Journal of Intelligent & Fuzzy Systems 35 (2018) 293–304
DOI:10.3233/JIFS-169588
IOS Press

Calculating the significance of automatic extractive text summarization using a genetic algorithm

Jonathan Rojas Simón*, Yulia Ledeneva* and René Arnulfo García-Hernández*
Universidad Autónoma del Estado de México, Unidad Académica Profesional Tianguistenco, Instituto Literario, CP, Toluca, Edo. Mex, México

Abstract. In the last 16 years with the existence of Document Understanding Conference (DUC), several methods have been developed in Automatic Extractive Text Summarization (AETS) that have allowed the continuous improvement of this task. However, no significant analysis has been performed to determine the significance of the AETS methods. In this paper, we present a new method based on a Genetic Algorithm to determine the best sentence combination of DUC01 and DUC02 datasets to rank the newest methods of AETS. Using three heuristics presented in the state-of-the-art, we rank the most recent AETS methods, obtaining upper bounds and recovering lower bounds of the state-of-the-art.

Keywords: Significance, *Topline*, text summarization, genetic algorithm, upper bounds

1. Introduction

Automatic Extractive Text Summarization (AETS) is a task contemplated in Natural Language Processing (NLP) that allows to reduce the textual content of a document or a set of them, by selecting a set of phrases or sentences more representative of the original text obtained from a method or a computational tool, using supervised and unsupervised learning techniques [30, 32, 50].

Among the first advances made in AETS, has been considered in Luhn [17] and Edmunson [16] as the pioneers of Automatic Text Summarization (ATS) and, particularly, AETS. However, the most consistent developments of the ATS were through Document Understanding Conferences (DUC) since 2001 to 2007 organized by the National Institute of Standards and Technology (NIST) [11].

With the existence of the DUC conferences, several methods have been developed that have employed automatic learning techniques [3, 18, 22, 23], text connectivity [2, 8], text representation through use of graphs [36–39], algebraic reduction [19–21] and the use of evolutionary models [25, 26, 30, 33, 34], with the purpose of generation of automatic extractive summaries that best resemble summaries made by humans.

One of the main challenges of AETS is to generate automatic extractive summaries that similar to summaries generated by humans (gold-standard summaries). However, for several domains, the gold-standard summaries are made abstracting summaries by substituting some terms and phrases of the original text. According to Verma and Lee [11, 27, 28, 35], the gold-standard summaries of DUC01 and DUC02 employ approximately 9% of words not found in the original documents [35]. Consequently, the level of

*Corresponding authors. Jonathan Rojas Simón, Yulia Ledeneva and René Arnulfo García-Hernández, Universidad Autónoma del Estado de México, Unidad Académica Profesional Tianguistenco, Instituto Literario No. 100, CP 50000, Toluca, Edo. Mex, México. E-mails: ids_jonathan_rojas@hotmail.com (J.R. Simón), yledeneva@yahoo.com (Y. Ledeneva), renearnulfo@hotmail.com (R.A. García-Hernández).

maximum similarity will be less than 100%, and even more, if compared from several gold-standard summaries, the upper bounds will be lower for any AETS method.

In several previous works [30, 31, 47, 48] heuristics have been used to compare the performance of the AETS methods. These heuristics are known as *Baseline* and *Baseline-random* [30, 47], which allow to establish the minimum performance limits (lower bounds) by which extractive summaries must be generated. However, the AETS methods have not been ranked because the best extractive summaries obtained from *Topline* heuristic were not known [14].

To know the best extractive abstracts of *Topline* heuristic, techniques based on exhaustive searches have been used in the state-of-the-art to find the best combinations of sentences that best resemble those used by humans. In some previous works, these summaries are known as *Oracle extracts* [7]. However, methods based on this technique have been used in short documents, and despite this, large-scale processing techniques (clusters) have been used to evaluate all possible combinations [7, 15], because the increase of sentences of an original text represents an exponential growth in the space of solutions. Therefore, using a method based on exhaustive searches to evaluate all possible combinations is not feasible to use.

In the other hand, the use of several evolutive methods in AETS has represented a viable solution generating extractive summaries of superior performance. These types of techniques include the use of Genetic Algorithms (GA) [30] and Memetic Algorithms (MA) [26]. Therefore, using optimization algorithms, is a viable solution to obtain extractive summaries closest to the best ones represented. In this paper, a GA is used to obtain the combinations of sentences that best resemble selected by humans using the ROUGE-1.5.5 system.

The rest of the paper is organized as follows: Section 2 presents some related works that have used techniques based on exhaustive searches to determine the best extractive abstracts. Section 3 describes the general process of a GA. Section 4 describes the structure and development of the proposed GA. Section 5 shows the GA experimental configuration to determine the highest performance sentence combinations for calculating *Topline* of DUC01 and DUC02 dataset. In addition to compare the performance of *Topline* with some methods and heuristics used in the state-of-the-art, a ranking in the performance of AETS methods are showed. Finally, Section 6 shows the conclusions and future work.

2. Related works

In the last years, with the existence of Document Understanding Conferences, many advances have been made in the development of ATS. However, to know and determine the best extractive summaries, few studies have been carried out, and some of them use techniques based on exhaustive searches to determine the best combination of sentences that best represent the judgments made by humans. Lin and Hovy [5–7] developed a comprehensive search-based method to find the best combinations of a document by taking the first 100 ± 5 and 150 ± 5 words of the DUC01 dataset, and evaluating sentence combinations by co-occurrence of bag-of-words of the ROUGE system.

This work arises due to the idea that the best combination of sentences is substantially better than any other AETS method in the state-of-the-art, allowing to know the upper bounds that any AETS method can achieve [7, 14]. However, the main drawback that affected the performance of this procedure was exponential increase of the search space that implies the number of sentences of each document. For example, if we use a document of 100 sentences and it is inferred that on average each sentence has a length of 20 words, then to find the best extractive summary of 100 words should take the best 5 sentences of the 100 available (C_{100}^{5}), generating 75,287,520 possible combinations of sentences to find the best.

