significant.

Approach	Rouge-1	Rouge-2	Rouge-SU	Average	p-value	SS
This work	**0.48681(1)**	**0.23334(1)**	**0.24954**	0.36007	–	–
FEOM (Song et al., 2011)	0.46575(6)	0.12490(4)	–	0.29532	0.010	yes
GA approach (García-Hernández and Ledeneva, 2013)	0.48270(4)	–	–	0.24135	0.000	yes
UnifiedRank (Wan, 2010)	0.48478(2)	0.21462(3)	–	0.34970	0.357	no
SFR (Vázquez et al., 2018)	0.48423(3)	0.22471(2)		0.35447	0.442	no
COSUM (Alguliyev et al., 2019)	0.46694(5)	0.12368(5)	–	0.29531	0.010	yes
NetSum (Svore et al., 2007)	0.44963(7)	0.11167(6)	–	0.28065	0.002	yes
CRF (Shen et al., 2007)	0.44006(8)	0.10924(7)	–	0.27465	0.001	yes

On the other hand, in strict terms, there is no statistical significance in the comparison of the feature generation methods for this dataset because the *p*-value obtained among the F-measure results was greater than 5%. However, the feature generation methods assembly might be useful in datasets with larger source texts where semantic models can be more robust.

We show in Table 6 the comparison between the results obtained in this study and those obtained by other approaches. As can be seen, our approach outperforms the previous methods. The comparison approaches are briefly explained below:

- Two general steps are carried out in the FEOM approach (Song et al., 2011); on the one hand, the authors applied a k-means algorithm to group sentences in order to organize them according to their content; on the other hand, an optimization model is proposed for the selection of phrases through the groups formed. Finally, the selected phrases form the final summary.
- The GA approach (García-Hernández and Ledeneva, 2013) proposes organizing like-minded sentences according to their content from a GA representation. In this scheme, each abstract is generated for a GA instance where the individual in the population represents the document to be summarized and the genes represent the sentences of the source document. The objective function is given by a measure F adapted for the automatic summary task; therefore, the authors generate the clustering representation based on this function and select centroids as key ideas. The latter form the summary.
- In the UnifiedRank approach (Wan, 2010), the score of a sentence is calculated with respect to two aspects: the relevance of the sentences in a particular document and in the whole set of documents. Therefore, the authors construct a graph that reflects the sentence-to-sentence relationship based on previously calculated scores. Finally, the raking of sentences is obtained based on a graph-based learning algorithm.
- In the SFR approach (Vázquez et al., 2018), several characteristics were evaluated for the selection of the key phrase in order to assign a weight to each one. The authors consider characteristics such as: similarity of title, position and length of the sentences, among others. In this way, a GA is applied to find optimal weights and produce a fitness function in order to generate good quality summaries.
- The COSUM approach (Alguliyev et al., 2019) proposes two stages to summarize documents. In the first stage, to discover all the topics of the document, a k-means algorithm is applied in order to group the sentences of the source document. In the second stage, the authors attempted to optimize an objective function to find key sentences in clusters by using an adaptive differential evolution algorithm.
- NetSum (Svore et al., 2007) is a supervised approach. The authors compiled a dataset consisting of 1,365 news items from CNN.com, then the training dataset was tagged with the goal of identifying the best sentences. Finally, a set of characteristics was extracted from each sentence in the training and test sets to generate the model based on the distribution of training characteristics. In this way, the model obtained is able to identify, as far as possible, the key phrases of the set of tests.
- The CRF approach (Shen et al., 2007) tackle the automatic summarization task as a sequence labeling problem. The authors calculate the relevance of the sentence according to various characteristics based on a conditional random forest. In turn, the proposed system can even take as inputs the output of other systems.

The proposed approach shares some basic aspects with respect to the current standard methods detailed above. For example, FEOM, COSUM and NetSum implement two stages: sentence clustering and key sentence selection across the clusters; however, the main difference in the first stage, with respect to the proposal of this study, is the GA implementation that can find an optimal clustering representation.

