Applied Soft Computing 113 (2021) 107915

Contents lists available at ScienceDirect

Applied Soft Computing

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

Sentiment-oriented query-focused text summarization addressed with a multi-objective optimization approach

Jesus M. Sanchez-Gomez [a,*], Miguel A. Vega-Rodríguez [a], Carlos J. Pérez [b]

[a] Department of Computer and Communications Technologies, University of Extremadura, Campus Universitario s/n, 10003 Cáceres, Spain
[b] Department of Mathematics, University of Extremadura, Campus Universitario s/n, 10003 Cáceres, Spain

ARTICLE INFO

Article history:
Received 28 January 2021
Received in revised form 26 July 2021
Accepted 7 September 2021
Available online 22 September 2021

Keywords:
Query-focused summarization
Sentiment-oriented summarization
Sentiment analysis
Multi-objective optimization
Crow search algorithm

ABSTRACT

Nowadays, the automatic text summarization is a highly relevant task in many contexts. In particular, query-focused summarization consists of generating a summary from one or multiple documents according to a query given by the user. Additionally, sentiment analysis and opinion mining analyze the polarity of the opinions contained in texts. These two issues are integrated in an approach to produce an opinionated summary according to the user's query. Thereby, the query-focused sentiment-oriented extractive multi-document text summarization problem entails the optimization of different criteria, specifically, query relevance, redundancy reduction, and sentiment relevance. An adaptation of the metaheuristic population-based crow search algorithm has been designed, implemented, and tested to solve this multi-objective problem. Experiments have been carried out by using datasets from the Text Analysis Conference (TAC) datasets. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics and the Pearson correlation coefficient have been used for the performance assessment. The results have reported that the proposed approach outperforms the existing methods in the scientific literature, with a percentage improvement of 75.5% for ROUGE-1 score and 441.3% for ROUGE-2 score. It also has been obtained a Pearson correlation coefficient of +0.841, reporting a strong linear positive correlation between the sentiment scores of the generated summaries and the sentiment scores of the queries of the topics.

1. Introduction

Nowadays, the size of digital information on the Internet is huge, and it follows growing. Besides, the Internet users are characterized by wanting to obtain specific information about a determined topic as quickly as possible, but the large volume of existing digital information complicates to carry out this task. One method to get the most important information is through text mining [1]. Using tools based on these approaches, it is possible to retrieve and summarize the most relevant information from a set of digital documents. Furthermore, the study of the users' opinion about news, political and social events, products preferences, and marketing campaigns, among other topics, is also another aspect that is currently gaining great relevance in many fields. In this respect, the area of sentiment analysis and opinion mining deals with the computational treatment of opinion and sentiment in order to analyze the polarity and the feelings shown in digital texts [2].

In the scientific literature, there are many types of automatic summarization methods, that can be classified in several ways. A summarization method can be generic, if no information is required from the user [3], or query-focused, in which case it is necessary that the user provides certain information as a query [4]. Secondly, summarization methods can be abstractive or extractive: while an abstractive method produces a summary that can include text that does not exist in the original source [5], an extractive method just selects sentences from the text [6]. In addition, extractive methods are commonly based in the vector-based word method and use term-weighting schemes and similarity measures [7]. Another classification for summarization methods is single-document or multi-document based on the number of documents in the source text [8]. Moreover, summarization methods can also be mathematically formulated as an optimization approach that can be single-objective or multi-objective. In single-objective approaches, a unique objective function that includes all the subjectively weighted criteria is optimized [9]. On the other hand, multi-objective approaches can simultaneously optimize many objective functions without weighting [10]. In this last case, it is necessary to apply some method to reduce the Pareto front to a single solution [11]. A more recent type of summarization method is the opinion

* Corresponding author.
E-mail addresses: jmsanchezgomez@unex.es (J.M. Sanchez-Gomez), mavega@unex.es (M.A. Vega-Rodríguez), carper@unex.es (C.J. Pérez).

https://doi.org/10.1016/j.asoc.2021.107915

or sentiment summarization. Sentiment-oriented summarization tries to obtain both the most relevant information and the general opinion orientation from a text [12]. In contrast to conventional methods, this one uses both text mining methods and natural language processing tools to generate a summary that takes into account the relevant information and the sentiment orientation.

In this paper, the query-focused sentiment-oriented extractive multi-document text summarization problem has been formulated as a multi-objective optimization problem. To the best of the authors' knowledge, this is the first time that this problem has been addressed with a multi-objective optimization approach. For solving it, the Query-focused Sentiment-Oriented Multi-Objective Crow Search Algorithm (QSO-MOCSA) has been designed, implemented, and tested. This algorithm optimizes simultaneously the objective functions of query relevance, redundancy reduction, and sentiment relevance. The experimentation has been carried out with Text Analysis Conference (TAC) datasets [13]. The obtained results have been assessed with the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics [14] and the Pearson correlation coefficient. Therefore, the principal contributions of this paper are:

- The query-focused sentiment-oriented extractive multi-document text summarization problem has been addressed as a multi-objective optimization problem. To the best of the authors' knowledge, it is the first time that this problem has been addressed in this way.
- The criteria of query relevance, redundancy reduction, and sentiment relevance have been defined as the three objective functions to be simultaneously maximized in order to produce a query-focused sentiment-oriented summary.
- For the first time, the metaheuristic population-based crow search algorithm has been adapted for the query-focused sentiment-oriented extractive multi-document text summarization task.
- A Query-focused Sentiment-Oriented Multi-Objective Crow Search Algorithm (QSO-MOCSA) has been designed, implemented, and tested for solving the problem.
- Using TAC datasets, the ROUGE values and the sentiment scores obtained for the generated summaries have been statistically analyzed.
- The results of the new approach show an important improvement compared with the existing methods in the scientific literature.

The remainder of this paper is organized as follows. In Section 2, the related work is presented. Section 3 defines the query-focused sentiment-oriented extractive multi-document text summarization problem. In Section 4, the main steps of QSO-MOCSA, its operators, and the method considered for selecting the final solution are described. Section 5 contains the description of the datasets, evaluation metrics, and parameter settings; and the results obtained along with the comparison with the other existing methods are also included. Finally, in Section 6 the conclusions and the future work are reported.

2. Related work

The scientific literature contains a substantial amount of works focused on automatic text summarization and sentiment analysis. Some surveys have been published very recently on these topics: automatic text summarization [15] and sentiment analysis [16]. In this section, the interest is focused on approaches that consider query-focused sentiment-oriented summaries and have experimented with standard datasets and used comparable evaluation metrics. Furthermore, the field of evolutionary computation has progressed considerably in recent years, as it is reported in [17], proposing a large number of algorithms that have been successfully used in different problems, such as in [18] and [19].

Firstly, [20] presented the CCNU (Central China Normal University, China) method, a sentiment orientation recognition module that uses the WordNet-based similarity vector to extract the part-of-speech terms' similarity. The main steps of this method are: content extracting and sentence splitting, resolution of the syntactic-based anaphora and sentence compression, computation of the polarity score of all terms for the document set, computation of the sentence scores (with a query-related score and a query-independent score), and dynamic sentence choosing with redundancy removal.

In [21], the IIITSum (International Institute of Information Technology, Hyderabad, India) system was presented. It leverages on the summarization engine and uses a classification approach for finding the opinionated sentences and the polarity of these opinions. The architecture also contains the following stages: first, analysis of the query; second, opinion mining with polarity detection; and, finally, the summarization of the opinionated sentences. In addition, this system also uses query-dependent and query-independent features to rank the sentences.

The ITALICA (Itálica Research Group, University of Seville, Spain) system was proposed in [22]. It is based on the combination of the snippets for the summary generation. The tasks followed by the system architecture are: documents preprocessing, retrieval of opinionated sentences according to the given query, clause extraction for minimizing the inconsistencies between the query and the summary, the sentence selection process based on the redundancy applying a clustering process, and, finally, a sentence transformation to generate the summary.

In [23], the NUS (National University of Singapore, Singapore) system was presented. This system uses the opinion snippets for locating the source sentences based on the cosine similarity, also expanding the context by considering their previous and following sentences with two purposes: the first one, to identify more accurately the polarities of the sentences and of the query; and, the second one, to include the context of the added sentences in a selective way, matching those that will be then synthesized by the summary generator.