In 2010, Ceilan [15] introduced a same method based on exhaustive searches to find the best sentence combinations. Unlike Lin and Hovy [7], this work was applied to summaries from different domains (literary, scientific, journalistic and legal) using a probability density function from the weights established by the ROUGE system to reduce space search solutions. However, in the experimentation stage, it was necessary to modify ROUGE-1.5.5 Perl-based script to process different combinations of sentences in a cluster of computers to distribute the processing of the documents. In addition, in the news domain it was necessary to divide the original document into several sub-sections to reduce the search and processing time of each document by discriminating the different possible combinations that can be generated.

In 2017, Wang [43] used a new strategy for finding the best combinations of one and multiple documents, using nine sentence reduction heuristics that present a low relation to the gold-standard summary. Subsequently, the remaining sentences are introduced through seven weighting methods to measure the similarity of the candidate sentences in relation to gold-standard summaries. However, the use of several heuristics to determine the best combinations of sentences in different domains and different entries allows the increase of the computational cost to find the best combinations of sentences. In addition, for summaries of a document only a single gold-standard summary was used and in the case of summaries for multiple documents only 533 documents of 567 of the DUC02 dataset were used, generating more biased results.

In this paper we propose the method based on the use of GAs to find the best combinations of sentences that can be generated from the summaries of DUC01 and DUC02 dataset and rank AETS state-of-the-art methods.

3. Basic genetic algorithm

The GAs [29, 39, 42, 49] is a technique of optimization and iterative, parallel, stochastic search inspired by the principles of natural selection proposed by Darwin in 1859 [4]. The GAs was proposed by John Holland in 1975 as a method that pretends to simulate the actions of nature in a computer to optimize a wide variety of processes [1, 24, 41]. Nowadays, GA is the most widely used evolutive computing method in the optimization problems [41].

A traditional GA is characterized by representing the solution of a problem in individuals, which are represented by variable bit strings and together form a population [24]. GA begins with a population of N_{pop} individuals who share a set of n characteristics for each generation g, where each i-th individual X_i is randomly generated as shown in Equation (1).

$$X_i(g) = \left[X_{i,1}(g),\ X_{i,2}(g), \ldots,\ X_{i,n}(g)\right], \quad i = 1,\ 2, \ldots,\ N_{pop} \tag{1}$$

Each individual $X_r(g)$ is evaluated from a specific adaptation value (fitness function) to determine the quality of individuals and its proximity to the optimal values of GA [24, 41]. From the value obtained as a fitness function, a selection of individuals is performed, where each pair of parents $X_p(g)$ and $X_m(g)$ is chosen to participate in the cross-step forming individuals $Y_i(g)$, which have combined characteristics of $X_p(g)$ and $X_m(g)$. Finally, the new individual $Y_i(g)$ is introduced to the mutation stage, where partial and minimal modifications are made to generate an individual $Z_i(g)$. As mentioned by Mendoza, the mutation of individuals is based on a probability P [26], as shown in Equation (2).

$$Z_i(g) = \begin{cases} Mutate\,(Y_i(g)) & if\ rand < P \\ Y_i & otherwise \end{cases} \tag{2}$$

where the function $Mutate\,(Y_i(g))$ modifies the order of one or more sentences selected as target from a random value *rand*, included in a probability P. Otherwise, the individual Y_i is not modified. Finally, the population is updated according to the new individuals generated from the crossing and mutation stages of individuals. During the new generations, the average fitness function of each generation is improved because each generation produces individuals with better fitness function.

The selection, crossing, and mutation of individuals are iterated until they meet a certain termination criterion, these criteria are based on the number of iterations, the convergence of individuals of a gene, and on a fitness function [49]. In summary, the process that conducts a GA is guided in Fig. 1 [24, 25].

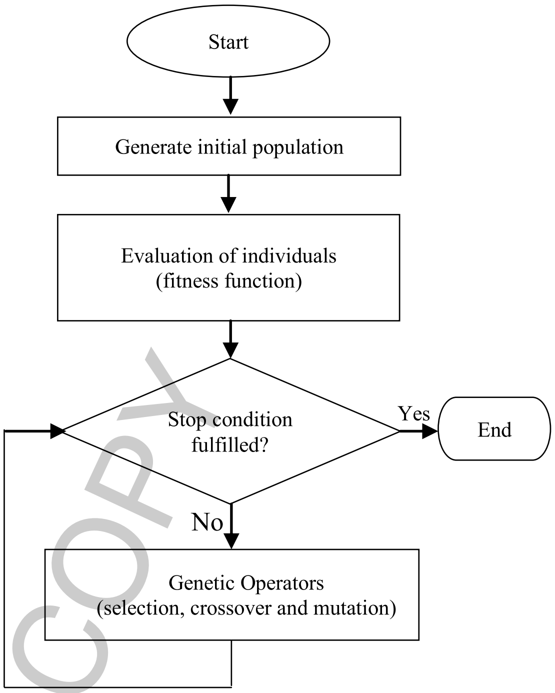
Start
Generate initial population
Evaluation of individuals
(fitness function)
Stop condition
fulfilled?
Yes
End
No
Genetic Operators
(selection, crossover and mutation)

Fig. 1. Stages of GA [24, 25].

4. Proposed method

In general, the proposed method is integrated of the steps and procedures of the basic GA of Section 3. The GA proposed evaluates several combinations of sentences in an optimized search space, which are candidates in representing the best extractive summary of one or multiple documents.

4.1. Solution representation

In the proposed GA, the solution is presented using a coding of individuals considering the order of sentences that can appear in extractive summary. Therefore, each individual X_i is represented in a vector of n positions $[P_1, P_2, \dots, P_n]$, where each position includes a set of sentences $\{S_1, S_2, \dots, S_n\}$ of the original document D, and the union of all the sentences will represent the content of the original document, as shown in Equation (3).

$$\bigcup_{i=1}^{n} S_i = D \tag{3}$$

For each coding to be considered as an extractive summary, the first sentences are considered from a set of words. For example, if we have a document with $n = 10$ sentences and we generate an extractive summary of 100 words with an average of 20 words per sentence, then the position vector can use a sequence equivalent to [1–9] indicating that the possible solution begins with sentences 4 and 1, ending with sentence 9, although only the first 5 sentences will be taken into account to comply with first 100 words as a summary.

