Journal of Intelligent & Fuzzy Systems 35 (2018) 353–365
DOI:10.3233/JIFS-169594
IOS Press

Sentence features relevance for extractive text summarization using genetic algorithms

Eder Vázquez*, René Arnulfo García-Hernández* and Yulia Ledeneva*
Autonomous University of the State of Mexico, Instituto Literario, Toluca, State of Mexico, Mexico

Abstract. Preprocessing, term selection, term weighting, sentence weighting, and sentence selection are the main issues in generating extractive summaries of text sentences. Although many outstanding related works only are focused in the last step, they show sophisticated features in each one. In order to determine the relevance of the sentences (sentence selection step) many *sentence features* have been proposed in this task (in fact, these features are related to all the steps). Recently, some good related works have coincided in the same features but they present different ways for weighting these features. In this paper, a method to optimize the combination of previous relevant features in each step based on a genetic algorithm is presented. The proposed method not only outperforms previous related works in two standard document collections, but also shows the relevance of these features to this problem.

Keywords: Extractive text summarization, genetic algorithms, sentence feature selection, fitness function

1. Introduction

In the last decade, a tremendous growth of digital information finds new forms of representations as computer text files, multimedia or web pages. The current information size availablein the Web means that users do not have the time to process the entire set of information stored there, whereby a lot of relevant and interesting information is wasted [15]. Therefore, novel tools, methods and models that allow us to automatically generate summaries are needed [14].

A summary is a set of phrases or sentences that best covers the relevant concepts of a documents [16]. Specifically, it is a reductive transformation of the content of a input document by the selection or generalization of the most important information in the document [51].

Automatic text summarization is a technique, where a computer is responsible for summarizing a text from the internal representation of an input document [32]. There are several classifications methods for automatic text summarization, but, according to the output summary, the main approaches are divided into: abstractive and extractive [19]. Abstractive summaries could create new phrases (not contained in the original text) that best describe the content of the original text [21]. Extractive text summaries only select the most important words, phrases or paragraphs from the source document to conform the output [21, 43]. Despite the lack of coherence between sentences in an extractive summary, it has been extensively investigated because it is more objective without presenting points of view.

Preprocessing, term selection, term weighting, sentence weighting, and sentence selection are the main issues in generating extractive summaries of text sentences. Although many outstanding related works only are focused in the last step, they show sophisticated features in each one. In order to determine the relevance of the sentences (sentence selection step) many *sentence features* have been proposed in this task (in fact, these features are related to all the steps).

There are methods that use effective features for extractive text summarization such as Similarity to title, Sentence position, Sentence length, Term

*Corresponding authors. Eder Vázquez, René Arnulfo García-Hernández and Yulia Ledeneva, Autonomous University of the State of Mexico, Instituto Literario, #100, Toluca, 50000 State of Mexico, Mexico. E-mails: eder2v@hotmail.com (E. Vázquez), renearnulfo@hotmail.com (R.A. García-Hernández) and yledeneva@yahoo.com (Yulia Ledeneva).

weight, Sentences cohesion, Coverage, Proper Noun, and others; whether each feature has it unique contribution, although, most of these features have been individually investigated [59]. Other studies [13, 34, 61] report the use of several features to extractive text summarization through combining features.

According to [12], extractive text summarization task can be either supervised or unsupervised. A supervised method uses features externally provided by a supervisor for a machine learning could learn how to extract sentences. In some cases, the external features are a large training set of human-generated summaries from the original documents. In other cases, the terms or the sentences are weighting using external features as taxonomies, key words, thesaurus, etc. [59]. Otherwise, unsupervised methods generate the summaries using only the original document. Unsupervised method does not require training data or a specific thesaurus, and can generate good summaries with new document without a prior adjustments [44].

On one hand, supervised extractive text summarization methods use machine learning algorithm like Support Vector Machine [39], Naïve Bayes classification, Neural Networks, and Decision Trees [11]. On the other hand, unsupervised methods use search and optimization algorithms to find patterns in the structure of the text like Clustering, Hidden Markov Model [60], and Genetic Algorithms [34].

In general, some years ago, supervised methods performed better results for extractive text summarization. However, nowadays, unsupervised methods have been reporting relevant results [59]. Supervised methods are dependent from the domain, author writing style, region and time of the document. Hence, they need a new external thesaurus and retraining time [44].

Recently, some researches [13, 34, 42] view extractive text summarization task as a combinatorial optimization problem where one or more objective functions are formulated for optimization. In this case, the objective functions are the features extracted of the documents, and each feature is weighted [42]. In particular, the genetic algorithm has been used to select the best sentences (summary) from a document that maximize the objective function. Also, in these works, the importance of the features is not relevant.

In this paper, a genetic algorithm to generate extractive summaries using objective functions based on new unsupervised features is presented. Additionally, it is possible to determine the relevance of the featureswith our method.

The rest of the paper is organized as follows: Section 2 introduces theoretical fundaments and work related of the automatic extractive text summarizations. Section 3 describes the proposed method based on genetic algorithms. In Section 4 the experimentation and analysis of results are presented; and finally, Section 5 presents conclusions and future works.

2. Background and related work

First, in this section, the main steps for automatic generation of extractive summaries are presented. In the second section, a brief description of the most important text features is presented.

2.1. Extractive text summarization steps

According to [27, 28], a typical extractive text summarization method consists in 5 steps: preprocessing, term selection, term weighting, sentence weighting and sentence selection.

Preprocessing step depends on the process in next steps because prepares a text into a structure representation. Sometimes the document needs standard transformation to obtain only the text without labels of format. In general, words or punctuation signs that do not contribute to any steps are eliminated. In particular, the so-called stop-words without meaning are eliminated, it means, prepositions, conjunctions, adverbs, etc. In order to find frequent or relevant concepts, the words are reduced to an approximate root using stemming algorithms. Generally, Porter stemmer [41] is used for this purpose.

A good extractive summary often rely in the inclusion of two important aspects: relevance (include sentences of the original text that are important) and non-redundancy (the sentences selected cannot be duplicated content) [17, 44]. Thus, all steps must consider these two important aspects.

Term selection step, here, one should decide what type and size of units of text are considered as basic terms. For instance, they can be words, *n-grams* or phrases.

Term weighting is an important step in extractive summaries, because it is used to adjust the importance of each term selected, that is, assigns a numeric value (weight) to each term.

