Expert Systems with Applications 41 (2014) 4158–4169

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Extractive single-document summarization based on genetic operators and guided local search

Martha Mendoza [a,b,*], Susana Bonilla [a], Clara Noguera [a], Carlos Cobos [a,b], Elizabeth León [c]

[a] Information Technology Research Group (GTI), Universidad del Cauca, Sector Tulcán Office 450, Popayán, Colombia
[b] Computer Science Department, Electronic and Telecommunications Engineering Faculty, Universidad del Cauca, Colombia
[c] Data Mining Research Group (MIDAS), Engineering Faculty, Universidad Nacional de Colombia, Bogotá, Colombia

ARTICLE INFO

Keywords:
Extractive summarization
Single document
Memetic algorithm
Guided local search

ABSTRACT

Due to the exponential growth of textual information available on the Web, end users need to be able to access information in summary form – and without losing the most important information in the document when generating the summaries. Automatic generation of extractive summaries from a single document has traditionally been given the task of extracting the most relevant sentences from the original document. The methods employed generally allocate a score to each sentence in the document, taking into account certain features. The most relevant sentences are then selected, according to the score obtained for each sentence. These features include the position of the sentence in the document, its similarity to the title, the sentence length, and the frequency of the terms in the sentence. However, it has still not been possible to achieve a quality of summary that matches that performed by humans and therefore methods continue to be brought forward that aim to improve on the results. This paper addresses the generation of extractive summaries from a single document as a binary optimization problem where the quality (fitness) of the solutions is based on the weighting of individual statistical features of each sentence – such as position, sentence length and the relationship of the summary to the title, combined with group features of similarity between candidate sentences in the summary and the original document, and among the candidate sentences of the summary. This paper proposes a method of extractive single-document summarization based on genetic operators and guided local search, called MA-SingleDocSum. A memetic algorithm is used to integrate the own-population-based search of evolutionary algorithms with a guided local search strategy. The proposed method was compared with the state of the art methods UnifiedRank, DE, FEOM, NetSum, CRF, QCS, SVM, and Manifold Ranking, using ROUGE measures on the datasets DUC2001 and DUC2002. The results showed that MA-SingleDocSum outperforms the state of the art methods.

1. Introduction

Due to the exponential growth of textual information available on the Web and the access to information by the users through new portable devices, it is necessary that the end user can access the information in summary form and without losing the most important aspects presented therein. Some of the application areas of the generation of extractive summaries from a single document are the summaries of web pages presented on the search engines (Porselvi & Gunasundari, 2013); the assignation of the labels to groups generated in the web document clustering (Carpineto, Osinski, Romano, & Weiss, 2009); and in the E-learning context is used to select the most important information from a text (Kumaresh & Ramakrishnan, 2012). The automatic generation of text summaries has been tasked with addressing this problem for many years, seeking to obtain short texts that present the most relevant ideas in a document (Lloret & Palomar, 2012; Nenkova & McKeown, 2012; Spärck Jones, 2007). To achieve this, several methods have been developed that summarize one or multiple documents, with the aim that the user select and review in the shortest time those documents that really meet their information needs.

Different taxonomies for the summaries exist (Ježek & Steinberger, 2008; Lloret & Palomar, 2012; Nenkova & McKeown, 2012), based on the way the summary is generated, the target audience of the summary, the number of documents to be summarized, and so on.

According to the way in which it is generated, the summary may represent either an extraction or an abstraction

* Corresponding author at: Information Technology Research Group (GTI), Universidad del Cauca, Sector Tulcán Office 450, Popayán, Colombia. Tel.: +57 28366524; fax: +57 28209810.
E-mail addresses: mmendoza@unicauca.edu.co, mendoza.martha.eliana@gmail.com (M. Mendoza).

0957-4174/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.12.042

(Ježek & Steinberger, 2008; Lloret & Palomar, 2012; Nenkova & McKeown, 2012). Extraction summaries are formed from the reuse of portions of the original text. Abstraction based summaries, on the other hand, are rather more complex, requiring linguistic analysis tools to construct new sentences from those previously extracted.

Depending on the target audience, summaries may be (Ježek & Steinberger, 2008; Lloret & Palomar, 2012; Nenkova & McKeown, 2012) generic, query-based, user-focused or topic-focused. Generic summaries do not depend on the audience for whom the summary is intended. Query-based summaries respond to a query made by the user. User-focused ones generate summaries to tailor the interests of a particular user, while topic-focused summaries emphasize those summaries on specific topics of documents.

With regard to the number of documents that are processed, summaries (Ježek & Steinberger, 2008; Lloret & Palomar, 2012; Nenkova & McKeown, 2012) can be either single document or multiple document. In addition, as regards the language of the document, they may be monolingual or multilingual, and regarding document genre may be scientific article, news, blogs, and so on.

A huge diversity is to be found among the methods of automatic generation of extractive summaries from a single document. These are mainly based on the handling of basic statistical features such as sentence position and terms frequency (Edmundson, 1969), based on machine learning techniques (Aone, Okurowski, Gorlinsky, & Larsen, 1999; Conroy & O'leary, 2001; Dunlavy, O'Leary, Conroy, & Schlesinger, 2007; Kupiec, Pedersen, & Chen, 1995; Shen, Sun, Li, Yang, & Chen, 2007; Svore, Vanderwende, & Burges, 2007; Wong, Wu, & Li, 2008), connectivity of texts (Barzilay & Elhadad, 1997; Louis, Joshi, & Nenkova, 2010; Marcu, 1998; Ono, Sumita, & Miike, 1994), graphs (Mihalcea & Tarau, 2004) (Wan, 2010), algebraic reduction (Gong, 2001; Lee, Park, Ahn, & Kim, 2009; Steinberger & Jezek, 2004; Steinberger & Ježek, 2006; Yeh, Ke, Yang, & Meng, 2005) and evolutionary models (Abuobieda, Salim, Kumar, & Osman, 2013; Aliguliyev, 2009a; Binwahlan, Salim, & Suanmali, 2009, 2010; Dehkordi, Kumarci, & Khosravi, 2009; Fattah & Ren, 2009; García-Hernández & Ledeneva, 2013; Litvak, Last, & Friedman, 2010; Qazvinian, Sharif, & Halavati, 2008; Shareghi & Hassanabadi, 2008; Steinberger & Ježek, 2006).

Evolutionary algorithms have traditionally shown good results in solving the problem of extractive summarization (Aliguliyev, 2009a; Binwahlan et al., 2009, 2010; Fattah & Ren, 2009; Litvak et al., 2010; Qazvinian et al., 2008; Shareghi & Hassanabadi, 2008; Steinberger & Ježek, 2006), while memetic algorithms (evolutionary algorithms with local search heuristics) have contributed to the successful resolution of different combinatory optimization problems (Cobos, Montealegre, Mejía, Mendoza, & León, 2010; Neri & Cotta, 2012). Nevertheless, memetic algorithms have not until now been used for solving the specific problem of extractive single-document summarization. In this paper, therefore, we propose a method of generic-extractive summarization for a monolingual document of any genre, based on memetic algorithms. In this case, the evaluation was done using news items.

This algorithm, called MA-SingleDocSum, defines the quality of a solution based on the weighting of individual statistical features of each sentence, such as position, sentence length and the relation of the summary to the title, combined with group features based on the similarity between candidate sentences in the summary and the original document, and the similarity among the sentences in the summary in order to obtain coverage of the summary and cohesion of summary sentences. The algorithm consists of rank-based and roulette wheel parent selection, one-point crossover, multi-bit mutation, guided search-based local optimization, and restricted competition replacement.

The rest of the paper is organized as follows: Section 2 introduces work related to automatic generation of the extractive summaries from a single document; document representation, similarity measures, and features of the objective function proposed are presented in Section 3; the strategies for selection, crossover, mutation, local search and replacement that make up the proposed memetic algorithm are described in Section 4; while the results of evaluation using data sets, along with a comparison and analysis with other state of the art methods, are presented in Section 5; and finally, Section 6 presents the conclusions and future work.

2. Related work

Early research suggests as relevant factors for the score of a sentence and its inclusion in the summary the use of the frequency of occurrence of a term in a text, the position of the sentences in the document, and the presence of keywords or words from the document title in the sentences (Edmundson, 1969).