4.2. Fitness function

The fitness function is an important stage for the performance of the GA and is the value by which the quality of the summaries is maximized with the passing of $(g + 1)$ generations. To measure the quality of each summary, F-measure maximization based on the co-occurrence of bag-of-words and bigrams evaluated from ROUGE-1.5.5 system was used [5]. The maximum F-measure value of the individual $X_k(g)$ obtained from $X_i(g)$ population determine the best combination of sentences found in GA. This maximization is shown in Equation (4)

$$Max\,(F1\,(X_k\,(g))) = \frac{\sum_{S \in S_{ref}} \sum_{gram_n \in S} Count_{match}\,(gram_n)}{\sum_{S \in S_{ref}} \sum_{gram_n \in S} Count\,(gram_n)} \tag{4}$$

where n determine the size of n-grams for evaluating the text combinations of the source text, $F1$ is the F-measure result of ROUGE system and $Count_{match}\,(gram_n)$ is the number of n-grams that co-occurring between the GA summary and the set of gold-standard summaries.

If the individual $X_k\,(g)$ have the greatest co-occurrence of n-grams from the all generations g of populations $X_i\,(g)$, then it will have the best combination of sentences when obtaining the largest number of retrieved n-grams.

4.3. Population initialization

The most common strategy for initializing the population (when $g = 0$) must be generated with codifications of random real numbers for signature each sentence of the set $D = \{S_1, S_2, \dots, S_n\}$ in each position P_i. Therefore, the first generation of individuals will be according to Equation 6

$$X_c\,(0) = \left[X_{c,1}\,(0),\; X_{c,2}\,(0),\; \dots,\; X_{c,n}\,(0)\right], \qquad X_{c,s} = a_s \tag{5}$$

where a_s represents a real integer number {1, 2, ..., n} that corresponds to the number of selected sentence from the original document D, $c = 1, 2, \dots, N_{pop}$, $s = 1, 2, \dots, n$, and n is the number of sentences of D. Therefore, each sentence has the same probability of being included as part of an extractive summary respecting a number W of requested words as condition, as shown in Equation (6)

$$\sum_{S_i \in Summary} l_i \leq W \tag{6}$$

where l_i is a length of the sentence S_i (measured in words) and W is the maximus number of words allowed for generating an extractive summary.

4.4. Selection

The selection is the GA step that allows to take a set of individuals X_c from a generation g to obtain the greatest fitness values with the purpose of obtain better individuals in $g + 1$ generations.

One of the methods of selection most known of GA is the elitism stage, which has the quality to choose a set of individuals of better aptitude in the generation g to pass to the generation $g + 1$.

According to [26], if we have $Pob\,(g) = \left\{X_1\,(g),\; X_2\,(g),\; \dots,\; X_{N_{pop}}\,(g)\right\}$ as a population of individuals ordered from greater to lesser fitness, then the set of individuals that will be pass to the next generation is $(g+1) = \{X_1(g),\ X_2(g), \ldots,\ X_e(g)\}$ where $E(g+1) \subseteq Pob(g)$, $e < N_{pop}$, and e is a parameter that specifies the number or percentage of individuals to be selected by elitism. However, for the selection of individuals it is required to use at least one other selection operator to maintain N_{pop} individuals for each generation.

To select the remaining individuals from each generation, we propose to generate new offspring from the tournament selection operator by taking several samples of N_{Tor} randomly selected individuals to obtain the best fitness value individual [41], as shown in Equation (7)

$$X_b(g) = Max\left(F1(X_1(g)),\ F1(X_2(g)), \ldots,\ F1\left(X_{N_{Tor}}(g)\right)\right) \tag{7}$$

where $X_b(g)$ is the individual with the best fitness value and $F1$ is the F-measure result of ROUGE system. To integrate the selection stage, we propose to use the elitism operator to choose the best individuals of each generation g, using a percentage of individuals. Finally, the remaining individuals are obtained from the tournament selection operator using samples of 2 and 3 randomly obtained individuals.

4.5. *Crossover*

For the crossover of individuals, we use the cycle crossover operator (CX). This operator has the capacity to generate new offspring from the genetic coding of each pair of parents, considering their hereditary characteristics [41]. For the CX operator to be started, a starting point must be selected for genetic exchange. Therefore, if we have a pair of parents $X_{p1}(g)$ and $X_{p2}(g)$ which represent pairs of parents to cross, then we use a randomly generated starting point to exchange information from both parents and generate a new individual $Y_i(g)$, as shown in Equation (8)

$$Y_{i,s} = \begin{cases} X_{p1,s}(g), & if\ s \leq ptC \\ X_{p2,s}(g) & otherwise \end{cases} \tag{8}$$

where $X_{p1,s}(g)$ represents the parent gene $X_{p1}(g)$, $X_{p2,s}(g)$ represents the parent gene $X_{p2}(g)$ and ptC is an integer value representing a start point selected randomly in a range of $[1, n]$, where n is the size of the individual. To generate a second offspring, the roles of $X_{p1}(g)$ and $X_{p2}(g)$ are exchanged with the first parent being individual $X_{p2}(g)$.

4.6. *Mutation*

Remembering the Equation (2) of the Section 3.1, the mutation stage takes a set of individuals $Y_i(g)$ to generate individuals $Z_i(g)$ modifying some features for each generation g. We used the insertion mutation operator to select a pair of genes of the individual $Y_{i,t}(g)$ and $Y_{i,r}(g)$ randomly to insert the gene $Y_{i,t}(g)$ in the gene $Y_{i,r}(g)$ [1], as shown in Equation (9). Therefore, if the random value *rand* is between the value 0 and P, then the mutation of individuals is performed by insertion operator, otherwise the individual is not modified.

$$Z_{i,s}(g) = \begin{cases} Y_{i,t}(g) = Y_{i,r}(g), \\ Y_{i,t+1}(g) = Y_{i,t}(g), \ldots,\ Y_{i,r}(g) = Y_{i,r-1}(g) \\ if\ 0 < rand \leq P \\ Y_{i,s}(g)\ otherwise \end{cases} \tag{9}$$

where r is the variable that relates the gene to be inserted, the variable t is the target gene to be inserted, which are a subset of numbers $s = \{1,\ 2, \ldots,\ n\}$, and n identifies the sentence number of the document the original set of documents.