Sentence weighting is the process of assigning a numerical value of usefulness to each sentence. Normally, the weights of the terms are used to estimate the usefulness of a sentence, but exists features like sentence position that determines the relevance of the sentences without using the terms.

Sentence selection is the step in which the final decision is reach out which sentences are selected as summary. In this step, both the most weighted sentences and those that produce non-redundant summaries must be selected.

Since the summary is generated from the sentence selection step, most of the related works only propose so-called *sentence features*. Nevertheless, these sentence features are based on the above mentioned sub-steps.

Even though, exists supervised dependent-linguistic features like Proper Noun [23, 33, 45], Cue-Phrase [8, 17, 33], Pronouns [17, 45], Thematic Words [1, 17, 26, 33, 47], Anaphor [56], Discourse Markers [10, 17, 45]; the unsupervised features have been presented in outstanding related works.

2.2. Unsupervised sentence features

Statistical and surface-level sentence features have been showed competitive results in extractive text summarization [11, 14, 34, 37]. Some of these features are:

- similarity to title [9, 17],
- sentence position [6, 9, 13, 20, 40],
- sentence length [1, 11, 17, 34, 38],
- sentences cohesion [17, 34],
- term length [2],
- term frequency [10, 24, 32],
- term weight [2, 17, 32, 38],
- numbers in sentence [1, 11],
- coverage [13, 34], etc.

A linear combination of 31 sentence features for extractive text summarization is presented in [31], but the results in English does not outperform the baseline. However, according to [13] using only sentence position and coverage features is enough to get good results. Other competitive work is present by [34] where similarity with the title, sentence position, sentence length, cohesion and coverage are used as sentence features. Since, the works of [13] and [34] present new ways of weighting the sentences and the terms, in the remainder of this section, such sentence features are described below.

Similarity with the title obtains a weighting of the sentence according to the similarity with the document title because contains it relevant words that can be taken as unsupervised keywords. Some similarity measures have been proposed, for mentioning some: Cosine, Euclidean, Dice, Jaccard, recently Soft Cosine [48], and other measures. However, normally these measures depend on term selecting and weighting steps. Specifically, [34] uses the classical cosine similarity as term weighting and 1-grams (words) as term selection, described in the Equation (1):

$$\mathrm{RT_s} = \sum\nolimits_{\forall \mathrm{S_i} \in \mathrm{Summary}} \frac{\mathrm{sim_{cos}(S_i,\ t)}}{0} \quad (1)$$

$$\mathrm{RTF_s} = \frac{\mathrm{RT_s}}{\max\limits_{\forall \mathrm{Summary}} \mathrm{RT}} = \delta \quad (2)$$

where $sim_{\cos(S_I,t)}$ is the cosine similarity of sentence S_i with the title t, O is the number of sentences in the summary, RT_s is the average of the similarity of the similarity in the summary S with the title, $max_{\forall Summary} RT$ is the average of the maximum values obtain from the similarities of all sentences in the document with the title (that is the average top greater O similarities of all sentences with the title), and RTF_s is the similarity factor of the sentences of the summary S with the title, and is calculated by Equation (2).

Sentence position gives more relevance to the first positions inside the document because the relevant information tends to appear in specific sections on it [30]. In fact, extractive text summarization baseline is an heuristic where the first sentences (baseline: *first*) of a document can be considered as a good summary [36]. *Baseline: first* is good sentence feature as it is difficult to outperform other methods.

Sentence position is the most studied feature in extractive text summarization [6, 9, 13, 20, 40], where different ways for weighting have been proposed, for example, the inverse order of the sentences [6], $\sqrt{1/number\ of\ sentences}$, etc. The problem of use the inverse order as sentence weighting is that, for example, with a 30-sentence text, the first sentence will be 30 times more important that the last one. It makes almost impossible that the last sentence could appear in the summary.

In [13] is proposed to make this difference softer using the linear equation with slope t, if t is -1 then it can be measured the sentence position as in [49, 53], and if t is 0, it will give the same relevance to each sentence. For a text with n sentences, if the sentence i is selected for the summary then its relevance is defined as: $t\,(i - x) + x$, where $x = 1 + (n - 1)\,/2$ and t is the slope for tuning. In order to normalize the sentence position measure (δ), showed in Equation (3), it is calculated the relevance of the first k

sentences, where k is the number of selected sentences. Note that sentence position feature does not need term selection or term weighting steps.

$$\beta = \frac{\sum_{|C_i|=1}^{n} t\,(i-x)+x}{\sum_{j=1}^{k} t\,(j-x)+x},\ x = 1 + \frac{(n-1)}{2} \quad (3)$$

Sentence length gives more relevance to large sentences, under the idea that short sentences contain less information. The relevance sentence weighting with the sigmoid function [18] is calculated in [34], which is normalized with the longest sentence in the document, described in Equation (4):

$$\gamma = \sum_{\forall \mathrm{s_i} \in \mathrm{Summary}} \frac{1 - \mathrm{e}^{\frac{-\mathrm{l(s_i)}-\mu(\mathrm{l})}{\mathrm{std(l)}}}}{1 + \mathrm{e}^{\frac{-\mathrm{l(s_i)}-\mu(\mathrm{l})}{\mathrm{std(l)}}}} \quad (4)$$

where $l\,(s_i)$ is the length of sentence S_i (measured in words), $\mu\,(l)$ is the average length of the sentence of the summary, and $std\,(l)$ is the standard deviation of the lengths of the sentences of the summary.

Term length has been used for weighting a term according to its length in chars under the idea the longer the term, the more meaning it has. In this sense, the stop-words that are considered empty words normally are the shorter terms. Term length weighting is defined as the number of characters of the term:

$$\mathrm{TL\,(t)} = \mathrm{LChar\,(t)} \quad (5)$$

Coverage is a feature for sentence selection and sentence weighting steps. This feature measures the coverage similarity between the resultant summary and the source document. A new way to weighting this feature is presented in [34, 58], but they address that it was not relevant according to their results. In change, a novel way for weighting the coverage that was useful in their results is presented in [13]. In this case, the selected sentences for making up the summary must contribute to bring relevant and non-redundant information. For this, the summary must contain all its different words and the most frequent words of the original document.

Coverage feature selects sentences based on the f-measure which is an information retrieval evaluation. The f-measure is considered a harmonic balance of recall and precision measures. Usually in information retrieval, precision is defined as the number of correctly recovered units divided by the number of recovered units; and recall is defined as the number of correctly recovered units divided by the number of correctly units. In this way, precision measures the fraction of retrieved units that are relevant, while recall measures the fraction of relevant instances that are retrieved. However, for generating a summary (S), the maximum-words threshold (m) of a summary is considered. Consequently, the number of recovery units always is limited by the maximum-word threshold. Therefore, the final summary must have, for one side, the most relevant words of the original text (T) and, for the other side, it must not be redundant.

