ISSN 2007-9737

Selection of Content Measures for Evaluation of Text Summaries Using Genetic Algorithms

Jonathan Rojas-Simón*, Yulia Ledeneva, René Arnulfo García-Hernández

Autonomous University of the State of Mexico, Toluca, Mexico

{jrojass@uaemex.mx, ynledeneva, reagarciah}@uaemex.mx

Abstract. Automatic Text Summarization (ATS) has played an essential role in condensing textual information from digital documents. Since 2001, the development of ATS has been significant, aiming to emulate the creation of human-like summaries. Thus, most of the methods and approaches are usually evaluated through ROUGE; however, it does not evaluate if human references are not available. Due to this, the Evaluation of Text Summaries (ETS) without human references has been proposed. In this sense, ROUGE-C, LSA, and SIMetrix have been presented as methods that compare the content between summaries and source documents. Although previous studies have demonstrated that combining these methods has improved automatic evaluation, it is still far from manual assessment. We assume this situation is due to the presence of different complexity levels in evaluation measures and source documents. Therefore, the performance of automatic evaluation varies according to the complexity level of each evaluation measure. In this paper, we propose a selection of content evaluation measures through a Genetic Algorithm (GA) to determine the most suitable evaluations for each summary. Calculating complexity levels in source documents and content measures may help to select the best measures to evaluate summaries without human references. Experiments in the DUC01 and DUC02 datasets demonstrate that the proposed selection improves the performance of this task.

Keywords Evaluation of text summaries content evaluation measures genetic algorithm text complexity indexes.

1 Introduction

In recent times, text documents have become the most essential resource for the user in daily life. Such documents show useful information in different formats (*e.g.*, books, scientific/news articles, monographs, and social media comments) that satisfy users' requirements. Nevertheless, they grow exponentially on the Internet, causing an information overload. For this reason, the Automatic Text Summarization (ATS) seeks to solve this problem because it is considered the most recognized kind of text condensation [1].

Over the last two decades, several methods have been proposed in the ATS that generate summaries of different characteristics (single- and multi-document; extractive, abstractive, and hybrid; generic and query-based; monolingual, multilingual, and cross-lingual) [2]. However, the Evaluation of Text Summaries (ETS) is a complex task that requires exhaustive studies to determine suitable criteria to assess summaries [3].

According to Jones and Galliers [4], each evaluation method may be *extrinsic* or *intrinsic*. The extrinsic evaluation measures the usefulness of summaries in another task, such as document categorization, relevance assessment, and web search [5, 6]. On the other hand, the intrinsic evaluation focuses on the suitability of the summarization approach in terms of text quality and content analysis [7].

In other words, it analyzes the summary's *content*, *coherence*, and *informativeness* [8]. Most of these methods generally compare the content between a summary to be evaluated (candidate summary) and a set of summaries written by the human expert (human references).

Depending on the degree of human intervention, an evaluation method may be *manual*, *semi-automatic*, or *automatic*. Manual assessment involves human judgments to decide whether a summary shows the essential information from one or more documents.

Nevertheless, this evaluation faces two drawbacks: (i) it is time-consuming, and (ii) several assessors are required to evaluate many summaries [8]. Considering the issues mentioned above, automatic evaluation has been proposed.

For this evaluation, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is the most representative package in the state-of-the-art that includes some measures, such as ROUGE-N, L, W, and S to evaluate the content of summaries [9]. However, these measures are inadequate when they do not have human references. Due to this situation, the ETS without human references has been proposed.

The ETS without human references has attracted attention since traditional methods are impractical and expensive. In this regard, ROUGE-C [10], LSA [7], and SIMetrix [11] have been proposed as methods that compare the candidate summary's content concerning its source document(s). These methods consider source documents as references because they contain enough information, providing helpful knowledge about the topic, but they are still far from manual assessment [12].

To solve this issue, previous studies have proposed the linear optimization of content measures through Genetic Algorithms (GAs) to improve automatic evaluation [3, 13]. The outcome of this research was SECO-SEVA, an evaluation package that combines 31 content measures derived from ROUGE-C, LSA, and SIMetrix. However, the GA's optimization assumes the presence of different levels of complexity in each evaluation measure.

In this paper, we propose a selection of content measures for the ETS without human references, using the GA. We assume that assigning a complexity value to each source document and measure may be a suitable indicator to estimate an adequate evaluation measure for each summary. For this, we have employed 13 complexity indexes known in the state-of-the-art and 31 evaluation measures used in [13].

The paper is organized as follows. In Section “Related Works”, we present a brief description of related works of this research. In Section “State-of-the-Art Evaluation Measures and Text Complexity Indexes”, we describe state-of-the-art evaluation methods and complexity indexes.

The proposed method is presented in Section “Proposed Method”. In Section “Experiments and Results”, we display the results of the proposed selection of measures and a comparison to other state-of-the-art methods. Finally, the conclusions and future works are drawn in Section “Conclusions and Future Works”.

2 Related Works

Over the last few years, several studies have been conducted to improve automatic evaluation. Some perform the linear optimization of measures through the Monte Carlo method [14], linear regression [15], and GA [13]. Other works employ single and ensemble learning classifiers that use evaluation measures as features to predict the score of summaries [16, 17].