4.7. *Replacement of individuals*

For the replacement of individuals, we propose to integrate the set of individuals generated by elitist selection $(E(g+1))$ and the set of individuals $Z_i(g)$ from the mutation stage, to integrate the population of the next generation $X_i(g+1)$, as shown in Equation (10).

$$X_i(g+1) = E(g+1) + Z_i(g) \tag{10}$$

4.8. *Termination criterion*

The termination criterion used to halt GA iterations is determined by several generations established as the execution parameter.

5. Experiments and results

In this section, we present the experiments carried out by the proposed GA to calculate *Topline* of extractive summaries using DUC01 and DUC02 datasets and the performance of some AETS methods and heuristics in the state-of-the-art.

Table 1
Datasets main characteristics

	DUC01	DUC02
Number of collections	30	59
Number of documents	309	567
Number of gold-standard summaries per document	2	1–2
Gold-standard summary length (in words)	100	100

5.1. Datasets

The DUC datasetsare the most used by researchers in the AETS of a document and multiple documents highlighting DUC01 and DUC02. To measure the proposed GA performance, we used DUC01 and DUC02. DUC01 and DUC02 are products of workshops organized by the National Institute of Standards and Technology (NIST) for the development of ATS. The documents that make up these collections are based on news articles from some news agencies such as The Financial Times, The Wall Street Journal, Associated Press and others [11, 27, 28].

DUC01 dataset consists of 309 English documents grouped into 30 collections, each collection containing an average of 10 documents based on news articles addressing natural disaster issues, biographical information, and others [26, 27]. Each original document of DUC01 was assigned two gold-standard summaries generated in abstractive form by two humans, containing approximately 100 words. For the ATS of multiple documents, two abstracts were generated for each collection generating 60 abstract summaries that have lengths of 50, 100, 200 and 400 words [27].

DUC02 dataset consists of 567 news articles in English grouped into 59 collections, each collection contains between 5 and 12 documents dealing with topics of technology, food, politics, finance, among others. Like DUC01, this dataset is mainly used for two tasks, the first is to generate summaries of a document, each document had one or twogold-standard summaries that had a minimum length of 100 words. The second task is to generate summaries of multiple documents. For the AETS of multiple documents, one or two abstracts were generated for each collection, generating 118 abstracts/extracts with lengths of 10, 50, 100, 200 and 400 words [28]. Table 1 shows the general data for each dataset.

5.2. Tuning proposed of GA

For determine the upper bounds of DUC01 and DUC02, different tests were carried out with some adjustments of parameters with the objective of obtaining the best extractive summaries. Table 2 shows the best tuning parameters applied to GA proposed to calculate the best extractive summaries of a document.

Table 2
AG parameters to calculate *Topline* of DUC01 and DUC02 for AETS

G	N_{Pop}	Selection				Crossover	Mutation	
		Operator	e	Operator	N_{Tor}		Operator	P
30	150	Elitism	10%	Tournament	3	CX	Insertion	8

The fitness value of each solution is obtained from the n-gram specification to be evaluated by the ROUGE system. In this paper, the unit of evaluation based on the co-occurrence of bag-of-words and bigrams (ROUGE-1 and ROUGE-2) was used, to compare the performance of the most novel state-of-the-art methods in relation to set of gold-standard summaries [6].

5.3. Comparison to state-of-the-art methods and heuristics

As mentioned in Section 1, the importance of knowing the best extractive summaries consist in determining *Topline* from the extractive summaries of one and several documents and reweight the most novels methods in the state-of-the-art. In this section, we present a performance comparison of the state-of-the-art methods and their advances with respect to performance obtained from the *Baseline-first, Baseline-random* [30] and *Topline* heuristics. The methods and heuristics involved in this comparison are the following:

– ***Baseline-first:*** It is a heuristic that allows to use the first sentences of an original text according to a length of words to present as a summary to the user [13, 30, 49]. The performance of this heuristic generates good results in the AETS. However, this heuristic must be overcome by state-of-the-art methods [14]. The results of this heuristic were reported in [30, 31].
– ***Baseline-random:*** It is a heuristic in the state-of-the-art that selects random sentences to present them as an extractive summary to the user [49]. In addition, this heuristic allows us to determine how significant is the performance of AETS methods are in the state-of-the-art [30, 31, 47]. The results of this heuristic were reported in [30, 31].

- ***Topline:*** It is a heuristic that allows to obtain the maximum value that any state-of-the-art method can achieve due to the lack of concordance between evaluators [14], since it selects sentences considering one or several gold-standard summaries. As mentioned in Section 2, efforts have been made in the state-of-the-art to know the scope of the AETS.
- **TextRank:** *TextRank* is an algorithm based on the weight of graphs to identify the importance of sentences/phrases of a text. This method is an adaptation of Google's PageRank algorithm [40]. The author of [36] proposes the use of *TextRank* to weight those sentences or phrases of greater relevance from an original text. The performance of this algorithm in the AETS of a document and multiple documents has improved the quality of several methods of the state-of-the-art [36–39].
- **GA-Summarization:** García et al. [30] present a method based on the use of a GA to generate extractive summaries. In this paper, emphasis is placed on evaluating each candidate summary from the fitness function. The value of the fitness function is obtained from the use of the following features: Frequency of terms in the document, frequency of terms in the summary and the importance of sentences according to their position from the source text.
- **Sentence features:** Vazquez [12] presents a GA-based method for determining the combination of unsupervised sentence features for the sentence selection step in the extractive text summarization method [47, 49]. The set of features are: Coverage, Sentence Position (some parameters was used from [30]), Sentence Length, and Similarity with the title. The performance of this method is better that other extractive text summarization methods.
- **UnitifiedRank:** Wan [45] proposes the use of a method to generate extractive summaries based on the approaches that involve the AETS of one and multiple documents. This approaches that take these approaches are incorporated (or unified) into a graph-based model to weight the most important sentences and obtain an extractive summary.
- **DE:** The method used by Aliguliyev [34] is based on the generation of extractive summaries based on the clustering of sentences. First, a sentence cluster of the original document is generated, and then the most representative sentences of each cluster are obtained. For this stage, a method based on the differential sentence evolution algorithm was used to determine the most representative sentences in each cluster.
- **FEOM:** Song [44] uses a Fuzzy Evolutionary Optimization Model (FEOM) to generate extractive summaries based on the document clustering. FEOM employs three control operators to regulate the parameter setting in the crossing and mutation stages of individuals to generate better clusters of sentences and obtain the most representative of each to generate an extractive summary.
- **NetSum:** Svore [23] uses an approach based on the use of neural networks to extract a set of features from each sentence and determine its importance of the original document. In the training stage, they used the RankNet learning algorithm to weight each sentence according to its importance to present the best sentences in an extractive summary.
- **CRF:** Unlike several methods of the state-of-the-art, Shen [10] addresses the problem of AETS as a sequential labeling of sentences using Conditional Random Fields (CRF). From this perspective, each document is processed by sequential tagging of sentences and the generation stage of the summary label a sentence with 1 and 0.
- **QCS:** Dunlavy [9] presents a system for Querying, Clustering and Summarizing documents (QCS). In QCS method, the relevant documents are retrieved, and then the documents retrieved are separate into several groups (clusters) of topics and create a single summary for each cluster. To generate an extractive summary, three stages are used, the first uses latent semantic indexing as a retrieval step, then k-means is used for document clustering, finally, and a Hidden Markov Model (HMM) is used to generate an extractive summary for each cluster.
- **SVM:** Yeh [22] proposes the use of two approaches to generate extractive summaries. The first is a Modified Corpus Based Approach (MCBA) to use a combined fitness value based on the analysis of highlighted features, and the use of a GA to determine optimized combinations of features. The second approach uses a Textual Relations Mapping based on Latent Semantic Analysis (LSA+TRM) deriving semantical structures from a document.