The relevance of a word w is represented by the term frequency of the word in the original text ($frequency\,(w, T)$). An expressive word is represented if only are considered the different wordsthat the summarycan have ($\{\mathrm{word}\ \in S\}$). In this sense, the best summary shouldcontain the most frequent words with respect to the original text and each word must be different. In order to have a normalized measure the sum of the frequencies of the different words in the summary is divided by the sum of the frequencies of the most frequent words with respect to the original text. Equation (6) describes the *coverage feature* (α):

$$\alpha = \frac{\sum_{\mathrm{p}=\{\mathrm{word}\ \in S\}}^{m} frequency\,(p, T)}{\sum_{\mathrm{q}=\{\mathrm{word}\ \in T\}}^{m} frequency\,(q, T)} \quad (6)$$

Sentences cohesion is a feature that determines the degree of relatedness of all sentences that make up the summary. The idea of this feature is that the sentences of the summary should be relatedness, describing the same topic [34]. However, it is not considered in our work, because according to [34] it is not relevant in this problem.

In this paper, we are interested in combining the above unsupervised mentioned features because they present new relevant ways in all weighting extractive text summarization steps (PP = Preprocessing, TS = Term Selection, TW = Term Weighting, SW = Sentence Weighting, SS = Sentence Selection), see Table 1.

Although the above features are mentioned as "sentence selection" features in the related work, it is possible to observe in which steps actually affect in Table 1. In fact, only the coverage feature [13] works in the sentence selection step. For this reason, it is

Table 1
Relation of extractives text summarizations steps [27, 28] and new unsupervised features presented in [13, 34]

Features\Steps	PP	TS	TW	SW	SS
Similarity with the title	✓	✓	✓	✓	
Sentence position				✓	
Sentence length	✓	✓	✓	✓	
Term length	✓	✓	✓		
Coverage		✓	✓	✓	✓

needed other method for combining these features in order to determine the best selection set of sentences that produce a better summary.

2.3. Combining unsupervised features

In the last decade, several approaches for automatic text summarization based on extractive ideas have been proposed. Some of this approaches use machine learning techniques [10, 60] and optimization techniques [5, 11]. Between the optimization approaches the evolutionary approaches have raised for producing good extractive text summarization results. Some evolutionary approaches [49, 53] uses a genetic algorithm for the text summarization task based on attribute sentence selection in a supervised classification scheme [53]. These approaches need to account with a previously set of golden summaries for training.

Other methods have used different evolutionary approaches like memetic algorithms [34] and genetic algorithms [13, 31] for linearly combining the sentence selection features. In specific, for n sentence selection features the fitness function is defined as:

$$fitness = \sum_{i=0}^{n} w_i m_i, \quad \text{where} \quad \sum_{i=0}^{n} w_i = 1 \tag{7}$$

w_i represents the associated weight to the sentence feature m_i.

In [31], an approach for extractive text summarization based on the linear optimization of several sentence features is presented. A genetic algorithm is used to find the optimal weighted linear combination of 31 statistical sentence features. The method, called MUSE, is evaluated on two languages: English and Hebrew. However, the obtained results do not overcome the *baseline: first* heuristic. In addition, since the weights are not normalized, there is not possible to know which features are stronger. In [53], the same problem is presented when a similar Equation (7) is applied because there is not restricted the weights values for 8 sentences features.

In [13, 34], an evolutionary algorithm is used to optimize a fitness function similar to the described in Equation (7). In particular, in [13] a genetic algorithm is used for a single extractive text summarization approach, where all parameters used for the genetic algorithm are automatically calculated considering the structure of the original text. In this case, the relevance of the used sentence features (sentence position and coverage) are equal, it is the contribution to improve results of each feature is 50%. Their results showed that the genetic algorithm proposed is competitive in the state of the art.

In [34], an extractive text summarization method for single documents based on memetic approach guided by local searchis proposed. In this work, the sentence position, similarity with the title, sentences length, cohesion and coverage features are linear combined with Equation (7). Thus, according to [34] the most important feature is sentence position, similarity with the title and sentence length.

In this paper, a method to combine a new extractive text summarization sentence features based on genetic algorithm is presented. The main objective of this work is to determine the relevance of the sentence features and how these features helps to composed good summaries to be relevant and not contains redundant information.

3. Proposed genetic algorithm

In this section, the basic steps for the proposed genetic algorithm to automatic text summarization are presented.

3.1. Genetic algorithm steps

Genetic Algorithm (GA) is the most traditional evolutionary technique that has proved to be an alternative solution for an optimization problem. This algorithms are based on the principle of evolution and heredity of characteristic, in which, each generation the stronger individuals (chromosomes) survive and its characteristics are inherited of its descendants, thus, new generations could be formed by stronger individuals in contrast to their ancestors [55]. The first basic step of a genetic algorithm is the *initial population step,* in which the genetic algorithm generates a population of random solutions at the problem. The initial population of chromosomes is then evaluated according to the objective function to be optimized, called *fitness function step*. All chromosomes in the population have different fitness values, some better than other, thus considering likely the best solutions a *parent selection step* is applied. Once two parents have been selected, the *crossover step* is applied. The algorithm proposes a new population mixing some parts from the canonical codification (*chromosome encoding*) of these parents. Eventually, the way of mixing some parts from the canonical codification could produce repeated solutions. The *mutation step* applies randomly small variations to the canonical codification in some individuals of the new population, to explore new solutions. Then, at the new population, the *fitness function step* is applied, and the process of genetic algorithm is repeated until a satisfactory solution is reached or until some arbitrary stop-criteria is reached (*stop condition*).

3.2. Genetic algorithm to combine text summarization features

Below, it is explained how the proposed genetic algorithm performs optimization based on text features to generate the text summaries.

Before of applied the genetic algorithm proposed to generate extractive summaries, the source text document must be adapted to it.

Preprocessing. Before the original text could be used for the genetic algorithm, it is needed to adapt the entry of the original text to the format of it. In this step, all stop words are removed and Porter Stemmer algorithm is applied. Also, all non-alphanumeric symbols are removed.