The PolyU (Hong Kong Polytechnic University, Hong Kong) system was presented in [24]. It is built on a feature-based framework that implements the extraction of sentences from the original documents to produce a summary. The summary is generated following three modules: the candidate sentence retrieval module, that transforms the input documents into candidate sentences; the sentence scoring module, that identifies the salient sentences according to characteristics (such as the centroid, the similarity to the query, the positive or negative sentiment score, the positive or negative orientation, and the position); and, the summary generation module, that produces the summary from the selected sentences.

In [25], the IITSummarizers (Indian Institute of Technology, Kharagpur, India) model was proposed. It is based on a statistical model for extracting the relevant opinions and the subsequent summarization process. This system is divided in three steps: first, the text extraction step, that develops a simple parser to extract the relevant text from the input documents; second, the opinion sentence extraction step, where a module extracts the sentiment sentences; and, finally, the summarization step, where the extractive summarization module produces the opinion summary.

Finally, the QMOS (Query-based Multi-document Opinion-oriented Summarization) method was recently proposed in [26]. This method uses a combination of sentiment analysis and summarization approaches. The two principal stages carried out are the following ones. The first one performs the sentiment analysis, also calculating the sentiment score for every sentence, the polarity recognition of every sentence and of the user's query, and the selection of the sentences with the same orientation as the user's query. The second stage consists of the summarizer, that determines the sentences that are relevant to the query by using a graph-based ranking model.

All the reviewed methods evaluated their summaries with the ROUGE metrics. Specifically, they used ROUGE-1 and ROUGE-2 scores. As for the datasets used, all the proposals performed their experiments with TAC2008 datasets. For these reasons, both these two ROUGE scores and the TAC2008 datasets have been used in this work for carrying out the experimentation.

3. Problem statement

The query-focused sentiment-oriented extractive multi-document text summarization problem is described in this section. It presents the preprocessing of the document collection, the analysis of the text similarity, the sentiment analysis, and, finally, the formulation of the multi-objective optimization problem.

3.1. Preprocessing

The document collection that contains the source documents has to be preprocessed in the first place. In this way, the text contained in the documents is normalized. The steps performed are:

1. Sentence segmentation. The sentences included in the document collection must be separated individually, determining the beginning and the ending of each one.
2. Word tokenization. The words from every sentence are divided with a token, as can be the blank space.
3. Stop words removal. The words that have no relevant meaning, such as prepositions, conjunctions, articles, and others, are removed from the sentences. Moreover, according to [27], stop words have not influence in the sentiment of a sentence since they have no sentiment score. The inventory of stop words is provided by the ROUGE package [28].
4. Part-of-speech tagging. This step lies in tagging each word of the sentences with its corresponding morphological category (noun, verb, adjective, adverb...). It is a very important task since the tag relies on the word's context and the sentiment score of the word depends on the tag. The Natural Language ToolKit (NLTK) has been used [29] for carrying out this task. NLTK is an open-source platform which supplies tools for working with human language data. Particularly, the part-of-speech tagging of the words has been performed with the WordNet lemmatizer [30].
5. Stemming. The last preprocessing step consists of extracting the roots of the words by using the Porter stemming algorithm [31]. Thus, the words that have a common lexical root can be processed as the same term.

3.2. Text similarity analysis

In extractive multi-document text summarization, the most commonly method used for analyzing the text similarity is the vector-based word method. This method is based on the representation of a sentence as a vector of words. Furthermore, to measure the similarity between sentences, they are compared by means of the cosine similarity measure, which is presented next.

Let $D = \{d_1, d_2, \ldots, d_N\}$ be a set of documents or document collection that contains N documents. D can also be represented as a set of sentences, $D = \{s_1, s_2, \ldots, s_n\}$, that contains the n sentences of the document collection. Now, let $T = \{t_1, t_2, \ldots, t_m\}$ be a set which contains the m distinct terms in the document collection. Thus, each sentence $s_i \in D$ is represented as an m-dimensional vector as $s_i = (w_{i1}, w_{i2}, \ldots, w_{im}), i = 1, 2, \ldots, n$, where each component is related to the weight of the term t_k in the sentence s_i. The value of this weight w_{ik} is calculated following the *term-frequency inverse-sentence-frequency* ($tf - isf$) scheme [32], as shown in the next equation:

$$w_{ik} = tf_{ik} \cdot \log \frac{n}{n_k}, \tag{1}$$

where tf_{ik} refers to the term frequency and is the number of occurrences of the term t_k in the sentence s_i, and $\log \frac{n}{n_k}$ concerns to the inverse sentence frequency, being n_k the total of sentences in which the term t_k is present.

Finally, the cosine similarity quantifies the resemblance between two sentences s_i and s_j, and it is calculated as:

$$cosim(s_i, s_j) = \frac{\sum_{k=1}^{m} w_{ik} \cdot w_{jk}}{\sqrt{\sum_{k=1}^{m} w_{ik}^2 \cdot \sum_{k=1}^{m} w_{jk}^2}}, \quad i, j = 1, 2, \ldots, n. \tag{2}$$

Moreover, just like a sentence, the query given by the user can also be represented as a vector of weights $Q = (q_1, q_2, \ldots, q_m)$, where each component q_k is the weight of the corresponding term k. These weights are calculated as indicated in Eq. (1).

3.3. Sentiment analysis

The sentiment analysis is in charge of recognizing the polarity of the sentences and computing their associated sentiment scores. This stage is divided in three steps: first, the sentiment dictionary is processed; second, the contextual polarity of the sentences and of the query are identified; and, third, the sentiment score is calculated for each sentence and for the query.

3.3.1. Sentiment lexicon

A sentiment lexicon is a dictionary that classifies the words according to several features, such as the part-of-speech tag, the sentiment score, and the meaning. In addition, a word may have several entries in the sentiment lexicon depending on the context.

The dictionary used in this approach has been SentiWordNet 3.0, which is a lexical resource widely applied in many research projects in the world [33]. SentiWordNet 3.0 includes 155,287 entries of words with relevant meaning, that is, only nouns, verbs, adjectives, and adverbs are included, and the entries are organized in 117,659 synsets. A synset is a category of cognitive synonyms in the WordNet lexical database that are considered semantically similar [34]. In particular, every synset from SentiWordNet 3.0 has two sentiment scores, positive and negative, with a numeric value within the range [0, 1] which specify the orientation of the words in the synset. Additionally, a word can be related to more than one synset if it has more than one meaning. Therefore, a word can have several associated sentiment scores. For this reason, a procedure is required to calculate the sentiment score of a word. The procedure followed in this work is based on the weighted average of the sentiment scores of a word [35]. As in SentiWordNet 3.0 the words are arranged in the synset depending on the frequency of use, the most frequent use will correspond to the greatest weight. Therefore, the calculation of the sentiment score of the word $word_i$ is performed as indicated in Eq. (3):

$senti(word_i)$

$$= \frac{\sum_{j=1}^{synsets^i_{num}} weight(word_i^j) \cdot \max\big(pos(word_i^j), neg(word_i^j)\big)}{\sum_{j=1}^{synsets^i_{num}} weight(word_i^j)}. \quad (3)$$

On the one hand, $synsets^i_{num}$ is the total of synsets of the ith word, and $weight(word_i^j)$ is the importance of the ith word in the jth synset, which is calculated as:

$$weight(word_i^j) = 1 - \frac{position(word_i^j)}{words^j_{num}}, \quad (4)$$

where $position(word_i^j)$ is the position (starting from zero) of the ith word in the jth synset, and $words^j_{num}$ is the total of words in the synset j. In this way, the value of the weight is 1 for the word in the first position in the synset, i.e., the most frequent use of the word corresponds to the maximum weight.

On the other hand, $pos(word_i^j)$ and $neg(word_i^j)$ are the positive and negative sentiment score of the ith word in the jth synset, respectively. The calculation of the maximum value of the sentiment score is performed as:

$$\max\big(pos(word_i^j), neg(word_i^j)\big) = \begin{cases} pos(word_i^j) \text{ if } |pos(word_i^j)| \geq |neg(word_i^j)| \\ neg(word_i^j) \text{ otherwise.} \end{cases} \quad (5)$$

Finally, the sentiment score of a word varies in the interval $[-1, 1]$: if $senti(word_i) > 0$, then the word sentiment score is positive; if $senti(word_i) < 0$, then the word sentiment score is negative; and if $senti(word_i) = 0$, then the word sentiment score is neutral. The words that do not exist in SentiWordNet 3.0 will have associated a $senti(word_i) = 0$.