The GA approach is very similar to the first stage of this study, that is, both approaches implement a GA algorithm to obtain an optimal number of clusters. However, the main difference is the fitness function; GA's approach uses a modified F-measure that requires parsing the dataset documents to perform a count of similarities, instead our approach uses a cluster validation index as a fitness function that not requires to analyze the documents.

As previously stated, unlike other approaches, the first stage (clustering) of our proposal is the use of the silhouette index as a fitness function. On the other hand, the main difference of the second stage of this study, with respect to others, is the use of the LDA model to identify the key ideas, instead of selecting centroids.

Table 7
Comparison of the Rouge-1 results of the proposed approach with those of other approaches (CIST, CLASSY, JRC and LIF) in different languages for the TAC11 dataset. The number in brackets represents a ranking among the proposed systems. In addition, statistical significance (SS) is shown with a 95% confidence interval, that is, a p-value less than 0.05 is statistically significant.

Lang.	This work	CIST	CLASSY	JRC	LIF
Arabic	**0.33913** (1)	0.23190 (5)	0.29188 (3)	0.29987 (2)	0.26279 (4)
Czech	0.43643 (5)	0.46863 (3)	0.48287 (2)	**0.48610** (1)	0.44620 (4)
French	**0.49841** (1)	0.46702 (4)	0.48789 (3)	0.49427 (2)	0.46006 (5)
Greek	**0.32770** (1)	0.24764 (5)	0.32589 (2)	0.25711 (4)	0.31683 (3)
Hebrew	0.30576 (3)	0.21566 (6)	0.30154 (4)	0.31205 (2)	**0.34731** (1)
Average	**0.33682**	0.27661	0.32650	0.32656	0.30924
p-value	–	0.000	0.079	0.079	0.000
SS	–	yes	no	no	yes

5.1.1. Baselines and human performance

With the aim of compare the performance of the proposed approach of this work, a set of proposed baselines (García-Hernández et al., 2008) were taken as reference: random line and top line. These baselines are calculated by using DUC02 dataset. Furthermore, we measured the human performance on this dataset.

- The baseline random consists of randomly selecting sentences to form the summary; the Rouge-1 F-measure obtained was 0.38981.
- The top line consists of obtaining a high value of Rouge F-measure by thoroughly forming several combinations of sentences; thus, a genetic algorithm was used to select among all sentences in the document those of them which maximizes the Rouge measure. It is worth noting that this process is not applicable for real summarization problems because Rouge requires ideal summaries to compare with candidate ones. The Rouge-1 F-measure obtained was 0.62367.
- In this work, human performance is measured using the summaries written by experts on DUC02. Due to the fact that there are two reference summaries for each document in the dataset, one of them was taken as a candidate and the other as a reference. This makes it possible to compare performance between humans because experts make their own selection of main information. The Rouge-1 F-measure obtained was 0.50195.

According to the proposed baselines, even using a brute force method, the best performance obtained was the top line (0.62367). On the one hand, it is worth noting that the top line is higher than human performance (0.50195) and also reaches statistical significance (p-value $= 2E - 5$). On the other hand, if we compare the best Rouge-1 result of this work on DUC02 (0.48681) and human performance (0.50195), the p-value obtained is 0.6102 which means that there is not statistical significance (i.e. p-value > 0.05). This makes our approach, in terms of the Rouge measure, comparable to human performance and better than random selection (random line). An example of a summary generated by this work approach and a human generated one is shown below.

Computer-based summary: *Lefer said his research team **simulated heart attacks** in 24 **rats** by partially blocking key arteries in their **hearts**. In 12 of the **rats**, the researchers injected a placebo. In the other 12, they injected **transforming growth factor beta**. There was about **50 percent less** injury with **TGF beta than** without it. Thus, by **measuring** for the loss of this **substance**, researchers could determine the **amount** of **heart damage**.*