Chromosome Encoding. One way to encoding the sentences extracted to genetic algorithm is consider a vector of n elements, according to number of sentences. One of the most used encoding values for chromosome is a binary representation. Thus, to represent the genes of a chromosome (C) with a vector of length n of binary values (C_n), where the C_i gene corresponds with the i-th sentence in the text. Each gene in the chromosome can be a binary value, 1 if the i-th sentence is included in the summary, 0 otherwise.

Initial Population. After the chromosome encoding is configured, it is possible to create the first generation considering some parameters. Each gene can take a binary random value ($C_{i=1\ldots n} = Random[0, 1]$). However, if a sentence is selected to appear in the summary ($C_i = 1$), then the number of words of the i-the sentence is summed to the number of words in the summary. To guarantee that each sentence could be selected for the summary, there are created n number of chromosomes in the initial populationand in each one a different gene is arbitrary set to 1.

Fitness function. One of the key steps of a genetic algorithm is the fitness function. In this case, the fitness function is based on Equation (7) because it allows to determine, in a linear combination, the relevance of each measure. In this case, the *sentence position,* the *sentence length, similarity with title* and *coverage feature* ideas showed in [34] are used.

One of the problem with the *sentence length* feature base on the number of words is that several sentences have the same measure. One way to increment the granularity (ability to distinguish two similar objects) of the sentence length is to use the *term length* weighting [2], described in Equation (5), that it is based on the number of characters. Therefore, the new *sentence length* feature is described in Equation (8):

$$\gamma = \sum\nolimits_{\forall s_i \in \text{Summary}} \frac{1 - e^{\frac{-TL(s_i)-\mu(l)}{std(l)}}}{1 + e^{\frac{-TL(s_i)-\mu(l)}{std(l)}}} \tag{8}$$

where $TL\,(s_i)$ is the length of sentence S_i (measured in characters), $\mu\,(l)$ is the average length of the sentence of the summary, and $std\,(l)$ is the standard deviation of the lengths of the sentences of the summary.

Others considered features are the similarity with the tile (RTF) described in Equations (1 and 2); sentence position described in Equation (3), sentence length (8) and coverage Equation (6). Therefore, the fitness function for this problem is based on Equation (7). Thus, the fitness function used in this work will be maximized and satisfy the Equation (9).

$$fitness = w_1\alpha + w_2\beta + w_3\gamma + w_4\delta \tag{9}$$

The fitness function showed in Equation (9) can be seen as a multi-objective function, where w_1, w_2, w_3, w_4 are coefficients to each objective function.

A classical approach to solve a multi-objective optimization problem is to assign a weight w_i to each normalized objective function ($\alpha, \beta, \gamma, \delta$) to converted the multi-objective problem to a single objective problem with a scalar objective function [25]. All weights w_i used are summed up to one, that in Equation (10).

$$\sum_{i=1}^{N} w_i = 1 \tag{10}$$

This approach is a multi-objective function based on weighting sum. Since the user is expected to provide the weights to each single objective [52] is called *priori approach.*

Parent selection. Selection operator is based on the fitness of the individuals. The evolution principle establishes that normally if two good solutions are crossing it could produce better solutions; nevertheless, in some cases the solution could be worse.

In this step, the roulette selection is used because gives more probability of being selected to the parent that have a greater fitness value. In this way, the worst chromosome has the possibility of being selected, although it is slight probable. Generally, only two chromosomes are selected to undergo parent selection.

Crossover. Classical crossover operators as n-point crossover does not work properly because the new child chromosome could represent a summary with more or less words than the user specified. Therefore, the new chromosome is created choosing randomly the genes from both parents, but considers only those with value 1. In this way, if the C_i gene has a value of 1 in both parents, it has more probability of being selected for the child chromosome. For automatic text summarization task, in each time a gene in the child chromosome is selected the minimum number of words for the summary is reviewed.

Mutation. Typically, mutation operator selects a random gene of the individual chromosome and replaces the corresponding gene by other information. In binary codification, the classical *inverse mutation operator* inverts the binary value of a randomly selected gene. In this step, the invert operator is applied twice to the child chromosome, but the first time only the genes with value 1 are considering for invert the value. If the number of words covers the minimum specified by the user, the mutation finishes. Otherwise, in the second time, only the genes with value 0 are considering for invert the value. If the number of words in the summary does not cover the minimum number of words specified by the user, another gene with value 0 is inverted. This process continues until the number of words specified by the user is satisfied.

Stop condition. The GA runs until it reaches the maximum number of generations for each document, which depends on its number of sentences (NS) and the number base (NG), described in Equation (11).

$$max\_generations = \sqrt{4 \times NG \times NS} \qquad (11)$$

In this way, all parameters needed by the GA are calculated automatically considering the structure of the original text. Also, it is important to mention that the sentences of the final summary are extracted from the source document according to the encoding of the best chromosome.

4. Experimentation

In the first section, the dataset and the evaluation methods used for the experiments are presented. In the second section, the experimentation for tuning some parameters of GA and the fitness function are presented. Finally, the results of our proposal are compared to other related works.

Datasets. It is a standard practice to run an algorithm over a standard corpus used in text summarization task that contains the source documents with their summaries created by humans [22, 46]. The most commonly corpus used to evaluate text summarization algorithms are the ones published by the Document Understanding Collection (DUC) and Text Analytics Conferences (TAC) [7]. While TAC[1] reports 31 publications related to the use of its specific corpus for text summarization task, DUC[2] reports 217 publications. Therefore, the DUC collections continue to be the state-of-the-art reference for the text summarization task.

The most used collections, DUC01 and DUC02, are used for the experimentation. DUC01 contains 309 news articles in English, where each one has the golden summaries created by two different people. DUC02 contains 567 news articles in English of different lengths and different topics. Also, the two gold standard summaries were created by two human experts. In both collections, the associated summaries have a length of about 100 words. It is worth mentioning, that the DUC collections are already divided by sentences with the aim of only deciding which selection of sentences produces the best summary.

Evaluation procedure. The ROUGE evaluation toolkit [29] is used to evaluate our results because it has a highly correlation with human judgments. It compares the summaries generated by a system to the human-generated (gold-standard) summaries. For comparison, it uses n-gram statistics. Our evaluation is done using n-gram (1, 1) setting of ROUGE, which was found to have the highest correlation with human judgments, namely, at a confidence level of 95%. ROUGE evaluates the f-measure that is a balance (not an average) of recall and precision results. The results are presented for ROUGE-1 and ROUGE-2 metrics to 100 words.