3.3.2. Contextual polarity identification

With the sentiment lexicon, the words already have an associated sentiment score. However, the words appear in the text in a determined context that may influence on them by changing their polarity. For this reason, it is necessary to identify the contextual polarity of every word. The different kinds of sentences considered that may alter the polarity are the following ones:

- Objective and subjective sentences. Objective sentences are characterized by not communicating any opinion or feeling, whereas subjective sentences express opinionated information by using sentiment words [36]. Thus, objective sentences will have a sentiment score of 0.
- Conditional and interrogative sentences. These sentences do not manifest opinions or sentiment, even if there are sentiment words contained in them [37]. The conjunction "if" makes a sentence conditional, whereas the question mark "?" makes interrogative sentences. Both kinds of sentences will have a sentiment score of 0.
- Negation and but-clause sentences. These types of sentences have the peculiarity of including some negation or but-clause words that can affect to the sentiment score of other words in the sentence. These words are named sentiment shifter words, and they can modify the sentiment score of the entire sentence [38]. In the case of a negation word, the negation handling lies in inverting the sentiment score of the following word, that is, its polarity is now the opposite. The set of negation words is obtained from [39]. As for the but-clause words, the but-clause handling consists of inverting the sentiment score of all the following words, i.e., their polarities are the contraries. The set of but-clause words is supplied by [40].

3.3.3. Sentiment score calculation

The last step of the sentiment analysis is the calculation of the sentiment score of the sentences. After the assignation of the sentiment score to each word through the sentiment lexicon and the identification of the contextual polarity of the sentences, now the sentiment score of each sentence s_i can be calculated as:

$$senti(s_i) = \frac{\sum_{j=1}^{words^i_{num}} senti(word_j)}{words^i_{num}}, \quad (6)$$

being $words^i_{num}$ the number of sentiment words included in the sentence s_i. The sentiment score of the sentences is in the same range as the sentiment score of the words, i.e., $[-1, 1]$: if $senti(s_i) > 0$, then the sentence orientation is positive; if $senti(s_i) < 0$, then the sentence orientation is negative; and, if $senti(s_i) = 0$, then the sentence orientation is neutral. In the same way, the sentiment score of the query, $senti(Q)$, ranges in the interval $[-1, 1]$.

Regarding the sentiment score of a summary S, $senti(S)$, it is calculated as the average sentiment score of its opinionated sentences, i.e.:

$$senti(S) = \frac{\sum_{i=1}^{opinSens^S_{num}} senti(s_i)}{opinSens^S_{num}}, \quad (7)$$

being $opinSens^S_{num}$ the number of opinionated sentences in the summary S.

3.4. Formulation of the multi-objective optimization problem

Once the preprocessing steps have been carried out over the document collection, and the text similarity and sentiment analysis have been performed, the formulation of the query-focused sentiment-oriented extractive multi-document text summarization problem is presented: the goal is the generation of a summary composed with sentences from the document collection, that is, $S \subset D$, considering the following four aspects:

- Query relevance. The summary must contain the most relevant sentences according to the query given by the user.
- Redundancy reduction. The summary must not include sentences that resemble each other to avoid repetition of information.
- Sentiment relevance. The average sentiment score of the sentences in the summary must be similar to the sentiment score of the query given by the user.
- Length restriction. The summary must have a determined length L (with a certain tolerance).

In the way these four aspects have been stated, the first three can be defined as the objective functions to be optimized, and the fourth one as the constraint of the problem. Therefore, the formulation of the multi-objective optimization problem involves the simultaneous optimization of the query relevance, the redundancy reduction, and the sentiment relevance, also considering the length constraint.

Before illustrating the problem, it is necessary to determine two binary decision variables: $x_i, y_{ij} \in \{0, 1\}$. The first variable considers the presence ($x_i = 1$) or absence ($x_i = 0$) of the sentence s_i in the summary. Thus, the representation of the solution, i.e., the decision vector is represented as $X = (x_1, x_2, \ldots, x_n)$. The second variable contemplates the simultaneous presence ($y_{ij} = 1$) or not ($y_{ij} = 0$) of the pair of sentences s_i and s_j in the summary. Now, the objective functions can be formulated.

The first objective function, $\Phi_{QR}(X)$, is related to the criterion of the query relevance. It is described as the cosine similarity between every sentence contained in the summary $s_i \in S$ and the query vector Q. Hence, the following objective function should be maximized:

$$\Phi_{QR}(X) = \sum_{i=1}^{n} cosim(s_i, Q) \cdot x_i. \tag{8}$$

The second objective function, $\Phi_{RR}(X)$, corresponds to the criterion of the redundancy reduction. This criterion tries to minimize the cosine similarity between every pair of sentences in the summary $s_i, s_j \in S$, or what is the same, it tries to maximize the redundancy reduction as follows:

$$\Phi_{RR}(X) = \frac{1}{\left(\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} cosim(s_i, s_j) \cdot y_{ij} \right) \cdot \sum_{i=1}^{n} x_i}. \tag{9}$$

The third objective function, $\Phi_{SR}(X)$, concerns the criterion of the sentiment relevance. It is defined as the difference between the sentiment score of the query, $senti(Q)$, and the sentiment score of the summary, $senti(S)$. This difference should be minimized. As it has been explained previously, the sentiment score ranges in the interval $[-1, 1]$, so the maximum value of the difference is 2, which is obtained when the sentiment score of the query and the sentiment score of the summary are in the opposite extremes. Therefore, the following objective function must be maximized:

$$\Phi_{SR}(X) = 2 - |senti(Q) - senti(S)| . \tag{10}$$

Finally, the formulation of the multi-objective query-focused sentiment-oriented extractive multi-document text summarization problem is presented as:

$$\max \ \phi(X) = \left\{ \Phi_{QR}(X), \Phi_{RR}(X), \Phi_{SR}(X) \right\}, \tag{11}$$

$$\text{subject to } L - \varepsilon \leq \sum_{i=1}^{n} l_i \cdot x_i \leq L + \varepsilon, \tag{12}$$

being l_i the length of the sentence s_i and ε the length tolerance. The value of ε is computed as:

$$\varepsilon = \max_{i=1,2,\ldots,n} l_i - \min_{i=1,2,\ldots,n} l_i. \tag{13}$$

4. Methodology

Taking into account the problem definition presented in the previous section and the set of objective functions to be optimized, the proposed optimization algorithm (including its main operators) and the method for reducing the Pareto front to a single solution are presented next.

4.1. The proposed algorithm: QSO-MOCSA

The algorithm designed and implemented to solve the query-focused sentiment-oriented extractive multi-document text summarization problem is a multi-objective adaptation of the crow search algorithm (CSA). CSA is a population-based metaheuristic algorithm based on the intelligent behavior of the crows [41]. This species of bird is considered one of the most intelligent ones, they have the ability to make and use tools, they can communicate among them, and they can remember the hiding places of their food even months later. Even more, crows observe and follow other birds in order to know where they hide their food with the purpose of stealing it, also taking precautions to keep away from being discovered. CSA supposes a flock of crows (with size $flock_{size}$), where every crow tries to search the best solution to the problem (the best food). A crow is able to follow another crow if it thinks that the other crow has a better food (solution). A crow selects the other possible crow to follow from a set, which is limited to a size given by $flight_{len}$ (flight length).

The crow search algorithm is an optimization algorithm that has been recently published to solve engineering optimization problems [41]. Nevertheless, it has been successfully applied to many kinds of problems in the last several years, such as for the segmentation of magnetic resonance brain images [42], diagnosis of Parkinson's disease [43], non-convex economic load dispatch problem [44], feature selection [45], operation of a photovoltaic/diesel generator hybrid energy system [46], or biometric key generation [47]. Moreover, its design makes it an easy-to-understand algorithm. Furthermore, it has a reduced number of configuration parameters, facilitating its use and adaptation for multiple problems. All these aspects have contributed to the choice of the crow search algorithm to solve the tackled problem.

Now, the main steps of the proposed algorithm QSO-MOCSA (Query-focused Sentiment-Oriented Multi-Objective Crow Search Algorithm) are enumerated and described. Its pseudocode is shown in Algorithm 1, and the corresponding explanations are presented below.