Summary written by an expert: *According to a study published in the journal Science, cardiac cell **damage** following a **heart** attack may be limited by a natural **substance** called **transforming growth factor beta**. In studies with **rats** induced to have **heart attacks**, those **rats** receiving injections of **TGF beta** had **50 percent less** cell **damage than** those not receiving the injections. **TGF beta** is normally present in **heart** cells but is missing from rat **heart** cells damaged by the **simulated heart** attack. The extent of **heart** cell **damage** was determined by **measuring** the **amount** of creatine kinase in the **heart** tissue. **Hearts** damaged by a **heart** attack tended to lose creatine kinase.*

As can be seen, the computer-based summary was able to obtain the main ideas present in the expert-written summary, for example, *heart damage*, *TGF beta*, and *rat studies*. Furthermore, the proposed approach of this study was able to obtain more specific information, such as the number of rats in the study or the percentage of less injury when using TGF beta. However, the writing of the human-made summary is more intuitive.

5.2. TAC11 dataset results: multi-language summary generation

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 compares the Rouge-1 results of the proposed approach with those of other approaches for the TAC11 task (ranking only the approaches that assessed all six languages). Each approach is briefly explained below.

- LIF (Hmida and Favre, 2011) is a system based on the Maximal Marginal Relevance (MMR) algorithm. MMR is a greedy method to extract iteratively the most relevant sentences relative to a query to generate a summary, while minimizing redundancy. At each iteration, the sentence added to the selection maximizes the similarity to the query while minimizing the similarity to sentences already selected.

Table 8
Comparison of the proposed approach with respect to other approaches on CNN/Daily Mail dataset.

Approach	ROUGE-1	ROUGE-2	ROUGE-L	Average	p-value	SS
This work	41.4	18.4	37.6	32.4	–	–
Narayan et al. (2018)	40.3	17.7	36.6	31.5	0.071	no
See et al. (2017)	39.5	17.2	36.3	31.0	0.011	yes
Zheng and Lapata (2019)	54.7	30.4	50.8	45.3	0.000	yes

- JRC (Steinberger et al., 2011) is a system based on Latent Semantic Analysis (LSA). Simple stated, this approach first builds a term-by-sentence matrix from the source, then applies Singular Value Decomposition (SVD) and finally uses the resulting matrices to identify and extract the most salient sentences. SVD finds the latent dimensions, which correspond to the different topics discussed in the source.
- CIST (Liu et al., 2010b) is a system based on hierarchical Latent Dirichlet Allocation that is a representative generative probabilistic model, which can mine latent topics and organize them into hierarchy to achieve deeper semantic analysis. Thus, the hLDA model is combined with some traditional features to select the most representative sentences of the source document.
- CLASSY (Conroy et al., 2011) is a system that attempts to estimate the probability that a term is included in a human generated summary. This system selects the main sentences based on the summary of terms created.

5.3. CNN/Daily mail dataset results

Finally, the proposed approach of this study was applied to the widely used CNN/Daily Mail dataset. Table 8 shows the comparison against other supervised and unsupervised approaches. For example, Narayan et al. (2018) and See et al. (2017) proposed supervised methods based on neural networks (detailed in Section 2); as can be seen, our proposal obtained competitive results with respect to these methods. On the other hand, Zheng and Lapata (2019) proposed an unsupervised graph-based approach that clearly outperform the approaches in the comparison table; however, the use of BERT to map vectors into a continuous semantic space required the training set of the CNN/Daily Mail dataset. Furthermore, the authors' proposal requires a specific version of BERT for each language. This makes the proposal a semi-supervised and language-dependent approach.

6. Conclusions

In this paper, a new approach for automatic 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 based on certain semantic and lexical features.

The semantic features were obtained using two methods: Doc2vec and LDA. The research findings show that LDA provides the most relevant information for generating good-quality summaries, as compared with the other methods addressed in this study. Thus, the key word selection process allows the more accurate detection of the representative sentences of the documents, because these words tend to be contained in the key sentences.

None of the procedures introduced in this paper require a priori information to generate vectors. The mapping methods, 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, as shown in the results of the TAC11 dataset, the proposed EATS system is language- and domain-independent.