Tuning AG parameters. As mentioned before, the stop condition depends on the number of generations

[1] https://tac.nist.gov/publications/referring_pubs.html

[2] http://www-nlpir.nist.gov/projects/duc/referring_pubs.html

Table 2
F-measure results of our proposed approach for the DUC01 collection varying the slope

Evaluation	t					
	–0.625	–0.70	–0.80	–0.85	–0.90	–0.95
ROUGE-1	0.44661	0.45032	0.44936	0.44822	0.44803	0.44924
ROUGE-2	0.19221	0.19644	0.19584	0.19456	0.19437	0.19502

Equation (11) which uses the *NG* parameter set to 30 in all the documents in both collections.

Tuning fitness function parameters. Regarding the fitness function, showed in Equation (9), four weights (w_1, w_2, w_3, w_4) are necessary for this propose: w_1 represents the weight associated to *coverage* feature (α); w_2 represents the weight associated to the *sentence position* feature (β); w_3 represents the weight associated tothe *sentence length* feature (γ); and w_4 represents the weight associated to *similarity with title* feature (δ). For each experiment, different values of importance to each feature were tested, and each set of values was manually entered as parameter to the algorithm. Another important parameter for tuning, is the slope *t* for the sentence position.

According to [13] the best slope for the sentence position feature is $t = -0.625$ only with the DUC02 collection. Using this value for the experimentation, we found that the next weights obtain good results in both collections: $\alpha = 0.59$, $\beta = 0.36$, $\gamma = 0.02$, $\delta = 0.03$. However, the slope $t = -0.625$ was tuning only for the DUC02 collection. Thus, in order to find a better slope for both collections, the slope is varied with –0.70, –0.80, –0.85, –0.90 and –0.95 values.

Table 2 shows the obtained results ROUGE-1 and ROUGE-2 for the DUC01 collection. The behavior of the values generated by evaluating the results with ROUGE-1 and ROUGE-2 is shown in Figs. 1 and 2 respectively.

Table 3 shows the obtained results ROUGE-1 and ROUGE-2 for the DUC02 collection. The behavior of the values generated by evaluating the results with ROUGE-1 and ROUGE-2 is shown in Figs. 3 and 4 respectively.

In Tables 2 and 3, it is possible to see that the best slope is $t = -0.70$, where for DUC01 obtains the f-measure of 0.45058 and for DUC02 obtains the f-measure of 0.48423 for ROUGE-1 evaluation, whereas for ROUGE-2 results are 0.19644 for DUC01 and 0.22471 for DUC02 collection.

4.1. Comparison to related works

The best results obtained from our proposed method are compared to other approaches that have

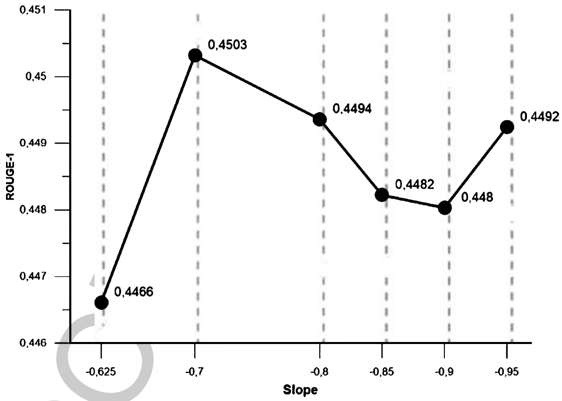

Fig. 1. ROUGE-1 results of our proposed approach for the DUC01 collection varying the slope.

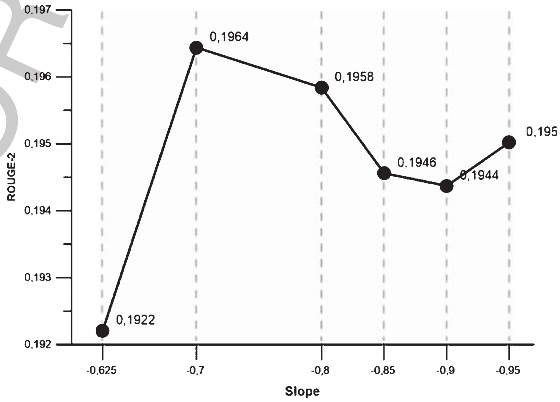

Fig. 2. ROUGE-2 results of our proposed approach for the DUC01 collection varying the slope.

used DUC01 and DUC02 collections, and ROUGE-1 and ROUGE-2 evaluations. Such approaches are briefly described in the next:

– UnifiedRank [57] is a method that proposes a novel unified approach to simultaneous single-document and multi-document summarization, which uses a graph-based representation.
– DE [3] is a summarization approach based on clustering sentences. Use a discrete Differential Evolution algorithm to optimize the objective function, selecting representative sentences of each cluster. Indicates that summarization

Table 3
F-measure results of our proposed approach for the DUC02 collection varying the slope

Evaluation	t					
	–0.625	–0.70	–0.80	–0.85	–0.90	–0.95
ROUGE-1	0.47946	0.48423	0.48032	0.48126	0.48274	0.48206
ROUGE-2	0.22016	0.22471	0.22115	0.22118	0.22399	0.22198

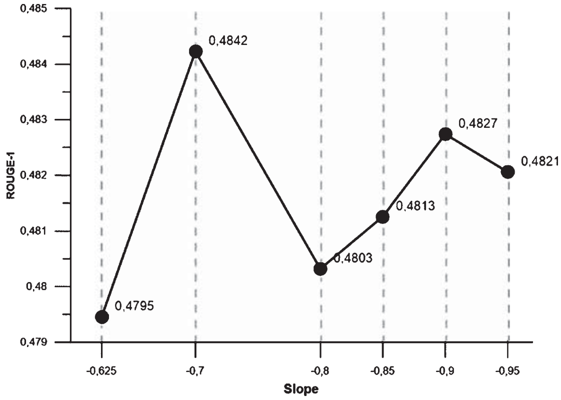

Fig. 3. ROUGE-1 results of our proposed approach for the DUC02 collection varying the slope.