- **Manifold Ranking:** Wan [46] proposes to relate all the sentences of documents with the main topic of the original text. The value of each sentence is obtained to measure their contribution with respect to the topic. To measure this set of sentences, they used a greedy algorithm to impose the penalty of diversity in each sentence. The summary is produced by choosing sentences with a level of contribution highly biased and a high information novelty.
- **MA-SingleDocSum:** Mendoza [26] treats AETS task as a binary optimization problem to determine the most important ideas of an original text. They use MAs to optimize the quality of extractive summaries with the use of features such as: position of sentence, length of sentence, and the relationship of summary generated with respect to title.

DE, *FEOM*, *QCS* and *MA-Single DocSum* methods do not participate in the following comparisons, because the sentence segmentation stage is not performed according to DUC01 and DUC02 workshops.

For comparing and reweigh the performance of the methods previously described with the heuristics of the state-of-the-art, we used the evaluation based on the co-occurrence of bag-of-words and bigrams (ROUGE-1 and ROUGE-2) of the ROUGE system [5, 6] using the function of Equation (11) to establish the performance of each state-of-the-art method respect to the best extractive summaries obtained by the proposed GA.

$$ROUGE-N = \frac{\sum_{S\in Summ_{ref}} \sum_{gram_n \in S} Count_{match}\left(gram_n\right)}{\sum_{S\in Summ_{ref}} \sum_{gram_n \in S} Count\left(gram_n\right)} \tag{11}$$

Table 3
Results of ROUGE-1 and ROUGE-2 methods and heuristics on DUC01

Method	ROUGE-1	ROUGE-2
Topline	**59.408**	**33.422**
NetSum	46.427	17.697
CRF	45.512	17.327
UnitifiedRank	45.377	17.646
GA-Summarization	45.120	19.762
Sentence Features	45.058	19.619
SVM	44.628	17.018
Baseline-first	**44.272**	**19.701**
Manifold Ranking	43.359	16.635
TextRank	41.083	14.054
Baseline-random	**36.587**	**11.251**

Table 4
Results of ROUGE-1 and ROUGE-2 methods and heuristics on DUC02

Method	ROUGE-1	ROUGE-2
Topline	**62.367**	**35.742**
UnitifiedRank	48.487	21.462
Sentence Features	48.423	22.471
GA-Summarization	48.277	22.338
Baseline-first	**47.294**	**22.208**
NetSum	44.963	11.167
TextRank	44.320	20.019
CRF	44.006	10.924
SVM	43.235	10.867
Manifold Ranking	42.325	10.677
Baseline-random	**38.817**	**13.395**

Tables 3 and 4 shows the average results of ROUGE-1 and ROUGE-2 when calculating the *Topline* of 309 documents of DUC01 dataset and 567 documents of DUC02 dataset using the GA parameters presented in Table 2. The performance of the state-of-the-art methods are shown in this comparison.

According to the results presented in Tables 3 and 4, *Topline* performance is substantially distant from other state-of-the-art methods, as mentioned by [7, 43]. For DUC01, *Topline* obtained a performance equivalent to 59.408 with ROUGE-1 and 33.422 with ROUGE-2, while the best state-of-the-art methods are *NetSum* obtaining 46.427 with ROUGE-1 and *GA-Summarization* obtaining 19.762 with ROUGE-2. For DUC02, *Topline* obtained a performance equivalent to 62.367 with ROUGE-1 and 35.742 with ROUGE-2, while the best state-of-the-art methods are *UnitifiedRank* obtaining 48.487 with ROUGE-1 and *SentenceFeatures* obtaining 22.471 with ROUGE-2.

A comparison of the level of advance of the most recent state-of-the-art methods is shown in Tables 5 and 6. To determine this performance, we use the formula (12) based on the premise that the performance of Topline heuristic is 100% and *Baseline-random* is 0%.

$$ROUGE-N = \frac{\left(ROUGE-N_{OM}-ROUGE-N_{BR}\right)\times 100}{ROUGE-N_{TL}-ROUGE-N_{BR}} \tag{12}$$

where $ROUGE-N$ specifies the F-measure performance of bag-of-words and bigrams, OM is the performance of other methods, TL is the performance of *Topline* heuristic and BR is the performance of *Baseline-random* heuristic.