The results on DUC02 show that our system outperforms previous methods, according to its evaluation at the level of unigrams (ROUGE-1), bigrams (ROUGE-2), and skip-grams (ROUGE-SU). This means that the generated summaries not only show matches of unique words, but also include context by matching adjacent words. Furthermore, the approach proposed in this study obtained competitive results in the CNN/Daily Mail dataset.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

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

References

Afsharizadeh, M., Ebrahimpour-Komleh, H., Bagheri, A., 2018. Query-oriented text summarization using sentence extraction technique. In: 2018 4th International Conference on Web Research (ICWR). IEEE, pp. 128–132.

Akter, S., Asa, A.S., Uddin, M.P., Hossain, M.D., Roy, S.K., Afjal, M.I., 2017. An extractive text summarization technique for Bengali document (s) using K-means clustering algorithm. In: 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (IcIVPR). IEEE, pp. 1–6.

Alguliyev, R.M., Aliguliyev, R.M., Isazade, N.R., Abdi, A., Idris, N., 2019. COSUM: Text summarization based on clustering and optimization. Expert Syst. 36 (1), e12340.

Bando, L.L., Lopez, K.R., Vidal, M.T., Ayala, D.V., Martinez, B.B., 2007. Comparing four methods to select keywords that use n-grams to generate summaries. In: Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007). IEEE, pp. 724–728.

Belkebir, R., Guessoum, A., 2015. A supervised approach to arabic text summarization using adaboost. In: New Contributions in Information Systems and Technologies. Springer, pp. 227–236.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3 (Jan), 993–1022.

Campr, M., Ježek, K., 2015. Comparing semantic models for evaluating automatic document summarization. In: International Conference on Text, Speech, and Dialogue. Springer, pp. 252–260.

Charitha, S., Chittaragi, N.B., Koolagudi, S.G., 2018. Extractive document summarization using a supervised learning approach. In: 2018 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER). IEEE, pp. 1–6.

Conroy, J.M., Schlesinger, J.D., Kubina, J., Rankel, P.A., O'Leary, D.P., 2011. CLASSY 2011 at TAC: Guided and multi-lingual summaries and evaluation metrics. Tac 11, 1–8.

Fattah, M.A., Ren, F., 2008. Automatic text summarization. World Acad. Sci. Eng. Technol. 37, 2008.

Gambhir, M., Gupta, V., 2017. Recent automatic text summarization techniques: a survey. Artif. Intell. Rev. 47 (1), 1–66.

García-Hernández, R.A., Ledeneva, Y., 2013. Single extractive text summarization based on a genetic algorithm. In: Mexican Conference on Pattern Recognition. Springer, pp. 374–383.

García-Hernández, R.A., Montiel, R., Ledeneva, Y., Rendón, E., Gelbukh, A., Cruz, R., 2008. Text summarization by sentence extraction using unsupervised learning. In: Mexican International Conference on Artificial Intelligence. Springer, pp. 133–143.

Giannakopoulos, G., El-Haj, M., Favre, B., Litvak, M., Steinberger, J., Varma, V., 2011. TAC 2011 MultiLing pilot overview. TAC.

Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P., 2015. Teaching machines to read and comprehend. In: Advances in Neural Information Processing Systems. pp. 1693–1701.

Hernández-Castañeda, A., García-Hernández, R.A., Ledeneva, Y., Millán-Hernández, C.E., 2020. Extractive automatic text summarization based on lexical-semantic keywords. IEEE Access 8, 49896–49907.

Hmida, F., Favre, B., 2011. LIF At TAC multiling: Towards a truly language independent summarizer. Theory Appl. Categ.

Huang, A., 2008. Similarity measures for text document clustering. In: Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand. pp. 49–56.

Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents. In: International Conference on Machine Learning. pp. 1188–1196.

Ledeneva, Y., García-Hernández, R.A., Gelbukh, A., 2014. Graph ranking on maximal frequent sequences for single extractive text summarization. In: International Conference on Intelligent Text Processing and Computational Linguistics. Springer, pp. 466–480.

Lin, C., 2004. ROUGE: A package for automatic evaluation of summaries. text summarization branches out; association for computational linguistics: Barcelona. Spain.

Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J., 2010a. Understanding of internal clustering validation measures. In: Data Mining (ICDM), 2010 IEEE 10th International Conference on. IEEE, pp. 911–916.