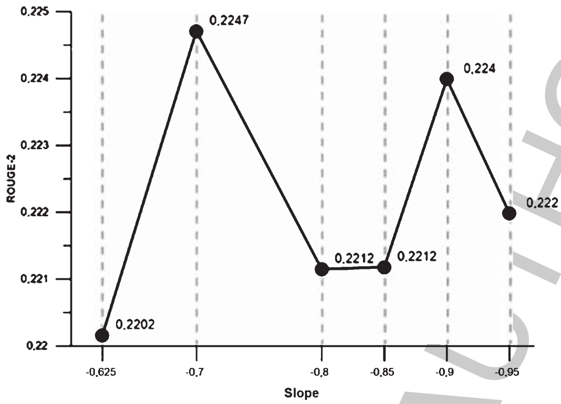

Fig. 4. ROUGE-2 results of our proposed approach for the DUC02 collection varying the slope.

results not only depend on optimized function, also depends on a similarity measure.

– FEOM [50] proposes a Fuzzy Evolutionary Optimization Model. In this approach, sentences are categorized in terms of its content, and after the most important sentence are selected for each cluster.
– NetSum [54] is an approach that use the RankNet learning algorithm to train a pair-based sentence ranker and score every sentence in the document and so identify the most important sentences.
– CRF [47] proposes a framework that take output of previous methods as features and seamlessly integrate them. Treat the summarization task as a sequence of labeling problem. The framework is based on Conditional Random Fields.
– GA [13] proposes a Genetic Algorithm to extractive summarization, where the parameters of the genetic algorithm are calculated automatically from sentence number of each text in a collection. Also, it proposes 2 sentence features: sentence position (slope based linear equation) and term frequency (f-measure based). Results showed are better than other state-of-the-art works.
– MA-SingleDocSum [34] proposes a memetic algorithm for extractive summarization based on genetic operators and guide local search. However, the MA-SingleDocSum method is excluded of the comparisons since it resolves a different problem to the defined in the original task by NIST (the problem of this paper). The difference resides that in the original DUC01 and DUC02 collections the documents are already divided in sentences and the problem is to find the subset of sentences that are more like the golden summaries. In contrast to MA-SingleDocSum method they applied in the preprocessing step a statistics tool for divided the text in sentences, which produce that an original sentence could have more than one sentence or, by the contrary, that two or more original sentences are grouped into one. Therefore, it is not a fair comparison.

Other method optimization-based is called ESDS-GHS-GLO, and it is presented in [35], but this method also is excluded in the comparisons because presents the same problem that in [34].

The comparison to ROUGE-1 and ROUGE-2 evaluation of our proposed approach with respect to the related methods for DUC01 is presented in Table 4.

In the same way, the comparison to ROUGE-1 and ROUGE-2 evaluation of our proposed approach with respect to the related methods for DUC02 is presented in Table 5.

Table 4
F-measure score ROUGE-1 and ROUGE-2 of the related works with the DUC01 collection

Method	ROUGE-1	ROUGE-2
Proposed	**0.45058 (6)**	**0.19619 (1)**
DE	0.47856 (1)	0.18528 (3)
FEOM	0.47728 (2)	0.18549 (2)
NetSum	0.46427 (3)	0.17697 (4)
CRF	0.45512 (4)	0.17327 (6)
UnifiedRank	0.45377 (5)	0.17649 (5)

Table 5
F-measure score ROUGE-1 and ROUGE-2 of the related works with the DUC02 collection

Method	ROUGE-1	ROUGE-2
Proposed	**0.48423 (2)**	**0.22471 (1)**
UnifiedRank	0.48478 (1)	0.21462 (2)
DE	0.46694 (3)	0.12368 (4)
FEOM	0.46575 (4)	0.12490 (3)
NetSum	0.44963 (5)	0.11167 (5)
CRF	0.44006 (6)	0.10924 (6)

Our proposed genetic algorithm obtains competitive results in comparison to related works. The proposal method obtains the first ranking for both collections with ROUGE-2 evaluation. In the case of ROUGE-1, our algorithm is ranked in sixth and second position for DUC01 and DUC02 collections, respectively.

4.2. Analysis of results

In order to have a final ranking between the results of the related works with our proposal, in Table 5 is calculated the global ranking proposed by [4], which is estimated from the partial ranking obtained from both collections and from both evaluations of all systems. The rank list is calculated according to Equation (12), which is defined as follows

$$rank\,(method) = \sum_{s=1}^{m} \frac{(m - s + 1)\, r_s}{m} \tag{12}$$

Table 6
Global ranking estimated by the partial rakings obtained in each evaluation of ROUGE-1 and ROUGE-2 in the DUC01 and DUC02 collections

Method/	Partial ranking bye valuations						Global
Rankings	1	2	3	4	5	6	ranking
Proposed	2	1	0	0	0	1	**3.0**
DE	1	0	2	1	0	0	2.8
FEOM	0	2	1	1	0	0	2.8
UnifiedRank	1	1	0	0	2	0	2.5
NetSum	0	0	1	1	2	0	1.8
CRF	0	0	0	1	0	3	1.0

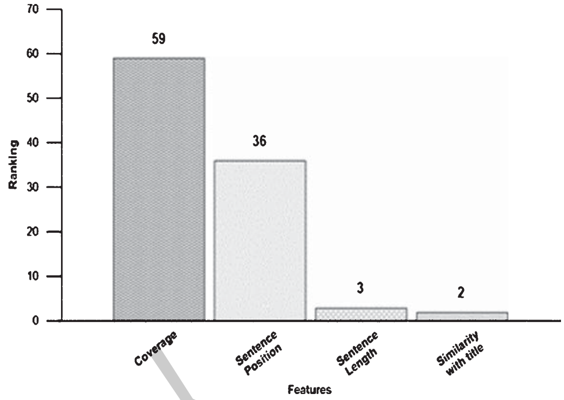

Fig. 5. Ranking of the sentence features calculated in this work.

where r_s denotes the number of times that the method appears in the s rank, and m indicates the number of methods included in the ranking. In Table 6 it is possible to observe that our proposal obtains the best global ranking value from previous related works.

Furthermore, with our work is possible to obtain a ranking of the previously proposed sentence features, where the *coverage* is the strongest feature with 59% of the relevance, the *sentence position* is the second relevance feature with 36% and the *similarity with the title* is the third feature with 2% and, finally, the *sentence length* with 3%. Figure 5 shows a representative graph of the ranking obtained by the sentence features used and reported in this work.

5. Conclusions

This paper proposes an approach based on a genetic algorithm to optimize an ensemble of novel sentence features for extractive text summarization. In the first part of the paper a novel analysis of the sentence features presents the essential steps that follow any extractive text summarization approach is described. With this analysis, it is possible to observe why the coverage, sentence position, similarity with the title and sentence length features were good candidates to compose an ensemble of measures. With the proposed analysis of the sentence features was possible to propose a new sentence length weighting that uses the term length feature. Moreover, the ensemble of the proposed measures into linear combination permits to determine the relevance ranking of the used sentence features.