Although such works seek the combination of evaluation methods to improve automatic evaluation, it does not always guarantee a better approximation toward human judgments. We assume this because if we include more perspectives to solve a problem, the complexity of the task is increased. This situation is similar to other Natural Language Processing (NLP) tasks.

In [18], García-Calderón et al. proposed a hybrid method based on a selection of Text Line segmentation (TLS) methods, using a complexity index called TLS-ICI (Text Line Segmentation Intrinsic Complexity Index). The TLS-ICI index is shown in Eq. (1), which is the average of four normalized subindexes that measure the amount of information presented in an individual interlinear space.

Let an image I composed of an interlinear space from a handwritten document; therefore, the complexity of such space is calculated using the Horizontal Projection Profile (HPP), the Vertical Projection Profile (VPP), the Histogram of Color of the Bitmap (HCB), and the Histogram of Color of the Ink (HCI):

$$TLS - ICI(I) = \frac{HPP(I) + VPP(I) + HCB(I) + HCI(I)}{4}. \quad (1)$$

Once the complexity of an interlinear space is calculated, it is estimated the overall complexity of a document according to Eq. (2), where D

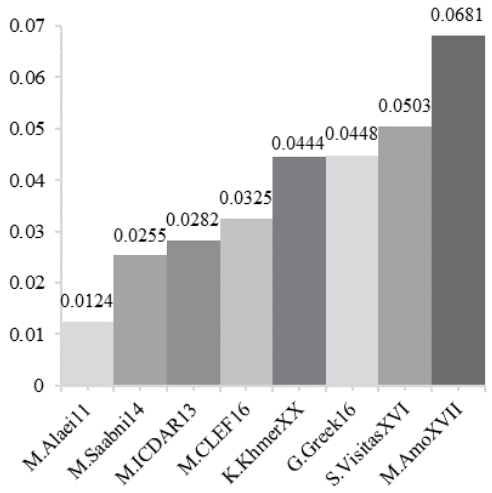

Fig. 1. Ranking of average TLS-ICI for each collection of historical documents [18]

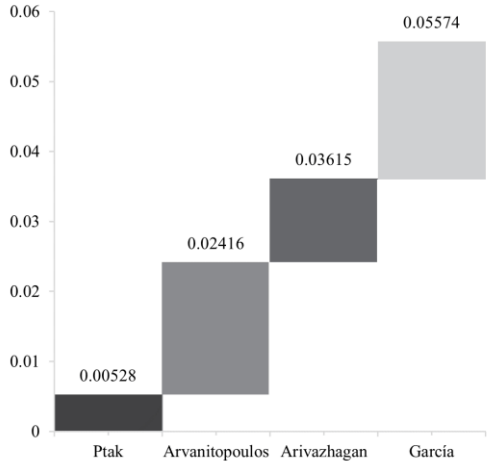

Fig. 2. Complexity values of each TLS method according to 95% of the accuracy of MatchScore

represents the handwritten text document composed of k images of interlinear spaces (I_k^D). Thus, the overall complexity is the average TLS-ICI of their interlinear spaces [18]:

$$TLS-ICI(D)=\frac{1}{k}\sum_{i=1}^{k}TLS-ICI\left(I_i^D\right). \quad (2)$$

Based on TLS-ICI calculates the complexity of handwritten documents, it likewise provides an order of complexity to collections of documents. Such an order establishes that the first collections show lower complexities than the subsequent ones. For the experimentation stage, this order was obtained from eight collections of contemporary and ancient texts written in English, Spanish, Arabic, Chinese, Greek, Khmer, Persian, Bengali, Oriya, Kannada, and Nahuatl [19, 20, 21, 22, 23, 24, 25, 26].

Fig. 1 shows the average TLS-ICI for each collection. As observed, all collections are sorted from lower to higher complexity values. According to the collection, M.AmoXVII shows the highest complexity. Therefore, it would be expected that, in principle, the most sophisticated method performs best the task in this collection. On the other hand, collections of lower complexities (e.g., M.Alaei11 or M.Saabni14) should be analyzed by straightforward TLS methods.

In addition to collections, measuring the complexity of state-of-the-art TLS methods was also necessary. Based on this, the authors selected Ptak [20], Arvanitopoulos [27], García [28], and Arivazhagan [29] methods since they are considered the best ones for the TLS.

Afterward, the complexity of each method was calculated by the maximum complexity threshold with 95% of the accuracy of $MatchScore$.[1] Fig. 2 shows the complexity of TLS methods. According to Fig. 2, the Ptak method shows the lowest complexity because it worked well in documents of the lowest complexity, followed by the Arvanitopoulos and Arivazhagan methods, showing higher complexities.

The García method displays the most increased complexity, which means this method can analyze more complex documents. Based on the values shown in Fig. 2, the complexity values of each TLS method were used to determine the ranges that each method should be selected, creating a Hybrid Method (HM). The HM method is formally described in Eq. (3), where d is the image of the input document and $TLS-ICI(d)$ is the complexity value of d.

[1] $MatchScore$ is an evaluation measure based on counting the number of one-to-one matches between the areas detected by the TLS method and the areas in the ground truth [57].