Using the machine learning approach, Bayes' Theorem has been applied to develop a function that estimates the probability that a sentence be included in a summary (Aone et al., 1999; Kupiec et al., 1995). As such, an approach is proposed based on the Hidden Markov Model (HMM), whose main feature is the recognition of local dependencies between sentences through a sequential model (Conroy & O'leary, 2001; Dunlavy, O'Leary, Conroy, & Schlesinger, 2007). Neural networks (Svore et al., 2007) and Conditional Random Fields (Shen et al., 2007) are also used. More recently, the Probabilistic Support Vector Machine (PSVM) and Naïve Bayesian Classifier were used in an semi-supervised learning approach (Wong et al., 2008)

Other works have applied approaches based on text connectivity, in order to establish the connections that may exist between different parts of a text to try to achieve more coherent and more understandable summaries (Marcu, 1998; Ono et al., 1994). Highlighted among these is the use of lexical chains. This approach starts with the segmentation of the original text and continues with the construction of lexical chains, the identifying the strongest chains and extracting the most significant sentences, completing the process of the production of the summary (Barzilay & Elhadad, 1997). More recently, the rhetorical structure theory approach has also been employed (Louis et al., 2010).

In addition, the graphs have been adapted for the automatic generation of extractive summaries (Mihalcea & Tarau, 2004), where the sequence of one or more lexical units extracted from a text and the relationships between them are the vertices and edges of the graph, respectively. A particular focus based on graphs is that proposed by Wan (2010), in which the automatic summarization of one and of multiple documents is carried out at the same time, making use of a local importance that indicates the relevance of a sentence within a document to generate the summary of a single document; and of a global importance, that indicates the relevance of the same sentence but at the level of the entire set of documents to generate the summary of multiple documents.

In the case of algebraic reduction, the most widely used method for extractive summarization is that based on Latent Semantic Analysis (LSA), which allows the extracting, representing and comparing the meaning of words using the algebraic-statistical analysis of a text, the basic assumption for which is that the meaning of a word is determined by its frequent occurrence next to other words. Gong (2001) proposed using LSA for automatic generation of generic summaries, applying Singular Value Decomposition (SVD). The semantic analysis process consists of two steps. The first is the creation of a terms by sentence matrix $A = [A_1, A_2, \ldots, A_n]$, where each column A_i represents the weight vector, based on the frequency of terms from the sentence i in the document. The next step consists of applying SVD to matrix A. To generate a summary,

the most important sentence is selected using topics identified from the decomposition of A. A similar approach is presented in (Steinberger & Jezek, 2004), but changing the selection criteria to include in the summary sentences whose vector representation in the matrix have bigger "length", rather than the sentences containing the highest index value for each "Topic". Yeh et al. (2005) propose another method that uses LSA and a text relationship map (TRM) to derive semantically salient structures from a document in which, after performing SVD on the terms by sentence matrix and reducing the dimensionality of the latent space, they reconstruct an additional matrix wherein each column denotes the semantic representation of the sentence. On the other hand, in Steinberger and Ježek (2006) the system proposed in Steinberger and Jezek (2004) is combined with a sentence compression algorithm that eliminates the unimportant parts of a sentence. Lately, Lee et al. (2009) propose an unsupervised method using Non-negative Matrix Factorization (NMF).

More recently, several approaches based on evolutionary models for extractive summarization have been explored. In Dehkordi et al. (2009), an evolutionary algorithm based on genetic programming is presented, which defines a sentence ranking function. Genetic algorithms have also been used as the means of extracting sentences that will make up a summary (García-Hernández & Ledeneva, 2013; Qazvinian et al., 2008) and, further, for optimizing the weights of factors that give the score for each sentence of a document (Fattah & Ren, 2009; Litvak et al., 2010). In Binwahlan et al. (2009) a model based on particle swarm optimization (PSO) is proposed, to obtain the weights of the sentence features, and thereby qualify the sentences and select that with the highest score for inclusion in the summary. Harmony Search too has been used to extract sentences that will include the final summary, using an objective function composed of such factors as cohesion, readability and relationship with the title (Shareghi & Hassanabadi, 2008). Further, in Aliguliyev (2009a) a differential evolution algorithm is used for clustering sentences – an individual is represented by permutations indicating the group wherein each sentence corresponding to a gene will be located. The centrality of each sentence in the cluster is measured and the most important are extracted to be included in the summary. Also, in Abuobieda et al. (2013) a differential evolution algorithm is used for clustering sentences and automatically generating a summary.

Binwahlan et al. (2010) proposed a fuzzy-swarm hybrid diversity model that combines three methods based on diversity, swarm and fuzzy-swarm. The diversity-based method forms sentence groups arranged in a binary tree according to their scores. It then applies Maximal Marginal Importance (MMI) to select the sentences for including in the summary. The method based on PSO binary is used to optimize the weight corresponding to each characteristic of the objective function. The position of the particle is a string of bits, where one means that the corresponding characteristic is selected, otherwise it has a zero. On obtaining the weights, the score is calculated for each sentence and the sentences with the highest score are chosen to be included in the summary. In the method based on swarms and fuzzy logic, the fuzzy algorithm calculates the sentence score by a system of inference, beginning with the weights found with PSO. It then converts the result of the inference process (final scores of the sentences), and the sentences are then sorted according to the resulting score and the summary is obtained. To finish, another procedure is employed in order to select the sentences from the summaries produced by each of the three methods above.

Song, Cheon Choi, Cheol Park, and Feng Ding (2011) proposed a fuzzy evolutionary optimization model (FEOM) to simultaneously carry out document clustering and generate summaries. The method for automatic summarization is based on the concept of clustering of document sentences. The most important sentences are then selected from each group to obtain the summary. FEOM uses genetic algorithms, generating a random population as the initial set of clustering solutions. Each individual in the population is a string of real numbers. The three evolutionary operators (selection, crossover and mutation) are used to produce new offspring until the termination criterion is met. Three control parameters (distribution coefficient, relative distance, effect of evolution) are applied to regulate the probability of crossover and mutation of each solution.

This paper similar to the others works address the generation of extractive summaries from a single document as a binary optimization problem. But unlike from these methods, in this study is realized the combination of population-based global search with a local search heuristic (memetic approach). This heuristic exploits the problem knowledge for redirect the search toward a local best solution. In this case, the guided local search was used to achieve this objective. In addition, the method proposed uses a fitness function that is the result of the weighting of statistical features of each sentence (position, sentence length, and the relationship of the sentence to the title) combined with group features (similarity between candidate sentences in the summary and the original document, and among the candidate sentences of the summary).

3. Problem statement and its mathematical formulation

3.1. Document representation and similarity measures

The representation of a document is performed based on the vector space model (VSM) proposed by Salton (Manning, Raghavan, & Schtze, 2008). Thus, a document is represented by the set $D = \{S_1, S_2, \ldots, S_n\}$ where S_i corresponds to the i-th sentence of the document and n is the number of sentences that comprise it. Likewise, a sentence in the document is represented by the set $S_i = \{t_{i1}, t_{i2}, \ldots, t_{ik}, \ldots, t_{im}\}$, where t_{ik} is the k-th term of the sentence S_i and m is the total number of terms of the whole document. Therefore, the vector representation of a sentence in the document is $S_i = \{w_{i1}, w_{i2}, \ldots, w_{ik}, \ldots, w_{im}\}$, where w_{ik} is the weight or weighting of the term t_k in the sentence S_i. This weight is calculated as the relative frequency of the term in the document (Manning & Raghavan, 2008) and is calculated according to Eq. (1)

$$w_{ik} = (f_{ik}/MaxFreq_i) * \log(n/(1 + n_k)) \quad (1)$$

where f_{ik} is the frequency of the term k in sentence S_i, $MaxFreq_i$ is an adjustment factor that indicates the number of occurrences of the most frequent term in the sentence S_i and n_k is the number of sentences where the term t_k appears.

In this context, the objective in generating a summary of a document is to find a subset of sentences $S \subseteq D$ that contain the main information of the document. For this, features are used whose purpose is to evaluate the subset of sentences to determine the extent to which they cover the most relevant information in the document. Some of these features are based on similarity measures between sentences. The similarity between two sentences S_i and S_j, according to the vector representation described is calculated as the *cosine similarity* (Manning & Raghavan, 2008), which is related to the angle of the vectors S_i and S_j, and is calculated according to Eq. (2)

$$sim_{\cos}(S_i, S_j) = \frac{\sum_{k=1}^{m} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{m} w_{ik}^2 \bullet \sum_{k=1}^{m} w_{jk}^2}} \quad (2)$$

where m is the total number of terms in the document, w_{ik} refers to the weight of the term k in the sentence S_i and w_{jk} is the weight of the term k in the sentence S_j.