Algorithm 1 QSO-MOCSA pseudocode

```
1: NDS ← ∅
2: initialize_flock(Flock)
3: evaluate_flock(Flock)
4: rank_and_crowding(Flock, flock_size)
5: for iter = 1 to iter_max do
6:   for crow = 1 to flock_size do
7:     RandomCrow ← choose_crow_randomly(Flock, flight_len)
8:     if dominate(RandomCrow, Flock[crow]) then
9:       MutatedCrow ← mutation(RandomCrow)
10:    else
11:      MutatedCrow ← mutation(Flock[crow])
12:    end if
13:    evaluate_crow(MutatedCrow)
14:    Flock[flock_size + crow] ← MutatedCrow
15:  end for
16:  rank_and_crowding(Flock, 2 · flock_size)
17:  save_flock(NDS, Flock, flock_size + 1, 2 · flock_size)
18: end for
19: save_flock(NDS, Flock, 1, flock_size)
20: NDS ← filter_solutions(NDS)
```

First of all, in line 1, the set that will contain all the non-dominated solutions (NDS) is initialized to an empty set. Then, all the $flock_{size}$ crows in the initial population/flock ($Flock$) are initialized in a random way in line 2. Each crow is a solution that represents a possible summary. The values of the objective functions of every solution are evaluated in line 3. After that, in line 4, the rank and crowding operators are performed over $flock_{size}$ crows of the flock. They are two multi-objective operators [48]: the rank operator is in charge of ranking the solutions into different Pareto fronts, considering the dominance relations among them; and the crowding operator takes into account the crowding distance among the solutions, and prefers the most distinct ones.

The steps from lines 5 to 18 are repeated until the predefined number of iterations, $iter_{max}$, is reached. Now, the steps from lines 6 to 15 are performed for each crow of the flock. Firstly, in line 7, a crow ($RandomCrow$) is randomly chosen from the flock to be possibly followed by the corresponding crow ($Flock[crow]$) to discover the position of its hidden food. This $RandomCrow$ is selected in a random way depending on the value of the flight length

($flight_{len}$). This value ensures that the selected crow is between the first position and the $flight_{len}$ position of the flock, also taking into account that it is not the current one. In this way, one of the best $flight_{len}$ crows is randomly selected. Then, the random crow selected, $RandomCrow$, and the corresponding crow, $Flock[crow]$, are compared in terms of dominance in line 8: if $RandomCrow$ dominates $Flock[crow]$, $RandomCrow$ has a better solution (food), and $Flock[crow]$ will follow it. Therefore, in this case, the mutation operator is carried out over $RandomCrow$ in line 9; otherwise, the mutation operator is performed with the current $Flock[crow]$ in line 11. The mutation operation is explained in detail in Section 4.2. After that, in line 13, the values of the objective functions of the resulting mutated crow ($MutatedCrow$) are calculated. In the last step of the loop, in line 14, the mutated crow is stored in its corresponding position in the flock. In this way, when all the crows have been processed, the size of the flock is duplicated ($2 \cdot flock_{size}$).

In line 16, the rank and crowding operators are performed again over the entire flock, i.e., over the $2 \cdot flock_{size}$ crows. These multi-objective operators sort the flock for the next iteration, assigning the best solutions in the first half of the flock. This first half will be the flock for the next iteration, restoring its original size ($flock_{size}$). For this reason, in line 17, at the end of each iteration, the second half of the flock is stored in the set of non-dominated solutions NDS. A repair operation is carried out in this step with the purpose of ensuring that the stored solutions meet the length constraint defined in Eq. (12). This repair operator is described in Section 4.3.

Finally, in lines 19 and 20, after the end of the main loop, the remaining first half of the flock is stored in the set of non-dominated solutions, also performing the repair operation, and the solutions stored in the final NDS set are filtered to remove the dominated and the equal solutions contained.

4.2. The mutation operator

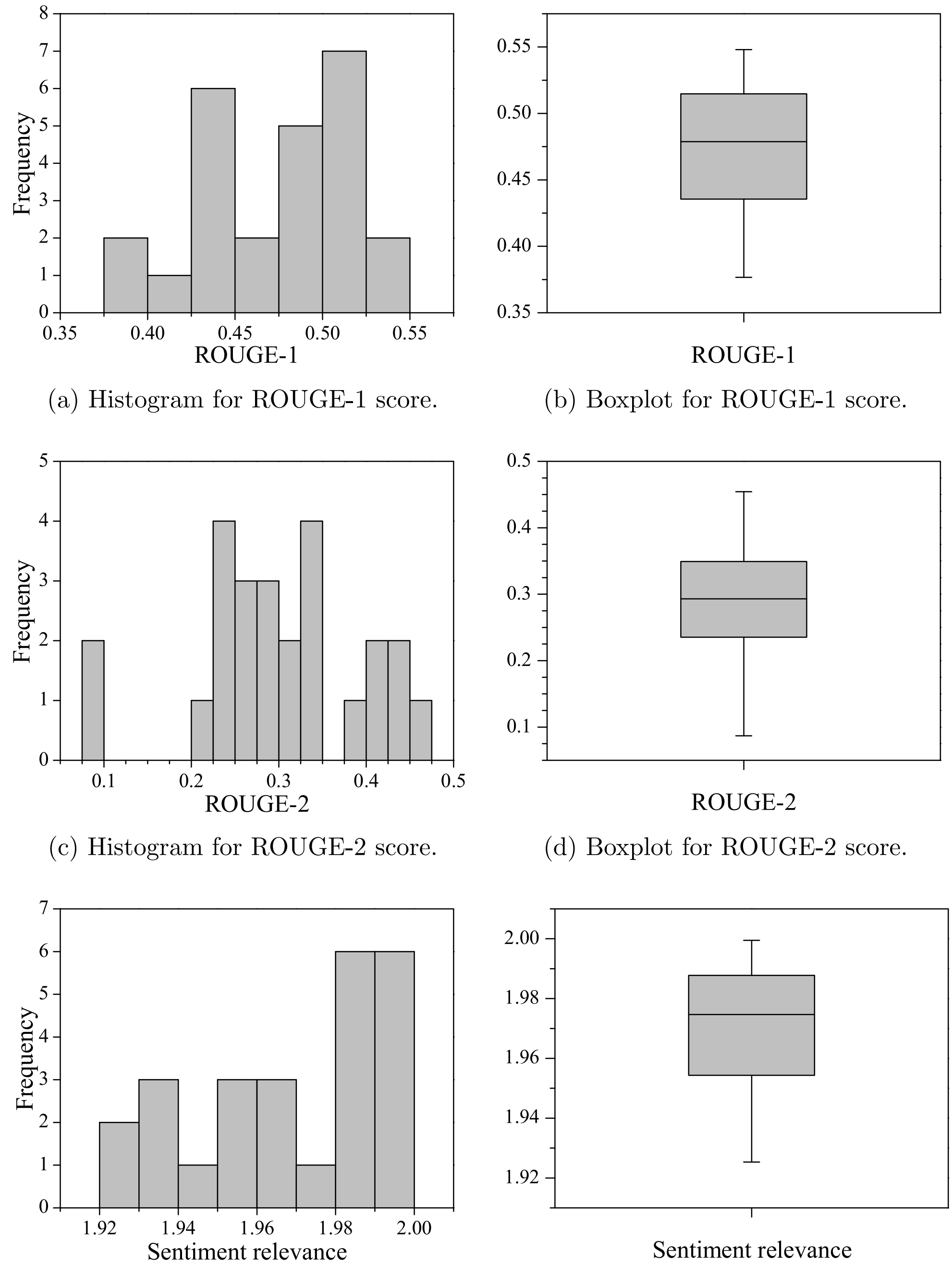
The mutation operator performed in QSO-MOCSA has been specifically designed for this problem. The operator considers the inclusion, removal, or exchange of a unique sentence from the summary. This means that the mutation probability (mut_{prob}) is equal to $1/n$, being n the number of sentences of the document collection. Besides, only one of these three operations will be randomly selected to be applied. The goal of this operator is to improve the quality of the summary both in terms of similarity and sentiment score according to the query. The operation is as follows:

1. Add a sentence. This operation incorporates a random sentence s_i from the document collection that was not contained in the summary S. The added sentence should improve the quality of the mutated summary S'. That is, the cosine similarity between the sentence and the query should be greater than the average cosine similarity between all sentences and the query, and the difference between the sentiment score of the mutated summary and the one of the query should be smaller than the difference between the sentiment score of the initial summary and the one of the query. These conditions are formulated as:
$$cosim(s_i, Q) > \frac{1}{n}\sum_{j=1}^{n} cosim(s_j, Q) \ \text{ and } \quad (14)$$
$$|senti(Q) - senti(S')| < |senti(Q) - senti(S)|.$$
If these two conditions are accomplished, then the sentence is added to the summary. Otherwise, these conditions are checked for every sentence $s_i \notin S$, chosen in a random way, and the first sentence that accomplishes these two conditions will be added. If no sentence fulfills these conditions, the added sentence will be the one with the highest mutation score, $score^{mut}_{s_i}$, which is calculated as follows:
$$score^{mut}_{s_i} = \frac{cosim(s_i, Q) - \frac{1}{n}\sum_{j=1}^{n} cosim(s_j, Q)}{\frac{1}{n}\sum_{j=1}^{n} cosim(s_j, Q)} - \left|\frac{senti(Q) - senti(S')}{senti(Q)}\right|. \quad (15)$$
In this way, the mutation operator ensures that a sentence will be added to the summary.
2. Remove a sentence. In this case, this operation discards a random sentence s_i from the summary S. The removing of this sentence should not degrade the quality of the mutated summary S', i.e., the cosine similarity between the discarded sentence and the query should be smaller than the average cosine similarity between all sentences and the query, and the difference between the sentiment score of the mutated summary and the one of the query should be smaller than the difference between the sentiment score of the initial summary and the one of the query. These two requirements are computed as:
$$cosim(s_i, Q) < \frac{1}{n}\sum_{j=1}^{n} cosim(s_j, Q) \ \text{ and } \quad (16)$$
$$|senti(Q) - senti(S')| < |senti(Q) - senti(S)|.$$
If the two requirements are met, then the sentence is removed from the summary. Otherwise, the requirements are verified for each sentence $s_i \in S$ (also randomly chosen), removing the first sentence that meets these two requirements. Finally, if none of the sentences of the summary fulfills the requirements, the sentence with the lowest mutation score calculated as is indicated in Eq. (15) will be removed. Thereby, the mutation operator guarantees that a sentence is removed from the summary.
3. Exchange a sentence with another one. This third choice replaces a sentence contained in the summary $s_i \in S$ by another sentence from the document collection $s_j \notin S$ that is not contained in the summary. The operation carried out here consists of, firstly, removing a sentence from the summary, and, secondly, adding a different one to the summary. That is, the points 2 and 1 are performed in this order.

4.3. The repair operator

The repair operator carried out in QSO-MOCSA also has been designed and implemented specifically for this problem. This operator is in charge of repairing the summaries that do not fulfill the length constraint established in Eq. (12) with the aim of improving its quality both in terms of similarity and sentiment score according to the query.

The summary length is verified as follows. If the summary length is less than the length constraint $L - \varepsilon$, then the summary is discarded since this case occurs very infrequently. Otherwise, if the length is greater than the length constraint $L + \varepsilon$, then the summary is repaired. The reparation of a summary that violates the length constraint, S^*, is carried out by removing the needed sentences until the length constraint is accomplished. These sentences are selected according to a repair score that takes into



(a) Histogram for ROUGE-1 score. (b) Boxplot for ROUGE-1 score.

(c) Histogram for ROUGE-2 score. (d) Boxplot for ROUGE-2 score.

(e) Histogram for SR score. (f) Boxplot for SR score.

Fig. 1. Histograms and boxplots for ROUGE-1, ROUGE-2, and sentiment relevance (SR) scores.

account both cosine similarity and sentiment score according to the query, which is calculated as follows:

$$score^{rep}_{s_i} = \frac{\left(cosim(O^{S^*}, Q) - cosim(O^{S^*-s_i}, Q)\right)}{cosim(O^{S^*}, Q)} + \left|\frac{senti(Q) - senti(S^* - s_i)}{senti(Q)}\right|, \tag{17}$$

being O^{S^*} the center of the summary S (including the sentence s_i) and $O^{S^*-s_i}$ the center of the summary (without including the sentence s_i). The center of a summary is represented as a vector $O^{S^*} = (o_1, o_2, \dots, o_m)$, and its components are calculated as indicated in Eq. (18):

$$o_k = \frac{1}{n^{S^*}} \sum_{i=1}^{n} w_{ik} \cdot x_i, \quad k = 1, 2, \dots, m, \tag{18}$$

where n^{S^*} is the number of sentences in the summary S^*.

Once the repair scores are calculated, the sentence with the lowest one is removed from the summary, and the repair operation is repeated while the length constraint is not satisfied.

4.4. Method for reducing the Pareto front to a single solution

The result obtained by QSO-MOCSA is a set of non-dominated solutions that are represented in a Pareto front. It is necessary to

Table 1
Features of the Opinion Summarization Track of TAC2008 datasets.

Feature	Value
Number of topics	25
Average number of documents	≈24
Average number of sentences	≈214
Average number of words	≈5029
Average number of different terms	≈994
Summary length constraint (words)	250

Table 2
Summary statistics considering the 25 topics for ROUGE-1, ROUGE-2, and sentiment relevance (SR) scores.

QSO-MOCSA	ROUGE-1	ROUGE-2	SR
Mean	0.4728	0.2987	1.9686
Median	0.4787	0.2930	1.9747
Standard deviation	0.0460	0.0958	0.0245
Q_1	0.4341	0.2340	1.9514
Q_3	0.5148	0.3670	1.9891
Minimum	0.3767	0.0869	1.9253
Maximum	0.5481	0.4542	1.9995

apply an automatic method for selecting one solution from the set. Consequently, in this work, a study of several methods for reducing the Pareto front to a single solution has been performed.

A total of eleven methods have been implemented and evaluated. They are based on the largest hypervolume, the consensus solution, the shortest distance to the ideal point (considering four different distances: Euclidean, Manhattan, Chebyshev, and Mahalanobis), and the shortest distance to all points (considering the same four previous distances in addition to the Levenshtein distance). The detailed mathematical descriptions of all these methods can be found in [11].

5. Experimental results

In this section, the algorithm proposed in Section 4 is used to solve the addressed problem, carrying out different experiments to evaluate its performance. The used datasets, the evaluation metrics, the experimental settings, the results, the comparison with other methods, and two examples of generated summaries are presented next.

5.1. Datasets

Datasets from TAC (Text Analysis Conference) have been used for experimentation. TAC is a series of annual conferences focused on applications of Natural Language Processing, providing also large data collections for testing. The TAC2008 datasets [13] have been used, and more specifically, the Opinion Summarization Track [49] has been considered. Table 1 shows several features of this track.

5.2. Evaluation metrics

In order to measure the quality of the generated summaries, the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics have been considered [14]. These metrics are the most widely used in the automatic text summarization field to evaluate the performance of automatic generated summaries by comparing them with summaries generated by expert humans. Moreover, they are considered by TAC as the official evaluation metrics.

ROUGE metrics measure the similarity between two summaries (a computer-generated one and a human-generated one) by means of adding up the amount of overlapping units. In this work, ROUGE-*N* metric has been used for the evaluation. It consists of computing the *N*-gram recall between the computer-generated summary and a set of reference summaries (generated by human experts). Particularly, ROUGE-1 and ROUGE-2 scores have been considered. ROUGE-1 calculates the number of overlapping unigrams, whereas ROUGE-2 counts the number of overlapping bigrams. ROUGE-*N* metric is calculated as:

$$ROUGE\text{-}N = \frac{\sum_{S \in ReferS} \sum_{N\text{-}gram \in S} count_{co}(N\text{-}gram)}{\sum_{S \in ReferS} \sum_{N\text{-}gram \in S} count(N\text{-}gram)}, \quad (19)$$

being $ReferS$ the set of reference summaries, $count_{co}(N\text{-}gram)$ the amount of N-grams co-occurring between the candidate summary and $ReferS$, and $count(N\text{-}gram)$ the quantity of N-grams in S.

As for the evaluation of the sentiment score of the generated summaries, the Pearson correlation coefficient r has been considered. This coefficient has been used to measure the linear correlation between the sentiment scores of the generated summaries and the sentiment scores of the queries of the topics from Opinion Summarization Track of TAC2008 datasets.

The Pearson correlation coefficient, r, will have a value of $+1$ when the linear correlation is positive and perfect, a value of -1 when the linear correlation is negative and perfect, and a value of 0 in the case that there is absence of linear correlation. Intermediate values indicate positive or negative relationships as well as strong association if the value approaches 1 or -1, and weak association if the value approaches 0.