Table 5
Ranking of state-of-the-art methods and heuristics for DUC01

Method	ROUGE-1	ROUGE-2
Topline	**100%**	**100%**
NetSum	43.12% (1)	29.07% (3)
CRF	39.11% (2)	27.41% (5)
UnitifiedRank	38.58% (3)	28.84% (4)
GA-Summarization	37.39% (4)	38.39% (1)
Sentence Features	37.12% (5)	37.74% (2)
SVM	35.24% (6)	26.01% (6)
Baseline-first	**33.68%**	**38.11%**
Manifold Ranking	29.67% (7)	24.28% (7)
TextRank	19.70% (8)	12.64% (8)
Baseline-random	**0%**	**0%**

Table 6
Ranking of state-of-the-art methods and heuristics for DUC02

Method	ROUGE-1	ROUGE-2
Topline	**100%**	**100%**
UnitifiedRank	41.06% (1)	36.10% (3)
Sentence Features	40.79% (2)	40.61% (1)
GA-Summarization	40.17% (3)	40.02% (2)
Baseline-first	**36.00%**	**39.44%**
NetSum	26.10% (4)	–9.97% (5)
TextRank	23.37% (5)	29.64% (4)
CRF	22.03% (6)	–11.06% (6)
SVM	18.76% (7)	–11.31% (7)
Manifold Ranking	14.90% (8)	–12.16% (8)
Baseline-random	**0%**	**0%**

Table 7
Improvement percent of *NetSum* to other methods (ROUGE-1)

Method	Improvement obtained by NetSum method (%)
	DUC01
CRF	10.25
UnitifiedRank	11.95
GA-Summarization	15.32
Sentence Features	16.16
SVM	22.37
Manifold Ranking	45.31
TextRank	118.86

The best state-of-the-art method of the Table 5 presents an advance equivalent to 43.12% for ROUGE-1 and 38.39% for ROUGE-2. Therefore, it follows that for the development of the AETS task there is 56.88% for ROUGE-1 and 61.61% for ROUGE-2 to be explored. In the other hand, it is observed that the performance of *Baseline-first* heuristic is better than several methods of the state-of-the-art in F-measure of ROUGE-1 (33.68%), surpassing to *Manifold Ranking* (29.67%) and *TextRank* (19.70%), in F-measure ROUGE-2 (38.11%) is better than *NetSum* (29.07%), *UnitifiedRank* (28.84%), *CRF* (27.41%), *SVM* (26.01%), *Manifold Ranking* (24.28%) and *TextRank* (12.64%) methods.

The best state-of-the-art methods present an advance equivalent to 41.06% for ROUGE-1 and 40.61% for ROUGE-2 (see Table 6). Therefore, it follows that for the development of the AETS task there is a 58.94% for ROUGE-1 and 59.39% for ROUGE-2 to be explored.

In the other hand, it is observed that the performance of *Baseline-first* heuristic is better to several state-of-the-art methods for ROUGE-1 (36.00%) and ROUGE-2 (39.44%). The performance of the *Baseline-first* heuristic remains better than several state-of-the-art methods. However, the methods *Unitified Rank*, *Sentence Features* and *GA-Summarization* are better that this heuristic.

The performance of *Baseline-random* heuristic (0%) was expected to be the lowest in this comparison. However, the methods *NetSum* (–9.97%), *CRF* (–11.06%), *SVM* (–11.31%) and *Manifold Ranking* (–12.16%) show the lowest performance of ROUGE-2 for DUC02.

In general, the Tables 5 and 6, the new reweighting of the state-of-the-art methods is observed. However, it is possible to determine the level of significance of the best methods of each comparison. To perform this determination, we used the ROUGE-1 and ROUGE-2 reweighting in DUC01 and DUC02 based on Equation (13).

$$\frac{Bestmethod - Othermethod}{Othermethod} \times 100 \qquad (13)$$

Tables 7 and 8 presents the results obtained to determine the improvement produced by *NetSum* and *GA-Summarization* methods with respect to the other state-of-the-art methods in F-measure of ROUGE-1 and ROUGE-2 on DUC01 data respectively. In general, the percent of improvement of *NetSum* method is in a range of 10–20 percent to some state-of-the-art methods. In the other hand, *NetSum* method presents an improvement percent greater than *Manifold Ranking* (45.31%) and *TextRank* (118.86%) methods (see Table 7). The percent of improvement of *GA-Summarization* method to other state-of-the-art methods is very distant for ROUGE-2 on DUC01 data. However, the *Sentence Features* method is close to *GA-Summarization* with 1.72% (see Table 8).

Table 9 presents the results obtained to determine the percentage of improvement of *Unitified Rank* method with respect to the other state-of-the-art methods in F-measure of ROUGE-1 on DUC02

Table 8
Improvement percentage of *GA-Summarization* to other methods (ROUGE-2)

Method	Improvement obtained by GA-summarization method (%)
	DUC01
Sentence Features	1.72
NetSum	32.04
UnitifiedRank	33.09
CRF	40.08
SVM	47.58
Manifold Ranking	58.08
TextRank	203.63

Table 9
Improvement percentage of *UnitifiedRank* with other methods on DUC01 (ROUGE-1)

Method	Improvement obtained by the UnitifiedRank method (%)
	DUC02
SentenceFeatures	0.67
GA-Summarization	2.22
NetSum	57.34
TextRank	75.73
CRF	86.36
SVM	118.88
Manifold Ranking	175.66

Table 10
Improvement percentage of *Sentence Features* with other methods on DUC02 (ROUGE-2)

Method	Improvement obtained by the sentence features method (%)
	DUC02
GA-Summarization	1.48
UnitifiedRank	12.50
TextRank	37.00
NetSum	(N/A)
CRF	(N/A)
SVM	(N/A)
Manifold Ranking	(N/A)

data. It is observed that the percent of improvement by *Unitified Rank* is very close to *Sentence Features* (0.67%) and *GA-Summarizarion* (2.22%) methods. Therefore, these improvement percent are not significant. In the other hand, the other methods show a more significant difference exceeding 50%.

Table 10 shows the comparison of the improvement percentage of the *Sentence Features* method in relation to the other state-of-the-art methods in F-measure ROUGE-2 on DUC02 data. The *GA-Summarization* method is not significantly distant, obtaining 1.48%. The other methods present percentages greater than 10%. In the other hand, some methods present the label (N/A), because their performance in Table 6 is less than 0% and therefore is not calculable.