3.2. Features of objective function

The automatic text summarization seeks to select the most relevant sentences in a document, so it is important to establish the features that help identify these sentences and thus improve the quality of the summaries generated. In this study a set of features was used, independent of the domain and language, to determine the quality of a summary based on the sentences of which it is comprised. These are: the position of the sentences in the document, the relationship of the sentences to the title, the length of sentences, the cohesion between sentences of the summary, and the coverage of the sentences of the summary. These features together form the objective function to be optimized by the memetic algorithm proposed. Each of the features above is described in the following.

3.2.1. Sentence position

If all the sentences in a document have the same importance, by reducing the size of the document to generate a summary, significant information would be lost. However, according to studies performed, the relevant information in a document, regardless of its domain (Lin & Hovy, 1997), tends to be found in certain sections such as titles, headings, the leading sentences of paragraphs, the opening paragraphs, etc. To evaluate a sentence based on its position, a selection criterion is defined that uses the distance that exists between the sentence and the start of the document, assigning greater value to the initial sentences. In information retrieval several techniques have been applied based on the position of the sentences and combined with other selection criteria. These have proven effective in determining the relevance of a sentence (Bossard, Genereux, & Poibeau, 2008; Fattah & Ren, 2009; Radev, Jing, StyÅ, & Tam, 2004). One such scheme is that used by Bossard et al. (2008), where a standard calculation is applied from the position based on Eq. (3)

$$P = \sum_{\forall S_i \in Summary} \sqrt{\frac{1}{q_i}} \tag{3}$$

where q_i indicates the position of the sentence S_i in the document, and P is the result of the calculation for all sentences of the summary. In this equation, P has high values when sentences in summary belong to the first sentences in the document, and P has low values when sentences in summary belong to the last sentences in the document.

3.2.2. Relation of sentences with title

This characteristic is based on the assumption that a good summary contains sentences similar to the title of the document (Silla, Nascimento, Pappa, Freitas, & Kaestner, 2004). Calculation of this similarity begins with their representation through the vector space model and the cosine similarity measure (Qazvinian et al., 2008; Shareghi & Hassanabadi, 2008) is used, as shown in Eq. (4)

$$RT_s = \sum_{\forall S_i \in Summary} \frac{sim_{\cos}(S_i, t)}{O}$$
$$RTF_s = \frac{RT_s}{\max\limits_{\forall Summary} RT} \tag{4}$$

where $sim_{cos}(S_i, t)$ is the cosine similarity of sentence S_i with title t, O is the number of sentences in the summary, RTs is the average of the similarity of the sentences in the summary S with the title, $\max_{\forall summary} RT$ is the average of the maximum values obtained from the similarities of all sentences in the document with the title (i.e. the average top greater O similarities of all sentences with the title), and $RTFs$ is the similarity factor of the sentences of the summary S with the title. RTF is close to one (1) when sentences in summary are closely related to the document title and RTF is close to zero (0) when sentences in summary are very different to the document title.

3.2.3. Sentence length

Some studies have concluded that the shortest sentences of a document ought to be less likely to appear in the document summary (Kupiec et al., 1995). One recent study carried out a normalization based on the sigmoid function for calculating this feature (Gupta, Chauhan, & Garg, 2012). This estimate takes into account the standard distribution of the data in order to reach a more balanced evaluation, which still favors the longest sentences, but does not completely rule out those of medium length, on the presumption that they could also have relevant information for the summary. Therefore, taking into account that the standard distribution represents the tendency of the data to vary either above or below the mean value, it is expected that a sentence with not too short a length will obtain a good grade in this characteristic. Based on these premises, Eq. (5) shows the calculation of length for the sentences of a summary (L),

$$L = \sum_{\forall S_i \in Summary} \frac{1 - e^{\frac{-l(S_i)-\mu(l)}{std(l)}}}{1 + e^{\frac{-l(S_i)-\mu(l)}{std(l)}}} \tag{5}$$

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

3.2.4. Cohesion

Cohesion is a characteristic that determines the degree of relatedness of the sentences that make up a summary (Qazvinian et al., 2008; Shareghi & Hassanabadi, 2008). Ideally, the connection between the ideas expressed in the sentences of the summary should be tightly coupled. For its calculation, the cosine similarity measure of one sentence to another is used, see Eq. (6)

$$CoH = \frac{\log(C_s * 9 + 1)}{\log(M * 9 + 1)}$$
$$C_s = \frac{\sum_{\forall S_i, S_j \in Summary} sim_{\cos}(S_i, S_j)}{N_s}, \quad N_s = \frac{(O) * (O-1)}{2} \tag{6}$$
$$M = \max Sim_{\cos}(i, j), \quad i, j \leqslant N$$

where CoH corresponds to the cohesion of a summary, C_s is the average similarity of all sentences in the summary S, $sim_{cos}(S_i, S_j)$ is the cosine similarity between sentences S_i and S_j, N_s is the number of nonzero similarity relationships in the summary, O is the number of sentences in the summary, M corresponds to the maximum similarity of the sentences in the document and N is the number of sentences in the document. In this way, CoH tends to zero when the summary sentences are too different among them, while that CoH tends to one when these sentences are too similar among them. Thus, this feature tends to favor the summaries that contain sentences about the same topic.

3.2.5. Coverage

Coverage attempts to measure the extent to which the sentences of a summary provide the reader with the most important information from the original document (Wei, Li, & Liu, 2010). Thus, this characteristic is defined as the similarity between the sentences that make up a summary and the full document. Each of the sentences the document is therefore represented through the vector space model and is weighted by calculating its relative frequency according to Eq. (7)

$$Cov = \sum_{\forall S_i \in Summary} \sum_{\forall S_j \in Summary, j>i} [sim_{\cos}(D, S_i) + sim_{\cos}(D, S_j)] \quad (7)$$

where D is the vector of weights of the terms in the document, and S_i and S_j are the vectors of weights of the terms in the sentences i and j, respectively, belonging to the summary.

4. Proposed memetic algorithm: MA-SingleDocSum

The memetic algorithm (MA) proposed in this research seeks to optimize the linear combination of the features of the Eqs. (3)–(7). This type of algorithm combines a population-based global search with a local search heuristic made by each agent, i.e. it couples genetic evolution with the learning that individuals achieve during their period of existence (Hao, 2012). The main objective of memetic algorithms, by incorporating individual optimizations, processes of cooperation and population competition, is to direct the exploration towards the most promising regions of the search space. A process of competition involves techniques for the selection of individuals, while a process of cooperation refers to the summarization of new individuals through the exchange of information.

4.1. Memetic algorithm basic

A basic memetic algorithm is executed throughout populations of individuals, which in this context are known as agents (Hao, 2012). An agent is a representation of a solution, or in some cases of several, and is characterized by its active behavior in the resolution of the problem that it addresses. The agents of a population compete and cooperate with each other during evolution, this being a prominent characteristic within the MA. The structure through which the genotypic information of an agent is represented is the chromosome structure. The MA starts with a population of ps n-dimensional agents, the i-th agent of the population in a time or generation g has n components (memes) as seen in Eq. (8).

$$X_i(g) = [x_{i,1}(g), x_{i,2}(g), \ldots, x_{i,n}(g)], \quad i = 1, 2, \ldots, ps \quad (8)$$

The generational step of a population in a time g to another in $g+1$ is accomplished through the processes of selection, reproduction and replacement of agents. Before the reproductive stage, two agents, $X_p(g)$ and $X_m(g)$ are selected, based on the fitness values obtained from the objective function. These will act as parents of a new agent in the new generation. In the reproductive stage, through a crossover operator, information is exchanged between $X_p(g)$ and $X_m(g)$ to give rise to a new agent $Y_i(g)$. Within the reproduction stage, the inclusion of information foreign to the agent generated is also accomplished through a mutation operator, which takes $Y_i(g)$, to partially modify and generate an agent $Z_i(g)$. The mutation is executed based on a mutation probability Mr, as shown in Eq. (9)

$$z_i(g) = \begin{Bmatrix} Mutate(Y_i(g)) & \text{if} & rand < Mr \\ Y_i(g) & & \text{otherwise} \end{Bmatrix} \quad (9)$$

where the method $Mutate(\cdot)$ modifies one or more memes of an agent.

Likewise, the agent generated is also optimized by a local search operator, based on a probability of optimization Opr, according to Eq. (10)

$$A_i(g) = \begin{Bmatrix} BL(Z_i(g)) & \text{if} & rand < Opr \\ Z_i(g) & & \text{otherwise} \end{Bmatrix} \quad (10)$$

where $BL(\cdot)$ method is the operator of local search that improves an agent.