According to the experimentation, with only four measures it is possible to obtain good results for two standard collections in English, DUC01 and DUC02. Specifically, the coverage and sentence position features obtain a very high relevance in comparison to other features. Coverage feature helps to reduce redundancy information in a summary, that is, reduce duplicated content, whereas sentence position feature selects the most relevant information of a source text. In general, the results obtained generate a global ranking that is better than other state-of-the-art works.

The main contribution indicates that relevant information and redundancy reduction are the most important aspects to create a summary. In this sense, our work is the state-of-the-art framework for using more sentences features.

Future work is expected to involve the application of the proposed method to other collections related to extractive text summarization. Also, it is planned search other sentence features that have more relevance that sentence length and similarity whit title features. Finally, it is planned to make a general genetic algorithm that find the best values for the sentence features used for the proposed method.

References

[1] A. Abuobieda, N. Salim, Y.J. Kumar and A.H. Osman, Opposition Differential Evolution Based Method for Text Summarization, in: *Intelligent Information and Database Systems: 5th Asian Conference, ACIIDS 2013, Proceedings, Part I*, A. Selamat, N.T. Nguyen and H. Haron, eds., Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 487–496.

[2] R.M. Alguliev and R.M. Aliguliyev, Effective summarization method of text documents, in: *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, 2005, pp. 264–271.

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

[4] R.M. Aliguliyev, Performance evaluation of density-based clustering methods, *Inf Sci* **179** (2009), 3583–3602.

[5] R.M. Aliguliyev and N.R. Isazade, Multiple documents summarization based on evolutionary optimization algorithm, *Expert Systems with Applications* **40** (2013), 1675–1689.

[6] A. Bossard, M. Genereux and T. Poibeau, Description of the LIPN systems at TAC 2008: Summarizing information and opinions, 2008.

[7] Z. Cao, W. Li, S. Li and F. Wei, *Improving Multi-Document Summarization via Text Classification*, 2017.

[8] G. Carenini, R.T. Ng and X. Zhou, Summarizing Emails with Conversational Cohesion and Subjectivity, in: *ACL*, 2008, pp. 353–361.

[9] H.P. Edmundson, New methods in automatic extracting, *Journal of the ACM (JACM)* **16** (1969), 264–285.

[10] M.A. Fattah, A hybrid machine learning model for multi-document summarization, *Applied Intelligence* **40** (2014), 592–600.

[11] M.A. Fattah and F. Ren, GA, MR, FFNN, PNN and GMM based models for automatic text summarization, *Computer Speech & Language* **23** (2009), 126–144.

[12] M. Gambhir and V. Gupta, Recent automatic text summarization techniques: A survey, *Artificial Intelligence Review* **47** (2017), 1–66.

[13] R. García-Hernández and Y. Ledeneva, Single Extractive Text Summarization Based on a Genetic Algorithm, in: *Pattern Recognition*, J. Carrasco-Ochoa, J. Martínez-Trinidad, J. Rodríguez and G. di Baja, eds., Springer Berlin Heidelberg, 2013, pp. 374–383.

[14] R. García-Hernández, R. Montiel, Y. Ledeneva, E. Rendón, A. Gelbukh and R. Cruz, Text Summarization by Sentence Extraction Using Unsupervised Learning, in: *MICAI 2008: Advances in Artificial Intelligence*, A. Gelbukh and E. Morales, eds., Springer Berlin Heidelberg, 2008, pp. 133–143.

[15] R.A. García-Hernández, Y. Ledeneva, G.M. Mendoza, Á.H. Dominguez, J. Chavez, A. Gelbukh and J.L.T. Fabela, Comparing Commercial Tools and State-of-the-Art Methods for Generating Text Summaries, in: *Proceedings of the 2009 Eighth Mexican International Conference on Artificial Intelligence*, IEEE Computer Society, 2009, pp. 92–96.

[16] D. Gillick, B. Favre, D. Hakkani-Tür, B. Bohnet, Y. Liu and S. Xie, The ICSI/UTD Summarization System at TAC 2009, 2009.

[17] V. Gupta and G.S. Lehal, A survey of text summarization extractive techniques, *Journal of Emerging Technologies in Web Intelligence* **2** (2010), 258–268.

[18] V.K. Gupta and T.J. Siddiqui, Multi-document summarization using sentence clustering, in: *2012 4th International Conference on Intelligent Human Computer Interaction (IHCI)*, 2012, pp. 1–5.

[19] U. Hahn and I. Mani, The challenges of automatic summarization, *Computer* **33** (2000), 29–36.

[20] T. Hirao, H. Isozaki, E. Maeda and Y. Matsumoto, Extracting important sentences with support vector machines, in: *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, Association for Computational Linguistics, 2002, pp. 1–7.

[21] E. Hovy and C.-Y. Lin, Automated text summarization and the SUMMARIST system, in: *Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998*, Association for Computational Linguistics, 1998, pp. 197–214.

[22] R.K.M. Jhalani and Yogesh, An abstractive approach for text summarization, *International Journal of Advanced Computational Engineering and Networking* **5** (2017), 5–10.

[23] A. Khan, N. Salim and Y. Jaya Kumar, A framework for multi-document abstractive summarization based on semantic role labelling, *Applied Soft Computing* **30** (2015), 737–747.

[24] Y. Ko and J. Seo, An effective sentence-extraction technique using contextual information and statistical approaches for text summarization, *Pattern Recogn Lett* **29** (2008), 1366–1371.

[25] A. Konak, D.W. Coit and A.E. Smith, Multi-objective optimization using genetic algorithms: A tutorial, *Reliability Engineering & System Safety* **91** (2006), 992–1007.

[26] J. Kupiec, J. Pedersen and F. Chen, A trainable document summarizer, in: *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 1995, pp. 68–73.

[27] Y. Ledeneva and R. García-Hernández, *Generación automática de resúmenes Retos, propuestas y experimentos Automatic Generation of Text Sumaries Challenges,*

Proposals and Experiments, Universidad Autónoma del Estado de México (Ed.)/Cigome, 2017.

[28] Y. Ledeneva, A. Gelbukh and R.A. García-Hernández, Terms Derived from Frequent Sequences for Extractive Text Summarization, in: *Computational Linguistics and Intelligent Text Processing: 9th International Conference, CICLing 2008, Haifa, Israel, February 17-23, 2008 Proceedings*, A. Gelbukh, ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 593–604.