5.3. Experimental settings

The parameters of QSO-MOCSA are listed below: population/flock size, $flock_{size}$, flight length, $flight_{len}$, and number of iterations or generations, $iter_{max}$. After experimenting with different parameter settings, the values established for these parameters are the following ones: $flock_{size} = 64$, $flight_{len} = 6$, and $iter_{max} = 500$.

As for the chosen method for selecting a single solution from the Pareto front, a comparative study of the eleven methods presented in Section 4.4 has been performed. The obtained results report that the method of the consensus solution has achieved the best average ROUGE scores, in addition to provide the best values in most of the topics.

The number of repetitions (independent runs per every experiment) has been 31 in order to guarantee the robustness and statistical reliability of the results. The experiments have been carried out in a compute node with 4 processors AMD Opteron Abu Dhabi 6376 and 96-GB RAM. QSO-MOCSA has been implemented in C/C++ language and developed with the Eclipse platform on Ubuntu 20.04 LTS.

5.4. Results

The results obtained in the experimentation are presented in this subsection. Firstly, both the ROUGE scores and the sentiment relevance (SR) scores are reported in Table 2. This table shows the summary statistics for the 25 topics from the Opinion Summarization Track of TAC2008 datasets.

As can be appreciated in Table 2, the obtained mean ROUGE scores are 0.4728 for ROUGE-1 and 0.2987 for ROUGE-2. They will be used in Section 5.5 in order to make comparisons with other methods. The mean score of the sentiment relevance is 1.9686, which is a very good value since the theoretical maximum (deduced from Eq. (10)) is 2. Now, the obtained results are graphically represented in Fig. 1.

ROUGE-1, ROUGE-2, and sentiment relevance distributions over the 25 topics are represented through the histograms of Figs. 1a, 1c, and 1e. These distributions are also graphically represented through boxplots in Figs. 1b, 1d, and 1f. Each boxplot presents

Table 3
Comparison of mean scores for ROUGE-1 and ROUGE-2 obtained by QSO-MOCSA and by the other methods. The best results are shown in bold.

Methods	ROUGE-1	ROUGE-2
QSO-MOCSA	**0.4728**	**0.2987**
QMOS [26]	0.4123	0.0985
CCNU [20]	0.2011	0.0554
IIITSum [21]	0.1511	0.0323
ITALICA [22]	0.3796	0.0745
NUS [23]	0.3484	0.0546
PolyU [24]	0.2932	0.0760
IITSummarizers [25]	0.3279	0.0439

Table 4
Percentage improvements reached by QSO-MOCSA with respect to the other methods.

Methods	ROUGE-1	ROUGE-2
QMOS [26]	14.7%	203.3%
CCNU [20]	135.1%	439.2%
IIITSum [21]	212.9%	824.9%
ITALICA [22]	24.6%	301.0%
NUS [23]	35.7%	447.1%
PolyU [24]	61.3%	293.1%
IITSummarizers [25]	44.2%	580.5%
Average	75.5%	441.3%

a box and two whiskers, being the median represented as the central segment, the first quartile Q_1 as the lower segment, and the third quartile Q_3 as the upper segment. In addition, the minimum value is represented as the lower whisker limit, and the maximum value as the upper one since no outliers have been obtained.

5.5. Comparison with other methods

The mean values provided by QSO-MOCSA are compared with the results reported by other methods. They have been previously reviewed in Section 2. Table 3 shows the average results for ROUGE-1 and ROUGE-2 scores obtained in the 25 topics from the Opinion Summarization Track of TAC2008 datasets for QSO-MOCSA and the different proposals existing in the scientific literature. The methods from other authors include a very recently published proposal (QMOS [26]) and the algorithms typically proposed in the field of query-focused sentiment-oriented summarization.

The results reported in Table 3 show that QSO-MOCSA achieves the best average values both in ROUGE-1 and ROUGE-2 scores. Table 4 presents the percentage improvements reached by QSO-MOCSA with respect to the other methods.

As can be appreciated in Table 4, the percentage improvements range from 14.7% to 212.9% in ROUGE-1 score and from 203.3% to 824.9% in ROUGE-2 score. In addition, the average percentage improvements are 75.5% and 441.3% for ROUGE-1 and ROUGE-2 scores, respectively. Note that the produced percentage improvements are important in both ROUGE scores, being those obtained in ROUGE-2 score much higher. This indicates that the proposed QSO-MOCSA performs even better when evaluating the bigrams overlapping with the summaries generated by human experts.

Other methods do not report sentiment relevance scores, so no comparison is possible. Nevertheless, by means of the Pearson correlation coefficient r, it is possible to compare the values provided by QSO-MOCSA with the ideal case, which occurs when $r = +1$. A value of $r = +0.841$ has been obtained, that is, there is a strong linear positive correlation between the sentiment scores of the summaries and the sentiment scores of the queries.

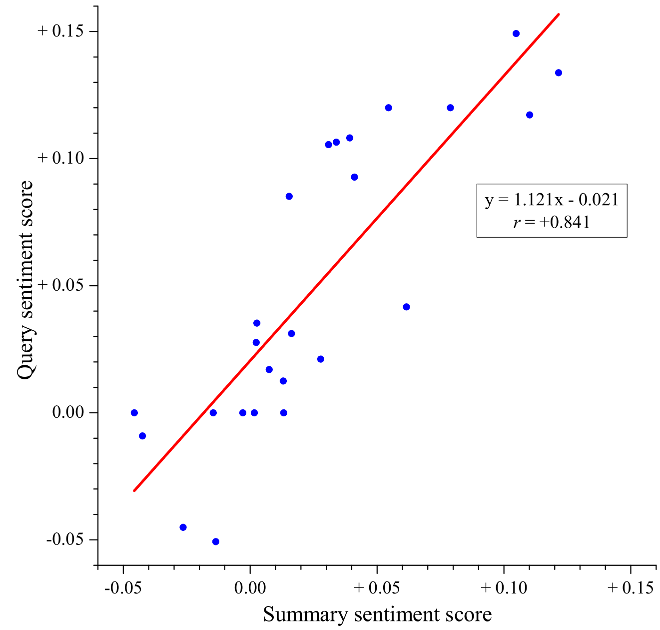


Fig. 2. Scatter plot and regression line for the summary sentiment scores and the query sentiment scores.

Table 5
Features of topics 1005 and 1049 of the Opinion Summarization Track from TAC2008 datasets.

Feature	Topic 1005	Topic 1049
Title	Windows Vista	YouTube
Query	“What features do people like about Vista?”	“What reason do users who prefer YouTube to any other video-sharing website cite as reasons?”
No. of documents	35	29
No. of sentences	268	247
No. of words	5962	6667
No. of terms	1097	1304

Fig. 2 represents the scatter plot and the regression line for the summary sentiment scores and the query sentiment scores.

Fig. 2 shows how the relationship between the summary sentiment scores and the query sentiment scores is strong and linear, with a regression line reporting that both scores simultaneously increase (positive slope).

5.6. Examples of summaries generated by QSO-MOCSA

In this subsection, some examples of the summaries generated by QSO-MOCSA are presented. Topics 1005 and 1049 of the Opinion Summarization Track from TAC2008 datasets are considered. Table 5 shows some features of these topics.

Firstly, the summary generated for topic 1005 by QSO-MOCSA and the reference summary from NIST human experts are presented in Fig. 3.

And secondly, the summary generated for topic 1049 and the corresponding reference summary from human experts are shown in Fig. 4.

Several conclusions can be extracted from both examples. In the first place, the sentences of the generated summaries have terms in common with the given query. Both summaries contain the terms presented in the queries: Vista and YouTube, respectively. In fact, some sentences are also present in the reference summaries from human experts. Thus, the generated summaries have a high degree of query relevance. Another aspect is that the

Cheap deals on Vista PCs will almost certainly deliver boxes that run on XP lines and be Vista in name only.

Windows Vista will automatically scale to the hardware it finds.

Windows Vista Business will help you lower your PC management costs, improve your security, enhance your productivity, and help you stay better connected.

He highlighted how Windows Vista will bring clarity to customers' digital world and help them easily accomplish everyday tasks, instantly find what they want, enjoy the latest in entertainment, keep their personal information safe, and stay connected at home or on the go.

The Media Center experience in Windows Vista can also be moved throughout the home via the Media Center Extender functionality of the Xbox 360 video game and entertainment system, as it can be today with Windows XP Media Center Edition 2005.

Gates demonstrated how Windows Vista will bring clarity to the world of personal computing so customers can more safely and easily accomplish everyday tasks, instantly find what they want, enjoy the latest in entertainment, and stay connected at home or on the go.