The population is updated by replacing an agent $X_r(g)$ according to a specific replacement technique for the new offspring according to its fitness value, as shown in Eq. (11)

$$X_i(g+1) = \begin{Bmatrix} A_i(g) & \text{if} & F(A_i(g)) > F(X_r(g)) \\ X_r(g) & & \text{otherwise} \end{Bmatrix} \quad (11)$$

where $F(\cdot)$ is the objective function to be maximized.

The selection, reproduction and update are run until the population size ps is completed. The generational process of competition and cooperation described is repeated until a stopping criterion is satisfied.

4.2. Representation of solution

In the memetic algorithm proposed, the coding of a solution or agent is performed using a binary vector. Thus, if a document is composed of n sentences $\{S_1, S_2, \ldots, S_n\}$ the candidate agent is composed of n memes, each representing a sentence in the document, taking the value of one if the sentence belongs to the summary represented by the agent, or zero otherwise. For example, if there is a document with $n = 10$, i.e. having ten sentences, the solution vector [0, 1, 1, 0, 1, 0, 0, 1, 0, 0] indicates that the summary represented by this agent is composed of the second, third, fifth and eighth sentence of the original document.

In that sense, the c-th agent of the present population (g generation) is represented as shown in Eq. (12)

$$X_c(g) = [x_{c,1}(g), x_{c,2}(g), \ldots, x_{c,s}(g), \ldots, x_{c,n}(g)] \quad (12)$$

where $x_{c,s}(g) \in \{0, 1\}$ is a binary integer, n is the number of sentences in the document, $c = 1, 2, \ldots, ps$, and ps is the population size.

4.3. Fitness function

The definition of the objective function is one of the most important steps in the design of memetic algorithms, as it helps to guide the exploration and exploitation mechanism. The objective function is responsible for evaluating and assigning a fitness value to the agents of the population, based on their ability to solve the problem addressed. To assess the quality of a summary represented by an agent X_k, an objective function is required, which will be maximized according to Eq. (13), whose components correspond to the mathematical formulas of Eqs. (3)–(7). These equations are the features it is desired to maximize for each agent. The coefficients of the objective function must satisfy the constraint of Eq. (14)

$$Max(f(X_k)) = \alpha P(X_k) + \beta RT(X_k) + \gamma L(X_k) + \delta CoH(X_k) + \rho Cob(X_k) \quad (13)$$

subject to

$$\alpha + \beta + \gamma + \delta + \rho = 1 \quad (14)$$

where α, β, γ, δ, ρ are coefficients which give a weighting to each objective function feature.

4.4. Population initialization

The most common strategy for initializing the population (time $g = 0$) is to generate each agent randomly. So that all sentences in the document have the same probability of being part of the agent, we define a random number between one and n (number of sentences in the document). A value of one is given to the gene (sentence) that corresponds to the random value defined, thereby indicating that this sentence becomes part of the summary in the current agent. Thus, the c-th agent of the initial population is created as shown in Eq. (15)

$$X_c(0) = [x_{c,1}(0), x_{c,2}(0), \ldots, x_{c,n}(0)],\ x_{c,s}(0) = a_s \quad (15)$$

where a_s is a binary integer $\{0,1\}$, $c = 1,2,\ldots,ps$, $s = 1,2,\ldots,n$, and n is the number of sentences in the document.

When a value $x_{c,s}(0)$ takes the value of one, the summary length constraint represented by the agent is verified based on Eq. (16)

$$\sum_{s_i \in Summary} l_i \leqslant S \quad (16)$$

where, l_i is the length of the sentence S_i (measured in words) and S is the maximum number of words allowed in the generated summary.

4.5. Selection

The generational step begins with this process, selecting a certain number of agents from the current population (time g), using an elitist strategy so that they pass unchanged to the next generation (time $g+1$).

Thus, if $Pob(g) = \{X_1(g), X_2(g), \ldots, X_{ps}(g)\}$ is the current population in descending order according to the fitness values of its members, the group of agents chosen to pass to the next generation corresponds to $E(g+1) = \{X_1(g), X_2(g), \ldots, X_e(g)\}$ where $E(g+1) \subseteq Pob(g)$, $e < ps$ and e is an predefined parameter that specifies the number of agents selected by elitism.

The rest of the population of the next generation is created as described below, selecting the parents of the new offspring. In this context, the father is selected by the *Rank selection* strategy (Sivanandam & Deepa, 2008), while the mother is chosen by *Roulette wheel* selection (Sivanandam & Deepa, 2008).

To select the father, $X_p(g)$, the agents from the current population are ordered in descending order by their fitness values and the range of each agent is calculated, so that for the i-th agent, the range is as seen in Eq. (17)

$$r(X_i(g)) = s - \frac{2(s-1)(j-1)}{(ps-1)} \quad (17)$$

where ps is the population size, j is the position of the agent in the ordered population, s is the selective pressure that may be determined as the relationship between the fittest individual and the medium individual.

Based on the values of range, a probability is defined, which for the i-th agent is calculated as shown in Eq. (18).

$$prb(X_i(g)) = \frac{r(X_i(g))}{ps} \quad (18)$$

Then a random value a in the range $[0,1]$ is generated. The first agent whose probability $prb(\cdot)$ exceeds the value of a, is selected as a father.

To select the mother $X_m(g)$, the cumulative probability of the current population is calculated as shown in Eq. (19)

$$Pacu = a * \sum_{i=1}^{ps} F(X_i(g)) \quad (19)$$

where $F(X_i(g))$ is the fitness value of the i-th agent in the current population and a is a random value in the range $[0,1]$.

The fitness values of the agents of population are then summed sequentially, such that the sum of the i-th agent corresponds to Eq. (20). The first agent for which $Sum_{acu}(\cdot)$ exceeds the probability value P_{acu}, is selected as a mother.

$$Sum_{acu}(X_i(g)) = \sum_{j=1}^{i} F(X_j(g)) \quad (20)$$

4.6. Crossover

To generate an offspring the one-point crossover strategy (Sivanandam & Deepa, 2008) is used. Thus, the selected parents $X_p(g)$ and $X_m(g)$ exchange part of their chains after a randomly selected point in order to generate the agent $Y_i(g)$, such that its s-th meme $Y_{i,s}(g)$ is calculated as in Eq. (21)

$$Y_{i,s}(g) = \begin{cases} x_{p,s}(g), & \text{if } s \leqslant ptC \\ x_{m,s}(g) & \text{otherwise} \end{cases} \quad (21)$$

where $x_{p,s}(g)$ is the s-th meme of the father $X_p(g)$, $x_{m,s}(g)$ is the s-th meme of the mother $X_m(g)$ and ptC is an integer that represents the randomly selected cutting point between $[1,n]$, where n is the size of the agent. To generate a second offspring, this same process is followed, exchanging the role of the parents. For each offspring, the summary length constraint represented by the agent is checked based on Eq. (16). If this restriction is not met, one of the sentences is randomly removed and the process repeated until the restriction is met.

4.7. Mutation

An agent $Y_i(g)$ is mutated according to Eq. (9) presented above. The mutation technique applied corresponds to a multi-bit strategy, in which it is decided whether or not a meme of agent should be mutated based on a second probability of mutation Mr_2, according to Eq. (22). Before mutating (placing the gene in one), the summary length constraint represented by the agent is checked based on Eq. (16). If the restriction is not met, the meme is not mutated.

$$Z_{i,s}(g) = \begin{cases} 1 & \text{if } a < Mr_2 \wedge y_{i,s}(g) = 0 \\ y_{i,s}(g) & \text{otherwise} \end{cases} \quad (22)$$

where a is a random real number between $[0,1]$.

4.8. Local search

An agent $Z_i(g)$ is optimized based on Eq. (10), to obtain an agent $A_i(g)$. The strategy used is based on *Guided local search* (GLS) (Voudouris & Tsang, 1995). As such, the characteristics of the GLS are represented by all the sentences in a document, such that if a document comprises n sentences, $D = \{S_1, S_2, \ldots, S_n\}$, the set of GLS characteristics equals n. The vector representing whether or not an agent X_b has some characteristic is $K_b = \{k_{b1}, k_{b2}, \ldots, k_{bn}\}$, where $k_{bi} \in \{0,1\}$. Thus, if $n = 10$, a vector $X_b = \{1, 0, 1, 0, 0, 1, 0, 1, 0, 0\}$ indicates that the agent X_b has the characteristics (or sentences) one, three, six and eight.