Liu, H., Liu, P., Heng, W., Li, L., 2010b. The CIST summarization system at TAC 2011. Theory Appl. Categ..

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119.

Narayan, S., Cohen, S.B., Lapata, M., 2018. Ranking sentences for extractive summarization with reinforcement learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics, New Orleans, Louisiana, pp. 1747–1759. http://dx.doi.org/10.18653/v1/N18-1158, https://www.aclweb.org/anthology/N18-1158.

Over, P., Liggett, W., 2002. Introduction to DUC: an intrinsic evaluation of generic news text summarization systems. Proc. DUC. Http://Wwwnlpir.Nist.Gov/Projects/Duc/Guidelines/2002.Html.

Paul, O., 2001. Introduction to DUC-2001: an intrinsic evaluation of generic news text summarization systems. In: Proceedings of DUC 2001 Document Understanding Conference. Vol. 49.

Rendón, E., Abundez, I., Arizmendi, A., Quiroz, E.M., 2011. Internal versus external cluster validation indexes. Int. J. Comput. Commun. 5 (1), 27–34.

Rojas Simón, J., Ledeneva, Y., García-Hernández, R.A., 2018. Calculating the upper bounds for multi-document summarization using genetic algorithms. ComputaciÓN Y Sistemas 22 (1), 11–26.

Rousseeuw, P.J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65.

Sakhadeo, A., Srivastava, N., 2018. Effective extractive summarization using frequency-filtered entity relationship graphs. ArXiv Preprint ArXiv:1810.10419.

Sarkar, K., 2018. Automatic text summarization using intenal and extemal information. In: 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT). IEEE, pp. 1–4.

Scanlon, L., Zhang, S., Zhang, X., Sanderson, M., 2020. Evaluation of cross domain text summarization. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1853–1856.

See, A., Liu, P.J., Manning, C.D., 2017. Get to the point: Summarization with pointer-generator networks. In: ACL.

Shen, D., Sun, J.-T., Li, H., Yang, Q., Chen, Z., 2007. Document summarization using conditional random fields. In: IJCAI. 7, pp. 2862–2867.

Sinha, A., Yadav, A., Gahlot, A., 2018. Extractive text summarization using neural networks. ArXiv Preprint ArXiv:1802.10137.

Song, W., Choi, L.C., Park, S.C., Ding, X.F., 2011. Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization. Expert Syst. Appl. 38 (8), 9112–9121.

Soto, R.M., García-Hernández, R.A., 2009. Comparación de tres modelos de texto para la generación automática de resúmenes. Procesamiento Del Lenguaje Natural 43, 303–311.

Steinberger, J., Kabadjov, M.A., Steinberger, R., Tanev, H., Turchi, M., Zavarella, V., 2011. Jrc's participation at TAC 2011: Guided and multilingual summarization tasks. TAC 11, 1–9.

Svore, K., Vanderwende, L., Burges, C., 2007. Enhancing single-document summarization by combining RankNet and third-party sources. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL).

Templeton, A., Kalita, J., 2018. Exploring sentence vector spaces through automatic summarization. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, pp. 55–60.

Vázquez, E., Arnulfo Garcia-Hernandez, R., Ledeneva, Y., 2018. Sentence features relevance for extractive text summarization using genetic algorithms. J. Intell. Fuzzy Systems 35 (1), 353–365.

Vazquez Vazquez, E., Ledeneva, Y., García Hernández, R.A., 2019. Learning relevant models using symbolic regression for automatic text summarization. ComputaciÓN Y Sistemas 23 (1), 127.

Wan, X., 2010. Towards a unified approach to simultaneous single-document and multi-document summarizations. In: Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, pp. 1137–1145.

Xu, R., Wunsch, D., 2008. Clustering. Vol. 10, John Wiley & Sons.

Zheng, H., Lapata, M., 2019. Sentence centrality revisited for unsupervised summarization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp. 6236–6247. http://dx.doi.org/10.18653/v1/P19-1628, https://www.aclweb.org/anthology/P19-1628.