Table 11
Ranking of the state-of-the-art methods

Method	R_r								Resultant rank
	1	2	3	4	5	6	7	8	
GA-Summarization	1	1	1	1	0	0	0	0	3.250
SentenceFeatures	1	2	0	0	1	0	0	0	3.250
UnitifiedRank	1	0	2	1	0	0	0	0	3.125
NetSum	1	0	1	1	1	0	0	0	2.875
CRF	0	1	0	0	1	2	0	0	2.125
TextRank	0	0	0	1	1	0	0	2	1.375
SVM	0	0	0	0	0	2	2	0	1.250
Manifold Ranking	0	0	0	0	0	0	2	2	0.750

To unify all the performances obtained from ROUGE-1 and ROUGE-2 on DUC01 and DUC02, we propose to use the data from the Tables 5 and 6 to show them in a unified ranking of positions, considering the position of each method of ROUGE-1 and ROUGE-2 on DUC01 and DUC02 datasets.

The results of the unification of these results are shown on the Table 11, using the Equation (14) which has been used by [26, 33].

$$Ran\,(method) = \sum_{r=1}^{8} \frac{(8 - r + 1)\, R_r}{8} \qquad (14)$$

where R_r refers to the number of times that the method affects the $r - th$ position. The number 8 represents the total number of methods involved of this comparison.

Table 11 shows that the performance of *GA-Summarization* (3.250) and *Sentence Features* (3.250) methods show the best positions in the method rankings. However, the methods *UnitifiedRank* (3.125) and *NetSum* (2.875) show a good performance with the same results.

In the other hand, the methods based on evolutionary approaches show a good tendency to obtain the best levels of performance compared to the machine learning methods. Therefore, these methods represent a viable solution for generating high performance extractive summaries.

6. Conclusions and future work

The state-of-the-art methods to obtain the upper bounds have been based on exhaustive searches to obtain the best extractive summaries. However, GAs have not been used to obtain extractive summaries from *Topline* heuristic to reweigh the performance of the AETS methods.

In this paper, some GA operators were used to obtain the best extractive summaries. As a fitness function, it was proposed to use ROUGE-N method of ROUGE-1.5.5 system to evaluate the quality of the generated GA combinations.

In the state-of-the-art, the maximum possible performance value of the AETS was unknown. However, it was possible to approximate the best summaries with the use of GAs, to know the scope of the methods of the AETS.

With the determination of the best sets of sentence combinations, it is possible to introduce them to a supervised machine learning model to improve the quality of extractive summaries.

The best state-of-the-art methods of DUC01 show performance equivalent to 43.12% for ROUGE-1 and 38.39% for ROUGE-2 (reported in Table 5). Therefore, it follows that there is still 56.88% for ROUGE-1 and 61.61% for ROUGE-2 to be explored. The best state-of-the-art method of DUC02 show a performance equivalent to 41.06% for ROUGE-1 and 40.61% for ROUGE-2 (reported in Table 6). Therefore, it follows that there is still a 58.94% of ROUGE-1 and 59.39% of ROUGE-2 to be explored.

With the use of AGs and *Topline* heuristic, it was possible to reweigh the AETS methods to obtain more objective results and to generate a rank matrix (reported in Table 11), which shows in general the performance of the state-of-the-art methods.

With the new ranking of the state-of-the-art methods, it was possible to determine the percentages of significant improvement among the best state-of-the-art methods.

In Tables 5 and 6, it is observed that the percentage of significance is much close between several methods of the state-of-the-art, so it will be very important to analyze the quality of the summaries generated by means of a Turing test, to demonstrate if the level of achieved performance of extractive summaries is confounded with summaries created by humans.

References

[1] A.E. Eiben and J.E. Smith, *Introduction to Evolutionary Computing*, 2nd ed., Springer-Verlag Berlin Heidelberg, vol. 12, no. 1995, 2015.

[2] A. Louis, A. Joshi and A. Nenkova, Discourse indicators for content selection in summarization, *11th Annu Meet Spec Interes Gr Discourse Dialogue*, 2010, pp. 147–156.

[3] C. Aone, M.E. Okurowski and J. Gorlinsky, Trainable, scalable summarization using robust NLP and machine learning, *Proc 36th Annu Meet Assoc Comput Linguist* -, vol. 1, 1998, p. 62.

[4] C. Darwin, The origin of species, 1859.

[5] C.Y. Lin, Rouge: A package for automatic evaluation of summaries, *Proc Work text Summ Branches out (WAS 2004)*, no. 1, 2004, pp. 25–26.

[6] C.Y. Lin and E. Hovy, Automatic evaluation of summaries using N-gram co-occurrence statistics, *Proc 2003 Conf North Am Chapter Assoc Comput Linguist Hum Lang Technol - NAACL '03*, vol. 2003, 2003, pp. 71–78.

[7] C.Y. Lin and E. Hovy, The potential and limitations of automatic sentence extraction for summarization, *Proc HLT-NAACL 03 Text Summ Work*, vol. 5, 2003, pp. 73–80.

[8] D. Marcu, Improving summarization through rhetorical parsing tuning, *Proc 6th Work Very Large Corpora*, 1998, pp. 206–215.

[9] D.M. Dunlavy, D.P. O'Leary, J.M. Conroy and J.D. Schlesinger, QCS: A system for querying, clustering and summarizing documents, *Inf Process Manag* **43**(6) (2007), 1588–1605.

[10] D. Shen, J. Sun, H. Li, Q. Yang and Z. Chen, Document Summarization using Conditional Random Fields, 2004, pp. 2862–2867.

[11] DUC, Document Undertanding Conferences. 2002.

[12] E. Vazquez, Y. Ledeneva and R.A. García-Hernández, Sentence Features Relevance for Extractive Text Summarization using Genetic Algorithms. JIFS. (to be published April 2018).

[13] E. Villatoro-Tello, L. Villaseñor-Pineda, M. Montes-y-Gómez and D. Pinto-Avendaño, Multi-Document summarization based on locally relevant sentences, *8th Mex Int Conf Artif Intell - Proc Spec Sess MICAI 2009*, 2009, pp. 87–91.

[14] G. Sidorov, Non-linear construction of n-grams in computational linguistics, 1st ed. México: Sociedad Mexicana de Inteligencia Artificial, 2013.

[15] H. Ceylan, R. Mihalcea, U. Öyertem, E. Lloret and M. Palomar, Quantifying the limits and success of extractive summarization systems across domains, *Hum Lang Technol* (2010), 903–911.

[16] H.P. Edmundson, New methods in automatic extracting, *J Assoc Comput Mach* **16**(2) (1969), 264–285.