The costs associated with the characteristics are represented by a vector constant $C = \{c_1, c_2, \ldots, c_n\}$, calculated at the beginning of the execution of the memetic algorithm, where the cost c_i of the i-th GLS characteristic is calculated as shown in Eq. (23)

$$c_i = \sqrt{\frac{1}{q_i}} + \frac{sim_{\cos}(S_i, t)}{Max(sim_{\cos}(S_i, t), \ldots, sim_{\cos}(S_n, t)} \quad (23)$$

where q_i is the position of the characteristic (or sentence) S_1 in the document, $sim_{cos}(s_i, t)$ is the cosine similarity of the characteristic S_1 with the title, and $MAX(sim_{cos}(s_1, t), \ldots, sim_{cos}(S_n, t)))$ is the maximum cosine similarity with the title of the sentences of the document.

The penalties of the GLS are represented by a vector $P = \{p_1, p_2, \ldots, p_i, \ldots, p_n\}$, whose values are zero on initiating the search and increase when a local optimum X_0 is arrived at, such that if K_0 is the vector of characteristics of the aforementioned local optimum, the penalty value p_i of the i-th characteristic is modified according to Eq. (24)

$$p_i = \begin{cases} p_i + 1 & \text{if} \quad k_{oi} = 1 \wedge Min(d_{a1}, \ldots, d_{aj}, \ldots, d_{am}) \\ p_i & \text{otherwise} \end{cases} \quad (24)$$

where d_i is the decision function that determines whether or not the characteristic S_i should be penalized, and is calculated as in Eq. (25), d_{aj} is the decision value of the j-th characteristic whose value k_{0j} equals one, $MIN(d_{a1}, d_{a2}, \ldots, d_{am})$ is the minimum value of the decision function evaluated on the m characteristics whose value is one in K_0.

$$d_i = \frac{c_i}{1 + p_i} \quad (25)$$

The reduced objective function, meanwhile, is calculated as in Eq. (26)

$$G(X_r) = F(X_r) - \lambda * \sum_{i=1}^{n} p_i * k_{ri} \quad (26)$$

where λ is a regularization parameter that controls the relative importance of the penalties with respect to cost $F(X_r)$ of the solution.

4.9. Replacement

The optimized agent $A_i(g)$ is included in the population according to Eq. (11). As a result, in order to select the replacement agent $X_r(g)$ a restricted competition approach is used, in which first a set of m competing agents is randomly selected from the current population, $Comp = \{X_{r1}(g), X_{r2}(g), \ldots, X_{rm}(g)\}$, where $m < ps$ and where $X_r(g) \in Comp$ and fulfills the expression of Eq. (27)

$$F(X_r(g)) < F(X_{rj}(g)), \forall X_{rj}(g) \in Comp \quad (27)$$

where $X_{rj}(g) \neq X_r(g)$.

4.10. Convergence of population

The convergence of the population is evaluated after the generation of a new offspring. To determine the trend of fitness values among the agents of the current population, a set of agents Ev is defined whose fitness varies by a percentage (in this case 5%) compared to the average fitness of the current population as in Eq. (28)

$$Ev = \{X_r(g) | F(X_r(g)) \in [\mu(F) * 0.95, \mu(F) * 1.05]\} \quad (28)$$

where $\mu(F)$ is the average fitness of the current population.

If $COUNT(Ev)$ represents the number of elements in Ev, the evaluation of convergence is defined as in Eq. (29).

$$Convergence = \begin{cases} \text{true} & \text{if} \quad COUNT(Ev) \geqslant ps * 0.9 \\ \text{false} & \text{otherwise} \end{cases} \quad (29)$$

If the population converges, the population is re-initialized in a similar manner to the initialization process of the population, while maintaining a predefined amount Er of the best agents from the current population.

4.11. Stopping criterion

The execution of the memetic algorithm ends when the stop condition is met, which was established as a maximum number of evaluations of the objective function.

4.12. Scheme of memetic algorithm

Fig. 1 shows the general outline of the MA-SingleDocSum algorithm described above, which is based on the approach presented by Hao (Hao, 2012).

```
P_t(N) = Initial population of individuals randomly;
Calculate-fitness (P_t(N));
Optimization ( P_t(N), Guided-local-search);
Repeat
  P_t+1(N)=Elitist (P_t(N), E);
  For n = 1 to ((N-E)/2) do
      Selection (Father1, Ranking selection);
      Selection (Father2, Roulette Wheel);
      Offspring1=Crossover (Father1, Father2, One-point);
      Mutation (offspring1, multi-bit);
      Optimization (offspring1, Guided-local-search);
      P_t+1(N)= Restricted-competition(offspring1, P_t(N));
      Offspring2=Crossover (Father2, Father1, One-point);
      Mutation (offspring2, multi-bit);
      Optimization (offspring2, Guided-local-search);
      P_t+1(N)= Restricted-competition(offspring2, P_t(N));
  End For;
  Evaluation-convergence (P_t+1(N));
  t=t+1;
Until (maximum number of objective function evaluations);
```

Fig. 1. Scheme of the MA-SingleDocSum method.

4.13. Generation of extractive summary

After the execution of the memetic algorithm, a solution vector X_m is obtained, whose positions with values equal to one (sentences of the summary) are ordered in descending according to the function value $f(S_{m,i})$ obtained by Eq. (30), evaluated on the sentences corresponding to the same positions in the document. The features of objective function are comprised in the function $f(S_{m,i})$. The gene of the agent is then decoded to obtain the respective sentences of the document, which eventually form the generated summary.

$$f(S_{m,i}) = \sqrt{\frac{1}{q_i}} + sim_{\cos}(S_i, t) + \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}} + \sum_{S_j \in Summary} w(S_i, S_j) + \sum_{j=i+1}^{n} [sim_{\cos}(D, S_i) + sim_{\cos}(D, S_j)] \quad (30)$$

where $S_{m,i}$ is the i-th sentence of the document represented by the i-th position of the solution vector X_m y α es $(l(S_i) - \mu(l))/std(l)$.

5. Experiment and evaluation

This section presents the evaluation and comparison of the MA-SingleDocSum method with other state of the art methods.

5.1. Datasets

For the evaluation of MA-SingleDocSum, data sets from the Document Understanding Conference (DUC) for the years 2001 and 2002 were used, a product of research by the National Institute of Standards and Technology (http://www-nlpir.nist.gov) in the area of automatic text summarization. These files consist of news reports in English, taken from newspapers and news agencies such as the Financial Times, Associated Press and the Wall Street Journal. The DUC2001 data collection consists of 30 sets of approximately 10 documents from news reports in English, consisting of 309 articles that cover such topics as natural disasters, biographical information, and so on (http://trec.nist.gov/overview.html). Each set is accompanied by reference summaries for single and multiple documents. The reference summaries for a single document comprise approximately 100 words. The DUC2002 collection, meanwhile, consists of 567 documents in 59 sets. As with DUC2001,

each set presents reference summaries for single and multiple documents, having a length of about 100 words (see Table 1).

5.2. Preprocessing

Before moving to the automatic generation of a summary, a preprocessing of the document is performed that includes linguistic techniques such as segmentation of sentences, removal of stop words, removal of upper case and punctuation marks, stemming and indexing (Manning & Raghavan, 2008).

5.2.1. Segmentation

The segmentation process consists of dividing the text into meaningful units, in this case sentences (Manning & Raghavan, 2008). For this, an open source segmentation tool called "splitta" (http://code.google.com/p/splitta) is used.

5.2.2. Stopwords

Stopwords are those words which, due to their low semantic content, do not contribute to distinguishing the most important sentences in a text (Manning & Raghavan, 2008), for example prepositions, articles, pronouns, etc. These words are very common within a text and are considered noisy terms or negative dictionary, so that their removal can be really helpful before the execution of a natural language processing task. Such removal is usually performed by word filtering with the aid of a list of stopwords.

In this work, we used the list built for the SMART (ftp://ftp.cs.cornell.edu/pub/smart/english.stop) information retrieval system.

5.2.3. Stemming

Stemming is a computational procedure that reduces the words with the same root, or stem, to a common form, eliminating the variable suffixes (Manning & Raghavan, 2008). Among the stemming algorithms that stand out are those of Porter and Lovins. Both perform a deletion of suffixes and go on to recode the treated text string. The Porter algorithm was used for this work.

5.2.4. Lucene

Lucene is an open source library licensed under the Apache Software Licence (http://lucene.apache.org). It aims to facilitate the indexing and searching in information retrieval tasks. It was originally implemented in Java, but has since been adapted for other programming languages such as C#, C++, Delphi, PHP, Phyton and Ruby. One of the main features of this tool is the abstraction of the documents as a set of text fields, very useful for coupling to systems based on the vector space model to represent documents. In this proposal, the Lucene library is used for indexing terms, while it also contributes to the tasks of removal of capital letters and punctuation marks, stopword removal and stemming.