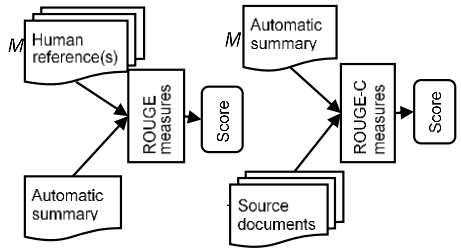

Fig. 1. Differences between ROUGE and ROUGE-C [10]

As observed, the HM method consists of four TLS methods, such that only one is applied according to the complexity of the input document:

$$HM(d) = \begin{cases} Ptak(d), 0 \leq TLS-ICI(d) \leq 0.00528, \\ Arvanitopoulos(d), 0.00528 \leq TLS-ICI(d) < 0.02416, \\ Arivazhagan(d), 0.02416 \leq TLS-ICI(d) < 0.03615, \\ Garcia(d), 0.03615 \leq TLS-ICI(d). \end{cases} \quad (3)$$

For instance, if a document obtains a $TLS-ICI$ value equal 0.02011, the HM method selects the Arvanitopoulos method ($Arvanitopoulos(d)$). On the other hand, if d gets a $TLS-ICI$ value equal or higher than 0.03615, the HM method selects the best available method ($Garcia(d)$).

3 State-of-the-Art Evaluation Measures and Text Complexity Indexes

Content-based measures and text complexity indexes are two groups of measurements that help to estimate specific attributes of texts. While the first group is typically used to evaluate summaries with or without human references [30], the second group employs different formulas that estimate the degree of ease or difficulty a text can be understood according to its vocabulary [31], readability [32], or word morphology [33]. In this section, we briefly describe evaluation measures and text complexity indexes that are part of this study.

3.1 Evaluation Methods and Measures

The ETS without human references has been an active area of research in recent years since traditional methods are expensive and impractical. Based on this, ROUGE-C, LSA, and SIMetrix have been proposed as methods that compare the content between summaries and their source document. Below, we briefly describe these methods and their underlying measures.

ROUGE-C. For automatic assessment, ROUGE is a well-known evaluation package of four measures (ROUGE-N, L, W, and S) that estimate the similarity between the automatic summary and its human references. For instance, ROUGE-N calculates the overlap of n-grams between the summary and its human references (HR), as shown in Eq. (4):

$$ROUGE\text{–}N = \frac{\sum_{S \in \{HR\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{HR\}} \sum_{gram_n \in S} Count(gram_n)}, \quad (4)$$

where n is the n-gram length. $Count_{match}(gram_n)$ is the maximum number of n-grams that co-occur between the summary and HR [9]. However, ROUGE-N and the other measures depend on creating human references. Thus, they cannot work without these documents. To address this situation, ROUGE-C has been proposed in [10].

Unlike ROUGE, ROUGE-C employs source documents to measure their similarity concerning the automatic summary. **Fig. 1** exhibits the differences between ROUGE and ROUGE-C. As observed, ROUGE measures receive human references and automatic summaries as *model* and *test* documents, respectively. On the contrary, ROUGE-C inverts the order of both documents, where the source document is used as a test, and the summary is used as a model.

Based on the differences shown in Fig. 1m ROUGE-N, L, W, S, and SU4 measures can be adapted, generating the following variants: ROUGE-C-N, L, W, S, and SU4. As an example, the evaluation of ROUGE-N without human references is called ROUGE-C-N, which is formally defined in Eq. (5):

$$ROUGE\text{–}C\text{–}N = \frac{\sum_{S \in \{Sum\}} \sum_{gram_m \in S} Count_{match}(gram_n)}{\sum_{S \in \{SD_{Sum}\}} \sum_{gram_n \in S} Count(gram_n)}, \quad (5)$$

where n stands for the length of the n-grams ($gram_n$). $Count_{match}(gram_n)$ is the maximum number of n-grams that co-occur between the summary (Sum) and its source document (SD_{Sum}). In other words, ROUGE-C-N measures the ratio of

n-grams between the summary and its source document(s). Therefore, it is a precision-based measure.

Latent Semantic Analysis (LSA). The LSA is a matrix processing method that represents, extracts and relates the contextual meaning of terms from a document in a "latent" semantic space [34]. For the ETS, the LSA evaluates summaries by measuring the contextual similarity between the summary and its source document(s) [7]. In general, the LSA consists of the following steps:

1 **Preprocessing.** The summary and its source document(s) are preprocessed by eliminating *stopwords* and performing *stemming*. After this, the remaining terms are grouped into n-grams of different lengths, preserving their order in sentences.

2 **Represent input documents as matrices.** The summary and its source document must be represented into two matrices A of $m \times n$ dimensions, where n and m are the number of sentences and terms, respectively. Each value of A (a_{ij}) weights the importance of each i^{th} term in the j^{th} sentence. To weigh the importance of terms, we use term-weighting formulas proposed in [35].