[29] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, Barcelona, Spain, 2004.

[30] C.-Y. Lin and E. Hovy, Identifying topics by position, in: *Proceedings of the fifth conference on Applied natural language processing*, Association for Computational Linguistics, Washington, DC, 1997, pp. 283–290.

[31] M. Litvak, M. Last and M. Friedman, A new approach to improving multilingual summarization using a genetic algorithm, in: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 927–936.

[32] H.P. Luhn, The automatic creation of literature abstracts, *IBM Journal of Research and Development* **2** (1958), 159–165.

[33] Y.K. Meena and D. Gopalani, Evolutionary algorithms for extractive automatic text summarization, *Procedia Computer Science* **48** (2015), 244–249.

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

[35] M. Mendoza, C. Cobos and E. León, Extractive Single-Document Summarization Based on Global-Best Harmony Search and a Greedy Local Optimizer, in: *Advances in Artificial Intelligence and Its Applications: 14th Mexican International Conference on Artificial Intelligence, MICAI 2015*, O. Pichardo Lagunas, O. Herrera Alcántara and G. Arroyo Figueroa, eds., Springer International Publishing, Cham, 2015, pp. 52–66.

[36] R. Mihalcea and P. Tarau, TextRank: Bringing Order into Text, in: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, 2004, pp. 404–411.

[37] J.L. Neto, A.A. Freitas and C.A.A. Kaestner, Automatic Text Summarization Using a Machine Learning Approach, in: *Advances in Artificial Intelligence: 16th Brazilian Symposium on Artificial Intelligence, SBIA 2002, Proceedings*, G. Bittencourt and G.L. Ramalho, eds., Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 205–215.

[38] C. Nobata, S. Sekine, M. Murata, K. Uchimoto, M. Utiyama and H. Isahara, Sentence Extraction System Assembling Multiple Evidence, in: *NTCIR*, 2001.

[39] Y. Ouyang, W. Li, S. Li and Q. Lu, Applying regression models to query-focused multi-document summarization, *Information Processing & Management* **47** (2011), 227–237.

[40] Y. Ouyang, W. Li, Q. Lu and R. Zhang, A study on position information in document summarization, in: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, Association for Computational Linguistics, 2010, pp. 919–927.

[41] M.F. Porter, An algorithm for suffix stripping, in: *Readings in information retrieval*, J. Karen Sparck and W. Peter, eds., Morgan Kaufmann Publishers Inc., 1997, pp. 313–316.

[42] P.S. Premjith, A. John and M. Wilscy, Metaheuristic Optimization Using Sentence Level Semantics for Extractive Document Summarization, in: *Mining Intelligence and Knowledge Exploration: Third International Conference, MIKE 2015, Proceedings*, R. Prasath, A.K. Vuppala and T. Kathirvalavakumar, eds., Springer International Publishing, Cham, 2015, pp. 347–358.

[43] P. Reddy and R.C. Balabantaray, Improvisation of the document summarization by combining the IR techniques with "code-quantity memory and attention" linguistic principles, *Procedia Technology* **6** (2012), 118–125.

[44] K. Riedhammer, B. Favre and D. Hakkani-Tür, Long story short – Global unsupervised models for keyphrase based meeting summarization, *Speech Communication* **52** (2010), 801–815.

[45] S. Saziyabegum and P.S. Sajja, Literature review on extractive text summarization approaches, *International Journal of Computer Applications* **156** (2016).

[46] L. Shao, H. Zhang, M. Jia and J. Wang, Efficient and Effective Single-Document Summarizations and A Word-Embedding Measurement of Quality, *arXiv preprint arXiv:1710.00284* (2017).

[47] D. Shen, J.-T. Sun, H. Li, Q. Yang and Z. Chen, Document Summarization Using Conditional Random Fields, in: *IJCAI*, 2007, pp. 2862–2867.

[48] G. Sidorov, A. Gelbukh, H. Gómez-Adorno and D. Pinto, Soft similarity and soft cosine measure: Similarity of features in vector space model, *Computación y Sistemas* **18** (2014), 491–504.

[49] C. Silla, G. Pappa, A. Freitas and C. Kaestner, Automatic text summarization with genetic algorithm-based attribute selection, *Advances in Artificial Intelligence–IBERAMIA 2004*, 2004, pp. 305–314.

[50] 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 Systems with Applications* **38** (2011), 9112–9121.

[51] K. Sparck, Automatic Summarising: Factors and Directions, in: *Advances in Automatic Text Summarization*, MIT Press, 1998, pp. 1–12.

[52] N. Srinivas and K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, *Evol Comput* **2** (1994), 221–248.

[53] L. Suanmali, N. Salim and M.S. Binwahlan, Genetic algorithm based sentence extraction for text summarization, *International Journal of Innovative Computing* **1** (2011).

[54] K.M. Svore, L. Vanderwende and C.J. Burges, Enhancing Single-Document Summarization by Combining RankNet and Third-Party Sources, in: *Emnlp-conll*, 2007, pp. 448–457.

[55] V. Thada and D.V. Jaglan, Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm, *International Journal of Innovations in Engineering and Technology (IJIET)* **2** (2013), 202.

[56] T. Vodolazova, E. Lloret, R. Muñoz and M. Palomar, Extractive Text Summarization: Can We Use the Same Techniques for Any Text? in: *Natural Language Processing and Information Systems: 18th International Conference on Applications of Natural Language to Information Systems, NLDB 2013*, E. Métais, F. Meziane, M. Saraee, V. Sugumaran and S. Vadera, eds., Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 164–175.

[57] X. Wan, 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, 2010, pp. 1137–1145.
[58] F. Wei, W. Li and S. Liu, iRANK: A rank-learn-combine framework for unsupervised ensemble ranking, *Journal of the American Society for Information Science and Technology* **61** (2010), 1232–1243.
[59] K.-F. Wong, M. Wu and W. Li, Extractive summarization using supervised and semi-supervised learning, in: *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, Association for Computational Linguistics, 2008, pp. 985–992.
[60] L. Yang, X. Cai, Y. Zhang and P. Shi, Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization, *Information Sciences* **260** (2014), 37–50.
[61] J.K. Yogan, O.S. Goh, B. Halizah, H.C. Ngo and C. Puspalata, A review on automatic text summarization approaches, *Journal Of Computer Science* **12** (2016), 178–190.