5.3. Evaluation metric

Evaluating the quality of the summaries generated by the MA-SingleDocSum method proposed in this article was conducted by means of the metric provided by ROUGE toolkit (Lin, 2004) in its version 1.5.5, which has been adopted by DUC for automatic summarization evaluation. ROUGE is a tool that measures the quality of the summary by counting the overlapping units between the reference summary and the candidate summary, based on n-gram recall between a generated summary and a set of reference summaries. Eq. (31) shows the calculation of this measure.

$$\text{ROUGE} - N = \frac{\sum_{s \in Summ_{ref}} \sum_{N-gram \in S} Count_{match}(N-gram)}{\sum_{s \in Summ_{ref}} \sum_{N-gram \in S} Count(N-gram)} \quad (31)$$

where N represents the length of the n-gram $(N\text{-}gram)$ and $Count_{match}(N\text{-}gram)$ is the maximum number of matching n-grams between a candidate summary and a set of reference summaries. The denominator in this formula is the sum of the number of n-grams in the reference summary. In these experiments N takes the value of 1 and 2, i.e. unigram metric ROUGE-1 and bigram metric ROUGE-2.

Table 1
Description of the data sets used.

	DUC2002	DUC2001
Number of topics	59	30
Number of documents	567	309
Data source	TREC	TREC
Summary length (in words)	100	100

5.4. Parameter tuning

The parameter tuning is based on the Meta Evolutionary Algorithm (Meta-EA) (Eiben & Smit, 2012; Smit & Eiben, 2009) using a version of harmony search (Cobos, Estupiñán, & Pérez, 2011). The parameters for MA-SingleDocSum are set as follows: population size $ps = 30$, mutation probability $Mr = 0.4$, optimization probability $Opr = 1$, number of agents selected by elitism $e = 1$, number of agents selected by elitism in re-initialization $er = 1$, number of competing agents in replacement $gr = 4$, GLS regularization parameter $\lambda = 0.5$.

The number of evaluations of the objective function was established in 1600 optimizations. The results presented in this section were obtained by evaluating summaries generated with 100 words, and averaging 30 runs of the algorithm, which was implemented on a Pentium 4 CPU 3.00 GHz, 2.99 GHz PC with 1 GB of RAM on Windows XP.

Regarding the process of tuning the weights of the objective function of MA-SingleDocSum, this was divided into two stages. In the first, a genetic algorithm (GA) was designed to obtain various ranges for each weight, which are evaluated in the objective function with MA-SingleDocSum to find the best combination of weights. In the second stage, this set of weights was taken in order to generate new ranges for each weight and get the best performance of the objective function. The weights found for the objective function are: $\alpha = 0.35$, $\beta = 0.35$, $\gamma = 0.29$, $\delta = 0.005$, $\rho = 0.005$; which correspond to the features of Position (P), Relationship to the title (RT), Length (L), Cohesion (CoH) and Coverage (Cov), respectively.

5.5. Comparison with different methods

The results obtained using MA-SingleDocSum were compared with other state of the art methods in extractive single-document summarization:

- UnifiedRank (Wan, 2010): This method is a *graph-based* approach in which single-document and multi-document summarizations are done at same time. This work examines the mutual influences between the two tasks and proposes a novel unified approach to simultaneous single- and multi-document summarizations.
- DE (Aliguliyev, 2009a): This method uses *differential evolution* to optimize the allocation of sentences to groups, representing an individual by means of permutations that indicate the groups where each sentence corresponding to a gene will be located. Selection of the summary sentences is done under a recursive scheme, which takes into account the degree of membership

of each sentence to the corresponding group, measuring the centrality of each sentence with respect to the group it belongs to, based on normalized google distance.

- FEOM (Steinberger & Ježek, 2006): In this work a model of *fuzzy evolutionary* optimization is proposed, which carries out document clustering. The sentences most relevant for each group are then selected to obtain the summary. FEOM uses genetic algorithms for the generation of the solution vectors with the groups, and applies three control parameters to regulate the probability of crossover and mutation of each solution.
- NetSum (Svore et al., 2007): This approach uses the RankNet learning algorithm. It trains a pair-based sentence ranker to score every sentence in the document and identify the most important sentences. This method realizes automatic summarization based on *neural nets*.
- CRF (Shen et al., 2007): This work treats the summarization task as a sequence labeling problem. In this view, each document is a sequence of sentences and the summarization procedure labels the sentences using 1 and 0. This work uses *Conditional Random Fields* (CRF).
- QCS (Dunlavy, O'Leary, Conroy, & Schlesinger, 2007): The summaries are produced using a *Hidden Markov Model* (HMM) to compute the probability that each sentence is a good summary sentence. The highest probability sentences are chosen for the summary. The HMMs uses features based upon "signature" and "subject" terms occurring in the sentences. The signature terms are the terms that are more likely to occur in the document (or document set) than in the corpus at large.
- SVM (Yeh et al., 2005): This work proposes two methods to achieve automatic text summarization: the Modified Corpus Based Approach (MCBA) and the LSA-based text relationship map (TRM) approach. The first is based on a score function combined with the analysis of salient features, and a *genetic algorithm* is employed to discover suitable combinations of feature weights. The second exploits *LSA* and a *TRM* to derive semantically salient structures from a document.
- Manifold Ranking (Wan, 2010): The *manifold-ranking* process can naturally make full use of both the relationships among all the sentences in the documents and the relationships between the given topic and the sentences. The ranking score is obtained for each sentence in the manifold-ranking process to denote the biased information richness of the sentence. Then a greedy algorithm is employed to impose diversity penalty on each sentence. The summary is produced by choosing the sentences with both high biased information richness and high information novelty.

Table 2 presents the results of ROUGE measures for MA-SingleDocSum and other state of the art methods for the DUC2001 dataset; Table 3 presents the information for DUC2002.

According to the data presented in Table 2 and Table 3, it can be seen that MA-SingleDocSum in the measure ROUGE-2 outperforms all other methods for both DUC2001 and DUC2002; in the measure ROUGE-1 for DUC2002, MA-SingleDocSum is second only to UnifiedRank; and in the case of DUC2001, it was outperformed by five other methods

Table 4 shows the improvement produced by MA-SingleDocSum with respect to the other methods, in the measure ROUGE-2 on DUC2001 and DUC2002 data, calculated by means of Eq. (32). Comparison with FEOM on DUC2001 dataset shows that MA-SingleDocSum improves performance by 8.59%, and compared to Unified Rank on DUC2002 dataset, MA-SingleDocSum improves performance by 6.24%.

$$\frac{Proposed\ method - Other\ Method}{Other\ Method} \times 100 \tag{32}$$

Table 5 shows the improvement obtained by DE in the measure ROUGE-1 on DUC2001 data with respect to the other methods. As can be seen, in comparison with FEOM, DE improves performance by 0.27%.

Table 6 shows the improvement obtained by UnifiedRank in the measure ROUGE-1 on DUC2002 data with respect to the other methods. In comparison with MA-SingleDocSum, UnifiedRank improves performance by 0.41%.

Given that ROUGE-2 evaluates matching bi-grams between the generated summary and the reference summary, and ROUGE-1 evaluates uni-grams, the results for MA-SingleDocSum indicate a better performance compared to state of the art works.

Because the results do not identify which method gets the best results on both data sets, a unified ranking of all methods is proposed, taking into account the position each method occupies for each measure. To obtain the resulting ranks of the methods we transformed Table 2 and Table 3 into one, shown in Table 7. The resultant rank in this table (last column) was computed according to the formula of Eq. (33) (Aliguliyev, 2009b):

$$Ran(method) = \sum_{r=1}^{9} \frac{(9 - r + 1)R_r}{9} \tag{33}$$

Table 2
ROUGE scores of the methods on DUC2001 data.

Method	ROUGE-1	ROUGE-2
MA-SingleDocSum	0.44862 (6)	**0.20142 (1)**
DE	**0.47856 (1)**	0.18528 (3)
UnifiedRank	0.45377 (5)	0.17646 (6)
FEOM	0.47728 (2)	0.18549 (2)
NetSum	0.46427 (3)	0.17697 (5)
CRF	0.45512 (4)	0.17327 (7)
QSC	0.44852 (7)	0.18523 (4)
SVM	0.44628 (8)	0.17018 (8)
Manifold Ranking	0.43359 (9)	0.16635 (9)

Table 3
ROUGE scores of the methods on DUC2002 data.