Windows Vista has many new features such as an updated graphical user interface called Aero, improved searching technology, a number of new security features, and completely new networking, audio, print, and display engines.

Tablet PC features in Vista: A number of Tablet PC features will be incorporated into.

One of the quantum leaps Windows Vista will make is the move from raster graphics to high quality vector graphics.

The visual information on computer screens running full versions of Vista will be assembled using lines and curves not grids and dots. These vector elements scale without breaking up into jagged edges and boxy mosaics, so will produce much better quality pictures and text. Similarly, improved handling of text will allow greater flexibility in the presentation.

Bill shows how the modern office will look like (big screen with three desktops), touchscreens, the connection with your mobile, real time news, etc.

Gates demonstrated how Windows Vista will bring clarity to the world of personal computing so customers can more safely and easily accomplish everyday tasks, instantly find what they want, enjoy the latest in entertainment, and stay connected at home or on the go.

Windows Vista Home Basic is designed to deliver improved reliability, security, and usability to home PC users who just want to do the basics with their PCs.

Windows Vista has many new features such as an updated graphical user interface called Aero, improved searching technology, a number of new security features, and completely new networking, audio, print, and display engines.

The Win Vista UI has search integrated through it, the Virtual folder functionality is seamless and it has many new shortcuts like the new address bar that makes navigating the system easier and getting to what you want to do faster.

Fig. 3. Summary generated by QSO-MOCSA and reference summary from NIST human experts for topic 1005 (Windows Vista).

Browsing through YouTube, even the highest ranked content seems to lack a certain level of resolution quality.

YouTube seems like it has more users but is less about friends and tagging and creating your own content.

Not at once, but the review monkeys over at YouTube are very alert and step in very often to remove copyrighted or obscene material.

With YouTube, I can browse on the tags I gave the clips, I can embed the clips into my blog, I can subscribe to RSS feeds from users, or have RSS feeds being generated on a specific tag or a combination of tags.

I predict is just a matter of time before we see a headline about a school district taking action against a student for a contentious video posted to YouTube. Interestingly, MySpace (the wildly popular digital native social networking site) tried to actually block member access (at least direct linking) to YouTube for a while, but apparently those issues have been resolved.

The ability to upload a web video is not really anything new, but as a web 2.0 social networking website enabling tagging and dynamically showing most viewed videos, it really does take global publication of digital, amateur video to another level.

YouTube is also a sharing network.

YouTube's blog states: "We know how popular that video is but YouTube respects the rights of copyright holders".

Browsing through YouTube, even the highest ranked content seems to lack a certain level of resolution quality.

With YouTube, I can browse on the tags I gave the clips, I can embed the clips into my blog, I can subscribe to RSS feeds from users, or have RSS feeds being generated on a specific tag or a combination of tags.

You can also use YouTube to share private videos with friends or family, making it very easy to upload clips from family reunions, dinner parties or birthdays and share them with other loved ones online.

YouTube, founded by Internet commerce pioneers from PayPal, has developed a new service that allows people to easily upload, tag, and share personal video clips through YouTube and across the Internet on other sites, blogs and through e mail.

YouTube is very web 2.0, has thousands and thousands of users and allows those users to add personal value to each clip.

I am sure YouTube has already found itself on the blacklist of most school Internet filters.

YouTube also provides easy to use copy paste code for emailing and posting on a website.

All the content on YouTube is uploaded by users but most of it isn't owned by the user. There are huge rights questions in that regard.

YouTube is a service through which individual fans can share stuff they're nuts about with others.

Fig. 4. Summary generated by QSO-MOCSA and reference summary from NIST human experts for topic 1049 (YouTube).

sentences included in the generated summaries are not redundant with each other, despite containing the same terms present in the queries. Finally, as for the sentiment scores, the sentiment words contained in the generated summaries are similar to the sentiment words presented in the queries.

6. Conclusion

The query-focused extractive multi-document text summarization task consists of generating a summary automatically according to a determined user information, that is given as a query. Additionally, the sentiment analysis and opinion mining task focuses on the analysis of the polarities of the sentences from a document collection, also considering their sentiment scores. Therefore, joining both issues in a single one, it is possible to produce a summary that includes the most relevant sentences for the user's query, also having a similar sentiment orientation. In this regard, query-focused sentiment-oriented extractive multi-document text summarization involves the simultaneous optimization of the criteria of the query relevance, the redundancy reduction, and the sentiment relevance.

A multi-objective adaptation of the crow search algorithm has been designed, implemented, and tested to solve the query-focused sentiment-oriented extractive multi-document text summarization problem. This new approach is named QSO-MOCSA (Query-focused Sentiment-Oriented Multi-Objective Crow Search Algorithm). In this paper, QSO-MOCSA has been explained in detail, including its new problem-aware operators.

The obtained results have shown that QSO-MOCSA outperforms the methods existing in the scientific literature. Specifically, the average percentage improvements have been 75.5% for ROUGE-1 and 441.3% for ROUGE-2. The reported value of the Pearson correlation coefficient has been $r = +0.841$, meaning a strong linear positive correlation between the sentiment scores of the generated summaries and the sentiment scores of the queries.

As a future research, QSO-MOCSA could be adapted for solving update summarization problems. Update summarization is an extension of traditional static multi-document text summarization, in which the document collection changes over time, and users just want to know the new information about a particular topic. In this way, the proposed model would generate an update summary that would also take into account the query given by the user and its sentiment orientation. In this case, it is very possible that more than 3 objectives are needed, therefore, requiring the use of many-objective optimization techniques. Another possible research line for the future is the development and application of alternative multi-objective approaches for comparative purposes, trying to improve even more the obtained results.

CRediT authorship contribution statement

Jesus M. Sanchez-Gomez: Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Miguel A. Vega-Rodríguez:** Conceptualization, Methodology, Formal analysis, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Carlos J. Pérez:** Conceptualization, Methodology, Formal analysis, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

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

Acknowledgments

Work supported by *Ministerio de Ciencia, Innovación y Universidades* - Spain and *Agencia Estatal de Investigación* - Spain (Projects PID2019-107299GB-I00/AEI/10.13039/501100011033 and MTM2017-86875-C3-2-R), *Junta de Extremadura* - Spain (Projects GR18090 and GR18108), and European Union (European Regional Development Fund). Jesus M. Sanchez-Gomez is supported by *Junta de Extremadura*, Spain and European Union (European Social Fund) under the doctoral fellowship PD18057.