3 **Singular Value Decomposition (SVD).** The matrices A of both documents are decomposed into individual matrices via SVD [36]. The SVD is a matrix factorization method defined in Eq. (6):

$$A = U\Sigma V^T, \tag{6}$$

where U is an $m \times n$ column-orthonormal matrix whose columns are *left singular vectors*. Σ is a diagonal matrix of $n \times n$, whose main diagonal includes nonnegative singular values sorted in descending order. The matrix V has $n \times n$ dimensions, where columns are called *right singular vectors*. Furthermore, each matrix needs to be reduced in r dimensions, obtaining the most important topics of each document. Thus, the order of U is $m \times r$, Σ is $r \times r$, and V^T is $r \times n$.

4 **Matrix similarity calculation.** Once the SVD has been performed, the resultant matrices of both documents are compared through Main Topic Similarity (MTS) or Term Significance Similarity (TSS). While MTS extracts and measures the similarity between the first left singular vector of the summary and source document, TSS extracts singular values and left singular vectors to measure the similarity between the summary and the source document.

Summary Input similarity Metrics (SIMetrix). SIMetrix is an evaluation package that employs 10 similarity and distance measures to evaluate summaries without human references [11]. Some measures are based on *vector space similarity*, *word probability distributions*, and *topic signature words* between the summary and the source document. However, the measures with the best correlation results derive from the Kullback-Leibler (D_{KL}) and Jensen-Shannon (D_{JS}) divergences.

Divergence measures are based on the concept of probabilistic uncertainty or *entropy* [37], whose use was originally proposed to quantify the loss of information between two communication signals [38, 39]. Nevertheless, both measures were later proposed to measure the loss of information between the summary concerning its source document. Formally, D_{KL} is defined in Eq. (7), where P and Q represent the distribution of words of the source document and the candidate summary, respectively:

$$D_{KL}(P \parallel Q) = \frac{1}{2}\sum_w P_w log_2 \frac{P_w}{Q_w}. \tag{7}$$

Although D_{KL} provides nonnegative values, it does not have the symmetric property, it does not satisfy the triangular inequality, and resultant values tend to infinity. For these reasons, the D_{JS} divergence was proposed, which is formally shown in Eq. (8):

$$D_{JS}(P \parallel Q) = \frac{1}{2}\left[\sum_w P_w log_2 \frac{2P_w}{P_w+Q_w} + Q_w log_2 \frac{2Q_w}{P_w+Q_w}\right], \tag{8}$$

where P_w is the probability distribution of the term w in the source document. On the other hand Q_w is the probability distribution of the term w in the

summary. The probability distribution of each term w is calculated according to Eq. (9):

$$P_w = \frac{C_w^T}{N}, Q_w = \begin{cases} \frac{C_w^S}{n_S}, & if\ w \in S, \\ \frac{C_w^T + \delta}{N + \delta \times B}, & otherwise, \end{cases} \quad (9)$$

where P_w stands for the ratio between the number of occurrences of the term w in the source document or T (C_w^T), and the number of terms obtained from the source document and the summary ($N = n_T + n_S$).

On the other hand, Q_w represents the ratio between the number of occurrences of the term w in the evaluated summary or S (C_w^S), and the number of terms retrieved from the summary (n_S), if the term w appears at least once in S. Otherwise, we employ a smoothing operation, where $\delta = 0.005$ and $B = 1.5|V|$. Based on these conditions, SIMetrix can use smoothed and unsmoothe versions of the D_{JS} divergence.

3.2 Text Complexity Indexes

According to CCSSO [40], text complexity is the inherent difficulty of reading and understanding a text, whose measurement depends on several factors. Some of them involve the text's readability, the text's levels of meaning or purpose, text structure, the language's conventionality, and the knowledge demands of the text [41]. Therefore, there are different manners to measure text complexity. This section briefly describes text complexity indexes.

Type-Token Relationship (TTR). The TTR index measures the linguistic diversity of any text document [31]. To calculate the TTR of an input document (d), we use Eq. (10), where $\#Types$ are the number of word types appearing in d. The term $\#Tokens$ represents the number of tokens/words of d. TTR values closer than 1 indicate a higher diversity of words in the document, which also shows a greater complexity:

$$TTR(d) = \frac{\#Types}{\#Tokens}. \quad (10)$$

Ratio of Stopwords (RSW). For many NLP tasks, stopwords are usually uninformative terms that represent noise (*e.g.*, *the*, *of*, *is*, *are*). Based on this, we employ the RSW index, which measures the overall presence of these terms of d. Formally, the RSW is shown in Eq. (11), where $StopWords(d)$ is a function that counts the total number of stopwords in d. $\#Tokens$ represent the length of d. RSW values near 1 indicate a high presence of these terms in d, representing noise and greater complexity:

$$RSW(d) = \frac{StopWords(d)}{\#Tokens}. \quad (11)$$

Ratio of Inflected Words (RIW). Word inflection is present in several languages around the world. For instance, in English, the word *organize* may be modified into multiple forms (*e.g.*, *organizing*, *organization*). Therefore, this characteristic is essential in determining the complexity of terms. In this sense, the RIW is a complexity index that measures the proportion of inflected words from a document.