Method	ROUGE-1	ROUGE-2
MA-SingleDocSum	0.48280 (2)	**0.22840 (1)**
DE	0.46694 (3)	0,12368 (5)
UnifiedRank	**0.48487 (1)**	0,21462 (2)
FEOM	0.46575 (4)	0,12490 (4)
NetSum	0.44963 (5)	0.11167 (6)
CRF	0.44006 (7)	0.10924 (7)
QSC	0.44865 (6)	0.18766 (3)
SVM	0.43235 (9)	0.10867 (8)
Manifold Ranking	0.42325 (8)	0.10677 (9)

Table 4
Comparison of MA-SingleDocSum with others methods (ROUGE-2).

Methods	Improvement obtained by MA-SingleDocSum (%)	
	DUC2001	DUC2002
DE	8.71	84.67
UnifiedRank	14.14	6.42
FEOM	8.59	82.87
NetSum	13.82	104.53
CRF	16.25	109.08
QSC	8.74	21.71
SVM	18.36	110.18
Manifold Ranking	21.08	113.92

Table 5
Comparison of DE with others methods on DUC2001 (ROUGE-1).

Method	Improvement obtained by DE method (%)
	DUC2001
MA-SingleDocSum	6.67
Unified Rank	5.46
FEOM	0.27
NetSum	3.08
CRF	5.15
QSC	6.70
SVM	7.23
Manifold Ranking	10.37

Table 6
Comparison of UnifiedRank with others methods, DUC2002 (ROUGE-1).

Method	Improvement obtained by the Unified Rank method (%)
	DUC2002
MA-SingleDocSum	0.41
DE	3.82
FEOM	4.09
NetSum	7.82
CRF	10.16
QSC	8.05
SVM	12.13
Manifold Ranking	14.54

Table 7
The resultant rank of the methods.

Method	R_r=									Resultant rank
	1	2	3	4	5	6	7	8	9	
MA-SingleDocSum	2	1	0	0	0	1	0	0	0	3.33
DE	1	0	2	0	1	0	0	0	0	3.11
FEOM	0	2	0	2	0	0	0	0	0	3.11
UnifiedRank	1	1	0	0	1	1	0	0	0	2.89
NetSum	0	0	1	0	2	1	0	0	0	2.33
QSC	0	0	1	1	0	1	1	0	0	2.22
CRF	0	0	0	1	0	0	3	0	0	1.67
SVM	0	0	0	0	0	0	0	3	1	0.78
Manifold Ranking	0	0	0	0	0	0	0	1	3	0.56

where R_r denotes the number of times the method appears in the rth rank. The number 9 represents the total number of methods with which the comparison was carried out.

Considering the results of Table 7, the following can be observed:

- The MA-SingleDocSum method ranks first in the unified ranking, beating methods like DE and UnifiedRank, despite the fact that in the measure ROUGE-1, these methods obtained better values.
- The performance of the DE and FEOM is the same, these methods – like MA-SingleDocSum – also address the automatic text summarization as an optimization problem, but DE and FEOM use the concept of clustering in the representation of the solution.
- The graph-based UnifiedRank outperforms supervised methods such as NetSum and CRF, probabilistic methods such as QCS, algebraic reduction methods such as SVM and Manifold Ranking. However, it is outperformed by the methods based on evolutionary models.
- The supervised methods – NetSum based on neural networks and CRF based on sequence labeling – just as with QSC based on probabilistic models, outperform those of algebraic reduction such as SVM and Manifold Ranking.

The experimental results indicate that optimization combining population-based global search with local search heuristics for each agent, thereby coupling genetic evolution with individual learning as in the case of MA-SingleDocSum is a most promising line of research. In MA-SingleDocSum, representation of the solutions is binary, indicating the presence or absence of the sentence in the summary, while in the case of DE and FEOM methods the representation is real, indicating the group to which the sentence belongs. A process is later carried out for the selection of the sentences to make up the summary. This requires that the methods DE and FEOM perform an additional process to obtain the summary, a process that is not necessary in the case of MA-SingleDocSum.

It is important to note that the ranking does not take into account the percentage of improvement. In the case of ROUGE-2, MA-SingleDocSum with DUC2002 improves on DE and FEOM with the considerably high percentages of 84.67% and 82.87%, respectively, and with DUC2001 improves on UnifiedRank, DE and FEOM by 14.14%, 8.71% and 8.59%, respectively. On the other hand, MA-SingleDocSum is outperformed by smaller percentages in the measure ROUGE-1, of 6.67% and 0.41%, respectively for DUC2001 and DUC2002. As such, if the improvement percentages of MA-SingleDocSum over the other methods are taken into account, the difference in resultant rank would be much greater.

6. Conclusions

In this paper a memetic algorithm for the extractive summarization of a single document (MA-SingleDocSum) is proposed. This algorithm addresses the generation of extractive summaries as a binary optimization problem. But unlike from methods of state of the art, in this proposal is combined the population-based global search with a local search heuristic (memetic approach). The local search heuristic exploits the problem knowledge for redirect the search toward a best solution. The MA-SingleDocSum method was compared with others the state of the art methods, using ROUGE measures on the datasets DUC2001 and DUC2002. And the results had shown that MA-SingleDocSum outperforms the state of the art methods.

The proposed algorithm is comprised of reproductive selection operators based on the range (Rank-based) for choosing the father of a new offspring, which attempts to avoid dominance by the fittest agents, encouraging diversity in the population; Roulette wheel selection for choosing the mother, through which selective pressure is greatly favored; one-point crossover to generate the offspring, which also favors selective pressure by retaining much of the genetic material from the parents; Multi-bit mutation that favors population diversity; and Restricted competition replacement whose adaptation fosters diversity, with random choosing of the group and selective pressure to eliminate the worst.

The local optimization algorithm used in MA-SingleDocSum is Guided Local Search, which maintains an exploitation strategy directed by the information of the problem, improving the quality of the summaries obtained in relation to other local optimization techniques evaluated. This is because it incorporates strategies to exploit the best characteristics of sentence evaluation. In that sense, in its configuration the sentences of the document are defined as the search characteristics, the cost of the characteristics is calculated as a combination of the Position and the Relationship with the Title factors, and for the regularization parameter (λ).

An objective function for the method MA-SingleDocSum was defined, formed by features such as: Position, Relationship with the Title, Length, Cohesion, and Coverage, and which proved effective in selecting relevant sentences from a document, the best results being obtained with MA-SingleDocSum in comparison with other state of the art methods. Following the process of tuning the weights of the objective function, it was found that the most influential characteristics are Position, Relationship with the Title and Length.

The MA-SingleDocSum method proposed was evaluated by means of the measures ROUGE-1 and ROUGE-2 on the data sets DUC2001 and DUC2002. When compared against other state of the art evolutionary methods, with the measure ROUGE-2, MA-SingleDocSum presents the best results, outperforming FEOM, the best prior method by 8.59% with DUC2001 and UnifiedRank by 6.42% with DUC2002. In the case of the measure ROUGE-1 for the dataset DUC2001 it is outperformed by the DE by 6.67% and by UnifiedRank at 0.41% with DUC2002. In addition, in the unified ranking of all methods, MA-SingleDocSum ranks first, outperforming all other methods.

Future work is expected to involve the study of other schemes of selection, crossover, mutation, replacement, and local search in order to obtain a new configuration for the memetic algorithm that improves upon the results presented in this paper. It is also intended to evaluate the proposed memetic algorithm with other data sets, other than news items, in order to analyze its performance under other conditions. Future work will further seek to perform non-parametric tests with detailed data for each topic of the data sets, allowing statistical evaluation of the significance of the results. Finally, it is planned to make a proposal for query-based single-document summarization using the approach developed in this paper and apply it to generate snippets that are then displayed in a web search engine.

References

Abuobieda, A., Salim, N., Kumar, Y., & Osman, A. (2013). An improved evolutionary algorithm for extractive text summarization. In A. Selamat, N. Nguyen, & H. Haron (Eds.). *Intelligent information and database systems* (Vol. 7803, pp. 78–89). Berlin Heidelberg: Springer.

Aliguliyev, R. M. (2009a). A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications, 36*, 7764–7772.

Aliguliyev, R. M. (2009b). Performance evaluation of density-based clustering methods. *Information Sciences, 179*, 3583–3602.

Aone, C., Okurowski, M. E., Gorlinsky, J., & Larsen, B. S. (1999). Trainable, scalable summarization using robust NLP and machine learning. In I. Mani & M. T. Maybury (Eds.), *Advances in automatic text summarization* (pp. 71–80).

Barzilay, R., & Elhadad, M. (1997). Using lexical chains for text summarization. In *Proceedings of the ACL/EACL'97 workshop on intelligent scalable text summarization* (pp. 10–17). Madrid, Spain.