[17] H.P. Luhn, The automatic creation of literature abstracts, *IBM J Res Dev* **2**(2) (1958), 159–165.

[18] J. Conroy and D.P. O'Leary, Text summarization via hidden markov models, *Proc 24th Annu Int ACM SIGIR Conf Res Dev Inf Retrieval*, 2001, pp. 406–407.

[19] J.H. Lee, S. Park, C.-M. Ahn and D. Kim, Automatic generic document summarization based on non-negative matrix factorization, *Inf Process Manag* **45**(1) (2009), 20–34.

[20] J. Steinberger and K. Jezek, Using latent semantic analysis in text summarization and summary evaluation, *7th Int Conf ISIM*, 2004, pp. 93–100.

[21] J. Steinberger and K. Jezek, Sentence compression for the LSA-based summarizer, *CEUR Workshop Proc* **180** (2006), 141–148.

[22] J.Y. Yeh, H.R. Ke, W.P. Yang and I.H. Meng, Text summarization using a trainable summarizer and latent semantic analysis, *Inf Process Manag* **41**(1) (2005), 75–95.

[23] K.M. Svore, L. Vanderwende and C.J.C. Burges, Enhancing single-document summarization by combining rank Net and third-party sources, *Comput Linguist*, 2007, pp. 448–457.

[24] L. Araujo and C. Cervigón, Algoritmos Evolutivos: Un Enfoque Práctico, 2nd ed. RA-MA, 2009.

[25] L. Suanmali, N. Salim and M. S. Binwahlan, Genetic algorithm based sentence extraction for text summarization, *Int J Innov Comput* **1**(1) (2011), 22.

[26] M. Mendoza, S. Bonilla, C. Noguera, C. Cobos and E. León, Extractive single-document summarization based on genetic operators and guided local search, *Expert Syst Appl* **41**(9) (2014), 4158–4169.

[27] P. Over and J. Yen, Introduction to DUC01: An Intrinsic Evaluation of Generic News Text Summarization Systems, in *Proceedings of DUC04 Document Understanding Workshop*, 2004, p. 30.

[28] P. Over and W. Ligget, Introduction to DUC02: An Intrinsic Evaluation of Generic News Text Summarization Systems Document Understanding Conferences (DUC), in *Proceedings of DUC04 Document Understanding Workshop*, 2004, p. 48.

[29] P. Ponce, Inteligencia artificial con aplicaciones a la ingeniería, 1st ed. México, 2010.

[30] R.A. García-Hernández and Y. Ledeneva, Single Extractive Text Summarization Based on a Genetic Algorithm, 2013, pp. 374–383.

[31] R.A. García-Hernández, et al., Comparing commercial tools and state-of-the-art methods for generating text summaries, *8th Mex Int Conf Artif Intell - Proc Spec Sess MICAI 2009*, 2009, pp. 92–96.

[32] R. Chettri and U.K. Chakraborty, Automatic text summarization, *Int J Comput Appl* **161**(1) (2017), 5–7.

[33] R.M. Alguliev, R.M. Aliguliyev and M.S. Hajirahimova, GenDocSum+MCLR: Generic document summarization based on maximum coverage and less redundancy, *Expert Syst Appl* **39**(16) (2012), 12460–12473.

[34] R.M. Aliguliyev, A new sentence similarity measure and sentence based extractive technique for automatic text summarization, *Expert Syst Appl* **36**(4) (2009), 7764–7772.

[35] R. Verma and D. Lee, Extractive Summarization: Limits, Compression, Generalized Model and Heuristics, 2017, p. 19.

[36] R. Mihalcea and P. Tarau, TextRank: Bringing order into texts, *Proc EMNLP*, vol. 85, 2004, pp. 404–411.

[37] R. Mihalcea and P. Tarau, A language independent algorithm for single and multiple document summarization, *Dep Comput Sci Eng* **5** (2005), 19–24.

[38] R. Mihalcea, Random walks on text structures, *Comput Linguist Intell Text Process* **3878** (2006), 249–262.

[39] R. Mihalcea, Graph-based ranking algorithms for sentence extraction, applied to text summarization, *Proc ACL 2004 Interact Poster Demonstr Sess* -, no. 4, 2004, p. 20–es.

[40] S. Brin and L. Page, Reprint of: The anatomy of a large-scale hypertextual web search engine, *Comput Networks* **56**(18) (2012), 3825–3833.

[41] S.N. Sivanandam and S.N. Deepa, Introduction to Genetic Algorithms, 2008.

[42] S. Russell and P. Norvig, Inteligencia artificial, 2009.

[43] W.M. Wang, Z. Li, J.W. Wang and Z.H. Zheng, How far we can go with extractive text summarization? Heuristic methods to obtain near upper bounds, *Expert Syst Appl* **90** (2017), 439–463.

[44] W. Song, L. Cheon Choi, S. Cheol Park and X. Feng Ding, Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization, *Expert Syst Appl* **38**(8) (2011), 9112–9121.

[45] X. Wan, Towards a Unified Approach to Simultaneous Single-Document, *Proceeding 23rd Int Conf Comput Linguist (Coling 2010)*, 2010, pp. 1137–1145.

[46] X. Wan, J. Yang and J. Xiao, Manifold-ranking based topic-focused multi-document summarization, *IJCAI Int Jt Conf Artif Intell*, 2007, pp. 2903–2908.

[47] Y. Ledeneva, A. Gelbukh and R.A. García-Hernández, Terms derived from frequent sequences for extractive text summarization, *Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)*, vol. 4919 LNCS, 2008, pp. 593–604.

[48] Y. Ledeneva, R.A. García-Hernández and A. Gelbukh, Graph Ranking on Maximal Frequent Sequences for Single Extractive Text Summarization, *Int Conf Intell Text Process Comput Linguist*, vol. 8404, 2014, pp. 466–480.

[49] Y. Ledeneva and R.A. García-Hernández, Generación automática de resúmenes. Retos, propuestas y experimentos. Automatic Generation of Text Summaries. Challenges, Proposals and Experiments, Universidad Autónoma del Estado de México, Toluca, 2017.

[50] Y. Meena and D. Gopalani, Evolutionary algorithms for extractive automatic text summarization, *Procedia Comput Sci* **48**(C) (2015), 244–249.