Formally, it is shown in Eq. (12), where d represents the input document, and $InflectedWords(d)$ is a function that counts the total number of words that belong to some derivation. $\#Tokens$ represent the number of tokens or words of d. RIW values closer than 1 indicate many inflected words in d, meaning greater complexity:

$$RIW(d) = \frac{InflectedWords(d)}{\#Tokens}. \quad (12)$$

Average of Characters per Word (ACW). The ACW index measures the mean word length from one or more documents [15]. Formally, it is shown in Eq. (13), where d represents the text document, $length(w_i)$ is a function that counts the number of characters for each i^{th} word (w_i), and n represents the number of words in d. This index has been widely used for readability assessment since it allows for calculating how long the words of a document can be. While longer words are included in d, the more complex the document is:

$$ACW(d) = \frac{1}{n}\sum_{i=0}^{n} length(w_i). \quad (13)$$

Average of Words per Sentence (AWS). Sentence length is a feature that indicates how understandable and readable a sentence can be. Therefore, this feature is a helpful indicator of complexity in text documents. According to this assumption, we define the AWS in Eq. (14), where d represents the text document, $length(s_i)$ is the

number of words in the i^{th} sentence (s_i), and m is the number of sentences in d. Thus, while AWS values are higher, it indicates longer sentences in d:

$$AWS(d) = \frac{1}{n}\sum_{j=0}^{m} length(s_j). \quad (14)$$

Automated Readability Index (ARI). The ARI measures the degree of readability of text documents, considering word and sentence lengths [42]. This index is formally shown in Eq. (15), where $\#Characters$, $\#Words$, and $\#Sentences$ are the number of characters, words, and sentences in d, respectively. The ARI values vary from 0 to 14, indicating the grade level needed to comprehend d:

$$ARI(d) = 4.71\left(\frac{\#Characters}{\#Words}\right) + 0.5\left(\frac{\#Words}{\#Sentences}\right) - 21.43. \quad (15)$$

Coleman-Liau Index (CLI). The CLI was proposed to gauge the understandability of texts by using the number of characters and sentences per 100 words [43]. Formally, this index is shown in Eq. (16), where L represents the average number of characters per 100 words, and S is the average number of sentences per 100 words. CLI values usually vary from 0 to 17, indicating the degree of comprehension of documents:

$$CLI(d) = (0.0588 \times L) - (0.296 \times S) - 15.8. \quad (16)$$

Word Entropy (H_w). According to [44], entropy is a measure that calculates the average uncertainty of a single random variable. For text complexity, Conroy *et al*. employed entropy to measure the complexity of documents at word and sentence levels [45]. Particularly, word entropy (H_w) is calculated according to Eq. (17), where p_i represents the probability value for the i^{th} term in d. The probability of each word is obtained by dividing the overall frequency of the term i and the total number of terms (n): $p_i = tf_i/n$. Resultant H_w values vary from 0 to 1, where 0 means all words have the same lengths, and 1 represents the opposite:

$$H_w(d) = -\frac{1}{\ln n}\sum_{i=0}^{n} p_i \ln p_i. \quad (17)$$

Sentence Entropy (H_s). Also known as *sentence length uniformity* [46], this index is calculated the same way as word entropy but employs the sum of term probabilities of each sentence. That is, it calculates the average uncertainty of sentence lengths. The H_s index is shown in Eq. (18), where $\bar{p}_j$ is the probability value of the j^{th} sentence, the letter m represents the number of sentences of document d, and n is the number of terms of each j^{th} sentence. The resultant values of H_s vary from 0 to 1, where 0 means a high uniformity of sentences, and 1 represents the opposite:

$$H_s(d) = \frac{1}{\ln m}\sum_{j=0}^{m} \bar{p}_j \ln \bar{p}_j\,, \bar{p}_j = \sum_{i=0}^{n} p_i. \quad (18)$$

ROUGE-N-based complexity indexes. As explained in Section 3.1, ROUGE-N measures the similarity between S and HR by overlapping n-grams. Resultant values are expressed in terms of Recall (R), Precision (P), and F-measure (F) [9]. However, these measures have been used as text complexity indexes by estimating the overlap of n-grams (g_n) between the source document (d_{src}) and a summary that represents the first W words extracted from d_{src} ($S(W)$) [47], as shown in Eqs. (19-21):

$$R_{ROUGE-N}(d_{src}, S(W)) = \frac{\sum_{D \in \{d_{src}\}}\sum_{gram_n \in D} Count_{match}(g_n)}{\sum_{D \in \{d_{src}\}}\sum_{g_n \in D} Count(g_n)}, \quad (19)$$

$$= \frac{P_{ROUGE-N}(d_{src}, S(W))}{\sum_{D \in \{d_{src}\}}\sum_{g_n \in D} Count(g_n) + \sum_{g_n \in S_{B:f}} Count_{unmatch}(g_n)}, \quad (20)$$

$$F_{ROUGE-N}(d_{src}, S(W)) = \frac{2 \times P_{ROUGE-N} \times R_{ROUGE-N}}{P_{ROUGE-N} + R_{ROUGE-N}}. \quad (21)$$

Based on such equations, resultant values near 0 indicate that the most essential information is dispersed throughout the entire d_{src}, and 1 implies the most important information is in the first sentences.