Binwahlan, M. S., Salim, N., & Suanmali, L. (2009). Swarm based text summarization. In *Proceedings of the international association of computer science and information technology – spring conference. IACSITSC '09* (pp. 145–150).

Binwahlan, M. S., Salim, N., & Suanmali, L. (2010). Fuzzy swarm diversity hybrid model for text summarization. *Information Processing and Management, 46*, 571–588.

Bossard, A., Genereux, M., & Poibeau, T. (2008). Description of the LIPN systems at TAC 2008: Summarizing information and opinions. In *Notebook papers and results, text analysis conference (TAC-2008)*.

Carpineto, C., Osinski, S., Romano, G., & Weiss, D. (2009). A survey of Web clustering engines. *ACM Computing Surveys, 41*, 1–38.

Cobos, C., Estupiñán, D., & Pérez, J. (2011). GHS + LEM: Global-best harmony search using learnable evolution models. *Applied Mathematics and Computation, 218*, 2558–2578.

Cobos, C., Montealegre, C., Mejía, M., Mendoza, M., & León, E. (2010). Web document clustering based on a new niching memetic algorithm, term-document matrix and Bayesian information criterion. In *Proceedings of the IEEE congress on evolutionary computation (IEEE CEC)* (pp. 4629–4636). Barcelona, Spain: IEEE.

Conroy, J., & O'leary, D. (2001). Text summarization via hidden Markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 406–407). New Orleans, Louisiana, United States: ACM.

Dehkordi, P.-K., Kumarci, F., & Khosravi, H. (2009). Text summarization based on genetic programming. In *Proceedings of the international journal of computing and ICT research* (Vol. 3, pp. 57–64).

Dunlavy, D. M., O'Leary, D. P., Conroy, J. M., & Schlesinger, J. D. (2007). QCS: A system for querying, clustering and summarizing documents. *Information Processing & Management, 43*, 1588–1605.

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

Eiben, A. E., & Smit, S. K. (2012). Evolutionary algorithm parameters and methods to tune them. In Y. Hamadi, E. Monfroy, & F. Saubion (Eds.), *Autonomous search* (pp. 15–36). Berlin Heidelberg: Springer.

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

García-Hernández, R., & Ledeneva, Y. (2013). Single extractive text summarization based on a genetic algorithm. In J. Carrasco-Ochoa, J. Martínez-Trinidad, J. Rodríguez, & G. Baja (Eds.). *Pattern recognition* (Vol. 7914, pp. 374–383). Berlin Heidelberg: Springer.

Gong, Y. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*.

Gupta, V., Chauhan, P., & Garg, S. (2012). An statistical tool for multi-document summarization. *International Journal of Scientific and Research Publications, 2*, 1–5.

Hao, J.-K. (2012). Memetic algorithms in discrete optimization. In F. Neri, C. Cotta, & P. Moscato (Eds.). *Handbook of memetic algorithms* (Vol. 379, pp. 73–94). Berlin Heidelberg: Springer.

Ježek, K., & Steinberger, J. (2008). Automatic text summarization (the state of the art 2007 and new challenges). In *Znalosti 2008* (pp. 1–12). Bratislava, Slovakia.

Kumaresh, N., & Ramakrishnan, B. (2012). Graph based single document summarization. In R. Kannan & F. Andres (Eds.). *Data engineering and management* (Vol. 6411, pp. 32–35). Berlin Heidelberg: Springer.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval*. Seattle, Washington, United States: ACM.

Lee, J.-H., Park, S., Ahn, C.-M., & Kim, D. (2009). Automatic generic document summarization based on non-negative matrix factorization. *Information Processing & Management, 45*, 20–34.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL-04 workshop on text summarization branches out* (pp. 74–81). Barcelona, Spain.

Lin, C.-Y., & Hovy, E. (1997). Identifying topics by position. In *Proceedings of the fifth conference on applied natural language processing* (pp. 283–290). San Francisco, CA, USA.

Litvak, M., Last, M., & Friedman, M. (2010). A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 927–936). Uppsala, Sweden: Association for Computational Linguistics.

Lloret, E., & Palomar, M. (2012). Text summarisation in progress: A literature review. *Artificial Intelligence Review, 37*, 1–41.

Louis, A., Joshi, A., & Nenkova, A. (2010). Discourse indicators for content selection in summarization. In *Proceedings of the 11th annual meeting of the special interest group on discourse and dialogue* (pp. 147–156). Tokyo, Japan: Association for Computational Linguistics.

Manning, C. D., Raghavan, P., & Schtze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Marcu, D. (1998). Improving summarization through rhetorical parsing tuning. In *Proceedings of the sixth workshop on very large corpora* (pp. 206–215). Montreal, Canada.

Mihalcea, R., & Tarau, P. (2004). Text-rank: Bringing order into texts. In *Proceeding of the conference on empirical methods in natural language processing*. Barcelona, Spain.

Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In C. C. Aggarwal & C. Zhai (Eds.), *Mining text data* (pp. 43–76). US: Springer.

Neri, F., & Cotta, C. (2012). Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation, 2*, 1–14.

Ono, K., Sumita, K., & Miike, S. (1994). Abstract generation based on rhetorical structure extraction. *Proceedings of the 15th conference on computational linguistics* (Vol. 1, pp. 344–348). Kyoto, Japan: Association for Computational Linguistics.

Porselvi, A., & Gunasundari, S. (2013). Survey on web page visual summarization. *International Journal of Emerging Technology and Advanced Engineering, 3*, 26–32.

Qazvinian, V., Sharif, L., & Halavati, R. (2008). Summarising text with a genetic algorithm-based sentence extraction. *International Journal of Knowledge Management Studies (IJKMS), 4*, 426–444.

Radev, D. R., Jing, H., StyÅ, M. G., & Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management, 40*, 919–938.

Shareghi, E., & Hassanabadi, L. S. (2008). Text summarization with harmony search algorithm-based sentence extraction. In *Proceedings of the 5th international conference on soft computing as transdisciplinary science and technology* (pp. 226–231). Cergy-Pontoise, France: ACM.

Shen, D., Sun, J.-T., Li, H., Yang, Q., & Chen, Z. (2007). Document summarization using conditional random fields. In *Proceedings of the 20th international joint conference on artificial intelligence* (pp. 2862–2867). Hyderabad, India: Morgan Kaufmann Publishers Inc.

Silla, J., Nascimento, C., Pappa, G. L., Freitas, A. A., & Kaestner, C. A. A. (2004). Automatic text summarization with genetic algorithm-based attribute selection. *Lecture Notes in Artificial Intelligence, 3315*, 305–314.

Sivanandam, S. N., & Deepa, S. N. (2008). *Introduction to genetic algorithms*. Springer Publishing Company, Incorporated.

Smit, S. K., & Eiben, A. E. (2009). Comparing parameter tuning methods for evolutionary algorithms. In *IEEE congress on evolutionary computation, 2009. CEC '09* (pp. 399–406).

Song, W., Cheon Choi, L., Cheol Park, S., & Feng Ding, X. (2011). Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization. *Expert Systems with Applications, 38*, 9112–9121.

Spärck Jones, K. (2007). Automatic summarising: The state of the art. *Information Processing & Management, 43*, 1449–1481.

Steinberger, J., & Jezek, K. (2004). Using latent semantic analysis in text summarization and summary evaluation. In *Proceedings of the 7th international conference ISIM*.

Steinberger, J., & Ježek, K. (2006). *Sentence compression for the LSA-based summarizer* (pp. 141–148).

Svore, K., Vanderwende, L., & Burges, C. (2007). Enhancing single-document summarization by combining RankNet and third-party sources. In *Proceedings of the EMNLP-CoNLL* (pp. 448–457).

Voudouris, C., & Tsang, E. (1995). Guided local search. In Technical Report CSM-247. Colchester: University of Essex.

Wan, X. (2010). Towards a unified approach to simultaneous single-document and multi-document summarizations. In *Proceeding of the 23rd international conference on computational linguistics (Coling 2010)* (pp. 1137–1145). Beijing.

Wei, F., Li, W., & Liu, S. (2010). IRANK: A rank-learn-combine framework for unsupervised ensemble ranking. *Journal of the American Society for Information Science and Technology, 61*(6), 1232–1243.

Wong, K.-F., Wu, M., & Li, W. (2008). Extractive summarization using supervised and semi-supervised learning. *Proceedings of the 22nd international conference on computational linguistics* (Vol. 1, pp. 985–992). Manchester, United Kingdom: Association for Computational Linguistics.

Yeh, J.-Y., Ke, H.-R., Yang, W.-P., & Meng, I. H. (2005). Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing & Management, 41*, 75–95.