References

[1] H. Hassani, C. Beneki, S. Unger, M.T. Mazinani, M.R. Yeganegi, Text mining in big data analytics, Big Data Cogn. Comput. 4 (1) (2020) 1.
[2] R.K. Bakshi, N. Kaur, R. Kaur, G. Kaur, Opinion mining and sentiment analysis, in: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, 2016, pp. 452–455.
[3] R.M. Alguliyev, R.M. Aliguliyev, N.R. Isazade, A. Abdi, N. Idris, COSUM: Text summarization based on clustering and optimization, Expert Syst. 36 (1) (2019) e12340: 1–17.
[4] M. Afsharizadeh, H. Ebrahimpour-Komleh, A. Bagheri, Query-oriented text summarization using sentence extraction technique, in: 2018 4th International Conference on Web Research (ICWR), IEEE, 2018, pp. 128–132, http://dx.doi.org/10.1109/ICWR.2018.8387248.
[5] S. Song, H. Huang, T. Ruan, Abstractive text summarization using LSTM-CNN based deep learning, Multimedia Tools Appl. 78 (1) (2019) 857–875.
[6] J.V. Tohalino, D.R. Amancio, Extractive multi-document summarization using multilayer networks, Physica A 503 (2018) 526–539.
[7] J.M. Sanchez-Gomez, M.A. Vega-Rodríguez, C.J. Pérez, The impact of term-weighting schemes and similarity measures on extractive multi-document text summarization, Expert Syst. Appl. 169 (2021) 114510.
[8] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E.D. Trippe, J.B. Gutierrez, K. Kochut, Text summarization techniques: A brief survey, Int. J. Adv. Comput. Sci. Appl. 8 (10) (2017) 397–405.
[9] R. Rautray, R.C. Balabantaray, An evolutionary framework for multi document summarization using cuckoo search approach: MDSCSA, Appl. Comput. Inform. 14 (2) (2018) 134–144.
[10] J.M. Sanchez-Gomez, M.A. Vega-Rodríguez, C.J. Pérez, Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach, Knowl.-Based Syst. 159 (2018) 1–8.
[11] J.M. Sanchez-Gomez, M.A. Vega-Rodríguez, C.J. Pérez, Comparison of automatic methods for reducing the Pareto front to a single solution applied to multi-document text summarization, Knowl.-Based Syst. 174 (2019) 123–136.
[12] N. Akhtar, N. Zubair, A. Kumar, T. Ahmad, Aspect based sentiment oriented summarization of hotel reviews, Procedia Comput. Sci. 115 (2017) 563–571.
[13] H.T. Dang, Overview of the TAC 2008 opinion question answering and summarization tasks, in: Proceedings of the Text Analysis Conference (TAC 2008), National Institute of Standards and Technology, 2008, pp. 1–45.
[14] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, 8, Association for Computational Linguistics, 2004, pp. 74–81.
[15] W.S. El-Kassas, C.R. Salama, A.A. Rafea, H.K. Mohamed, Automatic text summarization: A comprehensive survey, Expert Syst. Appl. 165 (2021) 113679.
[16] C. Zucco, B. Calabrese, G. Agapito, P.H. Guzzi, M. Cannataro, Sentiment analysis for mining texts and social networks data: Methods and tools, Wiley Interdiscipl. Rev. - Data Min. Knowl. Discov. 10 (1) (2020) e1333.
[17] D. Yazdani, R. Cheng, D. Yazdani, J. Branke, Y. Jin, X. Yao, A survey of evolutionary continuous dynamic optimization over two decades – part b, IEEE Trans. Evol. Comput. (2021) 1–21.
[18] I. Fister, M. Mernik, B. Filipič, A hybrid self-adaptive evolutionary algorithm for marker optimization in the clothing industry, Appl. Soft Comput. 10 (2) (2010) 409–422.
[19] R.-E. Precup, R.-C. David, E.M. Petriu, A.-I. Szedlak-Stinean, C.-A. Bojan-Dragos, Grey wolf optimizer-based approach to the tuning of PI-fuzzy controllers with a reduced process parametric sensitivity, IFAC-PapersOnLine 49 (5) (2016) 55–60.
[20] T. He, J. Chen, Z. Gui, F. Li, CCNU at TAC 2008: Proceeding on using semantic method for automated summarization yield, in: Proceedings of the Text Analysis Conference (TAC 2008), National Institute of Standards and Technology, 2008, pp. 1–8.

[21] V. Varma, P. Pingali, R. Katragadda, S. Krishna, S. Ganesh, K. Sarvabhotla, H. Garapati, H. Gopisetty, V.B. Reddy, K. Reddy, P. Bysani, R. Bharadwaj, IIIT hyderabad at TAC 2008, in: Proceedings of the Text Analysis Conference (TAC 2008), National Institute of Standards and Technology, 2008, pp. 1–12.
[22] F. Cruz Mata, J.A. Troyano Jiménez, F.J. Ortega Rodríguez, F. Enríquez de Salamanca Ros, The italica system at TAC 2008 opinion summarization task, in: Proceedings of the Text Analysis Conference (TAC 2008), National Institute of Standards and Technology, 2008, pp. 1–4.
[23] Z. Lin, H.H. Hoang, L. Qiu, S. Ye, M.-Y. Kan, NUS at TAC 2008: Augumenting timestamped graphs with event information and selectively expanding opinion contexts, in: Proceedings of the Text Analysis Conference (TAC 2008), National Institute of Standards and Technology, 2008, pp. 1–9.
[24] W. Li, O. You, Y. Hu, F. Wei, PolyU At TAC 2008, in: Proceedings of the Text Analysis Conference (TAC 2008), National Institute of Standards and Technology, 2008, pp. 1–4.
[25] S. Kumar, D. Chatterjee, IIT kharagpur at TAC 2008: Statistical model for opinion summarization, in: Proceedings of the Text Analysis Conference (TAC 2008), National Institute of Standards and Technology, 2008, pp. 1–8.
[26] A. Abdi, S.M. Shamsuddin, R.M. Aliguliyev, QMOS: Query-based multi-documents opinion-oriented summarization, Inf. Process. Manage. 54 (2) (2018) 318–338.
[27] O. Kolchyna, T.T.P. Souza, P. Treleaven, T. Aste, Twitter sentiment analysis: Lexicon method, machine learning method and their combination, in: G. Mitra, X. Yu (Eds.), Handbook of Sentiment Analysis in Finance, Albury Books, 2016, pp. 1–32.
[28] ROUGE, Summary evaluation package, 2017, URL: http://www.berouge.com/ (Last accessed: 14-July-2017).
[29] D. Yogish, T. Manjunath, R.S. Hegadi, Review on natural language processing trends and techniques using NLTK, in: International Conference on Recent Trends in Image Processing and Pattern Recognition, Springer, 2018, pp. 589–606, http://dx.doi.org/10.1007/978-981-13-9187-3_53.
[30] K. Samhith, S.A. Tilak, G. Panda, Word sense disambiguation using WordNet lexical categories, in: 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), IEEE, 2016, pp. 1664–1666, http://dx.doi.org/10.1109/SCOPES.2016.7955725.
[31] M. Porter, The porter stemming algorithm, 2020, URL: http://www.tartarus.org/martin/PorterStemmer/ (Last accessed: 26-July-2021).
[32] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inf. Process. Manage. 24 (5) (1988) 513–523.
[33] J. Kumar, J.K. Rout, A. Katiyar, S.K. Jena, Sentiment analysis using weight model based on SentiWordNet 3.0, in: Recent Findings in Intelligent Computing Techniques, Springer, 2018, pp. 131–139, http://dx.doi.org/10.1007/978-981-10-8633-5_14.
[34] Princeton University, About WordNet, 2012, URL: https://wordnet.princeton.edu/ (Last accessed: 26-July-2021).
[35] S. Verma, P. Bhattacharyya, Incorporating semantic knowledge for sentiment analysis, in: Proceedings of 6th International Conference on Natural Language Processing, 1, Macmillan Publishers, 2009, pp. 1–6.
[36] T. Chen, R. Xu, Y. He, X. Wang, Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN, Expert Syst. Appl. 72 (2017) 221–230.
[37] R. Narayanan, B. Liu, A. Choudhary, Sentiment analysis of conditional sentences, in: EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1, Association for Computational Linguistics, 2009, pp. 180–189, http://dx.doi.org/10.3115/1699510.1699534.
[38] R. Xia, F. Xu, J. Yu, Y. Qi, E. Cambria, Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis, Inf. Process. Manage. 52 (1) (2016) 36–45.
[39] U. Farooq, H. Mansoor, A. Nongaillard, Y. Ouzrout, M.A. Qadir, Negation handling in sentiment analysis at sentence level, J. Comput. 12 (5) (2017) 470–478.
[40] B. Liu, Sentiment analysis and opinion mining, Synth. Lect. Hum. Lang. Technol. 5 (1) (2012) 1–167.
[41] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[42] D. Oliva, S. Hinojosa, E. Cuevas, G. Pajares, O. Avalos, J. Gálvez, Cross entropy based thresholding for magnetic resonance brain images using Crow Search Algorithm, Expert Syst. Appl. 79 (2017) 164–180.
[43] D. Gupta, S. Sundaram, A. Khanna, A.E. Hassanien, V.H.C. De Albuquerque, Improved diagnosis of parkinson's disease using optimized crow search algorithm, Comput. Electr. Eng. 68 (2018) 412–424.
[44] F. Mohammadi, H. Abdi, A modified crow search algorithm (MCSA) for solving economic load dispatch problem, Appl. Soft Comput. 71 (2018) 51–65.
[45] G.I. Sayed, A.E. Hassanien, A.T. Azar, Feature selection via a novel chaotic crow search algorithm, Neural Comput. Appl. 31 (1) (2019) 171–188.
[46] S. Makhdoomi, A. Askarzadeh, Optimizing operation of a photovoltaic/diesel generator hybrid energy system with pumped hydro storage by a modified crow search algorithm, J. Energy Storage 27 (2020) 101040.
[47] Z.O. Ahmed, A.A. Khorsheed, Biometric key generation using crow algorithm, Indones. J. Electr. Eng. Comput. Sci. 21 (1) (2021) 208–214.
[48] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[49] TAC 2008 Opinion Summarization Track, Text analysis conference, 2017, https://tac.nist.gov/2008/summarization/op.summ.08.guidelines.html (Last accessed: 26-July-2021).