Average Complexity Index (ACI). Besides previous indexes, the arithmetic average of all indexes has been used in [47] to estimate the overall complexity for each source document in the DUC01 and DUC02 datasets. Formally, the ACI is displayed in Eq. (22), where d_{src} is the input source document, $Index_i$ is the i^{th} complexity index that receives d_{src}, and n represents the number of complexity indexes included:

$$ACI(d_{src}) = \sum_{i=0}^{n} Index_i(d_{src}). \quad (22)$$

Furthermore, previous works have employed the before-mentioned 12 indexes ($n = 12$), which were normalized in the range $[0 - 1]$ because it allows assigning the same degree of importance to each one in the process.

4 Proposed Method

In this section, the proposed method is described, which is based on the GA to optimize the selection of measures derived from ROUGE-C, LSA, and SIMetrix methods.

4.1 Computational Cost

The selection of evaluation measures involves assigning each measure a level of complexity, allowing us to determine in what situations an evaluation measure should be selected. Therefore, it is necessary to obtain a vector of values that indicate a selection of appropriate measures depending on the text complexity of the source document. This vector must consider real values between 0 and 1 with five precision digits. Thus, there are 10,000 possible values for just one measure and 310,000 for 31 measures. Finding a balance of these values to improve automatic evaluation needs to be addressed by optimization techniques, such as the GA.

4.2 Genetic Algorithm

The Genetic Algorithm (GA) is one of the most used evolutionary techniques in the state-of-the-art, based on Darwin's natural selection principles to solve optimization problems. Like other evolutionary techniques, the GA represents the solution of a problem through *chromosomes* [48]. The chromosome is a simple data structure whose genes represent individual variables of the problem, and a set of them depicts a *population*. This population is updated according to genetic operators that intend to explore and manipulate the abovementioned variables.

Firstly, the GA generates a random *initial population* of chromosomes. This population is then evaluated according to a *fitness function*, which quantifies the degree of suitability of each solution. The result of this evaluation is obtaining a fitness value per chromosome. As an *a priori* appraisal, some chromosomes may have better characteristics than others, which are selected through the *parent selection operator*.

Once parents are chosen, the *crossover operator* is applied to mix different solution characteristics. However, the chromosomes of this population usually repeat several characteristics. To solve this, the *mutation operator* is used by modifying the minimum parts of chromosomes of the population.

Finally, we obtain a new population of chromosomes, evaluated by the fitness function, and then reintroduced to the selection, crossover, and mutation operators until a *stop condition* is reached (*e.g.*, number of generations). As the GA iterates the genetic operators, we obtain better solutions to the problem.

4.3 Proposed Genetic Operators

Below, it is described what genetic operators were used to select content evaluation measures. Moreover, we explain the preprocessing steps, chromosome encoding, and the fitness function used in the GA.

Preprocessing. Summaries and source documents must be preprocessed by eliminating stopwords and performing stemming. These steps are suggested for each evaluation measure because previous studies have demonstrated that removing unnecessary words and suffixes may improve the precision of evaluation methods [11]. The result of applying evaluation measures is obtaining scores, which will be used for the proposed fitness function. Moreover, source documents are computed by all complexity indexes described in Section 3.2, bringing the ACI for each document.

Chromosome's encoding. For the proposed GA, we employ binary encoding, whose chromosome's genes are binary values (1 and 0).

The length of each chromosome is defined by N segments. Each segment must contain M genes, as shown in Eq. (23):

$$X_i(g) = [s_1, s_2, \dots, s_N], s_j = \left[X_{i,1}(g), X_{i,2}(g), \dots, X_{i,M}(g)\right], X_{i,k}(g) \in \{0,1\}. \quad (23)$$

The expression $X_i(g)$ represents each i^{th} chromosome in the g generation. The variable N represents the number of evaluation measures involved in the proposed selection; the variable M is used to specify the complexity value (henceforth $C(a_j) = [0 - 1]$) and precision for each evaluation measure in binary values. Therefore, each j^{th} segment provides the complexity value for each j^{th} measure.

Initial population. Once we have defined the chromosome's encoding, we must initialize a population of N_{pop} chromsomes. The most usual way to initialize such a population (when $g = 0$) is by generating random binary values for each k^{th} gene from each i^{th} chromosome ($X_{i,k}(0) = Random[0,1]$). Thus, each gene of each j^{th} segment must select binary values between 0 and 1, but the chain of genes of the same segment must represent real values between 0 and 1.

Fitness function. The proposed fitness function is based on the formula shown in Eq. (24), where a_{lj} represents the evaluation score of the measure j for each l^{th} candidate summary. The j^{th} measure belongs to the set of N measures, and it is selected if its complexity value ($C(a_j)$) is greater than or equal to the ACI calculated from the corresponding source document ($ACI(d_{src})$):

$$f(X_i(g)) = \frac{1}{R}\sum_{m=1}^{R} r_m(a_{lj}, b_l), a_j \in \{a_1, \dots, a_N\}, a_j \text{ is selected if } C(a_j) \geq ACI(d_{src}). \quad (24)$$

Notice that this selection is performed for each l^{th} candidate summary to evaluate. Therefore, we seek to maximize the correlation between the selection of measures and human judgments (b_l), considering R correlation coefficients, where r_j is the correlation value from the j^{th} coefficient. Based on N as a parameter, we have optimized under the Pearson, Spearman, and Kendall coefficients.

Parent selection. Parent selection operators employ the fitness value of chromosomes to select and introduce the best ones to the next genetic operators. Typically, selection tends to choose chromosomes of high fitness, following the evolution principle (*i.e.*, if they are crossed, they generally produce better offspring). However, generated offspring could be worse in some cases. Therefore, we have used two genetic operators.

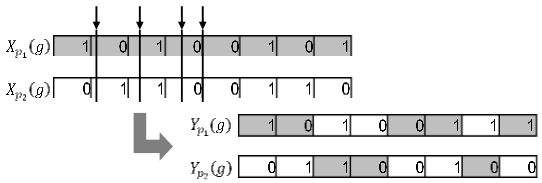

Fig. 4. Uniform crossover

The first one is called *elitism selection*, which chooses n_{elite} chromosomes with the highest fitness from the population to pass them to the next generation. On the other hand, the remaining chromosomes are selected through *tournament selection*, randomly generating $N_{pop} - n_{elite}$ samples of K_{Tourn} chromosomes. Afterward, the chromosome with the highest fitness is chosen from each sample, obtaining a population of parents.

Crossover. As mentioned in previous studies [13], crossover operators perform the genetic exchange of chromosomes to obtain better offspring. For the proposed GA, we have used the *uniform crossover*. This operator generates varying cut points between couples of chromosomes to interchange their genes, as shown in Fig. 4

The chromosomes $X_{p_1}(g)$ and $X_{p_2}(g)$ are two previously selected parents from the tournament selection operator. Both parents are introduced to the uniform crossover operator, mixing their genes based on cut points to obtain $Y_{p_1}(g)$ and $Y_{p_2}(g)$. Each gene may be selected as a cut point, depending on the probability value specified as a parameter.

Typically, this probability is set to 0.5, establishing uniform cut points [49]. Furthermore, this operator considers an additional parameter (P_c), determining whether each couple of parents crosses their genes.

Mutation. The mutation randomly selects genes from a population of $Y_i(g)$ chromosomes, replacing their values with other information to generate a new population ($Z_i(g)$). For the proposed GA, we employed the *flipping operator*, which inverts the value of the m gene in the chromosome $Y_i(g)$, according to Eq. (25):

$$Z_{i,m}(g) = \begin{cases} Random[0,1], & if\ 0 < p_m \leq P_m, \\ Z_{i,m}(g), & otherwise \end{cases} \quad (25)$$

where P_m represents the mutation probability in a range of $[0.0–1.0]$. Therefore, if the random value p_m is in the range $(0–P_m]$, the following random function is applied: $Random[0,1]$. Otherwise, the gene m is not modified ($Y_{i,m}(g)$).

In addition to the flipping operator, the *cataclysmic mutation* operator is also included, preserving the diversity of chromosomes through the restart procedure [50]. This operator is applied in the last generation of the GA, selecting the best chromosome from such generation.

Afterward, the remaining chromosomes are randomly generated to initialize a new population of N_{pop} chromosomes and restart the GA again [51]. Considering this procedure, the number of restarts ($\#Rst$) is used as a parameter in the proposed GA.

Stop condition. Once the GA has applied the genetic operators, it generates a new population of chromosomes. However, it requires iterating these operators several times to explore the search space. Typically, the GA runs until it reaches a certain number of generations (G). Thus, we have used G for the proposed GA.

5 Experiments and Results

This section is divided as follows: First, we present and describe the datasets used to evaluate summaries and select content measures. Second, we describe the configuration of evaluation measures employed for the proposed method. Third, we show the tuning of GA parameters that maximizes the correlation between the proposed selection of measures and human judgments. Fourth, we compare the performance of the proposed selection with state-of-the-art evaluation measures. Finally, we describe the analysis of the obtained results.

5.1 Datasets

To evaluate the performance of the proposed selection of measures, we have employed the DUC01 and DUC02 datasets. Indeed, previous studies suggest using these datasets because they have been widely used to generate and evaluate summaries [2, 3]. The DUC01 dataset contains 309 newspaper documents written in English, which are grouped into 30 collections. Each collection comprises 10 documents that address a particular topic (*e.g.*, natural disasters, bibliographic information, etc.).

This dataset is commonly used for Single- and Multi-document summarization tasks of 100 words, leading to ATS systems being evaluated according to well-defined criteria. In particular, DUC01 holds 1776 summaries evaluated using $Retention_w$ as a human judgment criterion [52].

On the other hand, the DUC02 dataset contains 567 newspaper documents written in English, which are grouped into 59 collections. Each has between 5 and 12 documents addressing topics such as technology, food, politics, natural disasters, finances, etc. Like DUC01, this dataset is used for Single- and Multi-document summarization tasks of 100 words. In addition, DUC02 contains 4107 summaries evaluated through Coverage as a human judgment criterion [53].

5.2 Configuration of Evaluation Methods

According to the overall description of evaluation methods shown in Section “Evaluation Methods and Measures”, we have established certain parameters from them to generate several measures. Below, it is explained the configuration of each.

ROUGE-C. Besides eliminating stopwords and performing stemming, it is necessary to specify what measures were obtained from ROUGE-C. From ROUGE-C-N, we extracted n-grams from 1 to 5 to generate RC-1, 2, 3, 4, and 5, respectively. Furthermore, we employed Longest Common Subsequences (LCS) and skip-bigrams to obtain RC-L and RC-SU4, respectively.

LSA. The measures derived from the LSA depend on the combination of term-weighting formulas proposed in [35].

Based on this, 16 measures were generated from this method using the MTS (see Section “Evaluation Methods and Measures). Table 1 shows the name of each measure.

SIMetrix. As mentioned in Section “Evaluation Methods and Measures”, the best measures of

Table 1. LSA-based measures

Global \ Local	FW	BW	AW	LW
NW	LSA-1	LSA-2	LSA-3	LSA-4
ISF	LSA-5	LSA-6	LSA-7	LSA-8
GF	LSA-9	LSA-10	LSA-11	LSA-12
EF	LSA-13	LSA-14	LSA-15	LSA-16

Table 2. Tuning of GA parameters

No.	G	N_{pop}	K_{Tourn}	P_c	P_m	$\#Rst$
1	100	100	2	0.98	0.00175	3
2	300	800	2	0.98	0.00190	5
3	100	800	2	0.98	0.00190	3
4	100	200	2	0.98	0.00175	3
5	200	200	2	0.98	0.00175	3
6	200	200	2	0.98	0.00180	5
7	200	200	2	0.95	0.00190	5
8	200	500	2	0.90	0.00175	5
9	200	500	2	0.95	0.00175	5
10	200	500	2	0.97	0.00175	5
11	200	500	2	0.98	0.00175	5
12	200	500	2	0.98	0.00195	5
13	300	500	2	0.98	0.00175	3
14	300	500	2	0.98	0.00190	3
15	300	800	2	0.98	0.00190	3
16	500	800	2	0.98	0.00190	3

FW: Frequency Weight, BW: Binary Weight, AW: Augmented Weight, LW: Logarithmic Weight, NW: No Weight, ISF: Inverse Sentence Frequency, GF: GFidf, and EF: Entropy Frequency.

SIMetrix derive from D_{KL} and D_{JS} divergences. However, we do not consider the D_{KL} divergence because it is not symmetrical and generates nonfinite divergence values. From the D_{JS} divergence, we employed n-grams from 1 to 4 to extract different features and generate various measures. In addition to this, we used smoothing functions to obtain smoothed and non-smoothed D_{JS} divergence measures. In total, eight measures were generated, of which four do not use smoothed functions ($JS_1, \ldots, JS_4$), and the remaining four so ($JS{-}S_1, \ldots, JS{-}S_4$).

5.3 Tuning of GA Parameters

The GA parameters described in Section "Proposed Genetic Operators" are frequently used in other studies to adjust the performance of each genetic operator. Thus, we adjusted these parameters to improve the selection of evaluation measures described in the previous section. Table 2 shows the most representative tunings of GA parameters, where each was executed thrice. As observed, we have focused on modifying G and N_{pop} to enable more diversification of chromosomes. Such modifications vary from 100 to 500 generations and 100 to 1000 chromosomes. On the other hand, we keep similar P_c values.

Regarding P_m and $\#Rst$, minor variations were performed across all experiments, since we sought to improve intensification for each generation of the GA and when the population converged in the last generation. Finally, it is worth mentioning that K_{Tourn} remains with the same value (2) in all experiments because this value typically performs better in the GA exploration process.

For each experiment shown in Table 2, training and test sets were defined to evaluate the performance of the proposed GA under the Pearson (P), Spearman (S), and Kendall (K) correlations. Nevertheless, DUC01 and DUC02 do not explicitly consider both partitions of data.

Due to this, we have translated the documents of both datasets into Spanish using the Google Translate API, which is currently available in a Python library (https://pypi.org/project/googletrans). After that, the English documents were used as a training set to optimize the selection of measures, and translated documents were used as a test set to evaluate the performance of the GA.

The translation of documents allows us to evaluate whether the proposed selection of measures is suitable for different languages. It is also necessary because neither DUC01 nor DUC02 provide enough evaluation data to test the proposed method. Previous studies [54] suggest translating documents as an alternative that seeks to preserve the performance of individual evaluation measures when datasets do not provide enough human judgment data.

Table 3 shows the correlation results between the proposed GA-based selection of measures and human judgments on the DUC01 dataset, considering the tuning of GA parameters shown in Table 1 and Table 2.

According to the correlation results shown in Table 3, the performance of the proposed selection of measures in the English language (training set) is improved when we increment G, N_{pop}, and P_m. However, the performance of the GA in the test set is decreased. In other words, the GA produces

Table 3. Correlation results between the proposed selection of measures and human judgments ($Retention_w$) on the DUC01 dataset, considering the tuning of GA parameters

No.	English (Train)			Spanish (Test)		
	P	S	K	P	S	K
1	0.4910	0.4784	0.3348	**0.4675**	0.4398	0.3071
2	0.5110	0.4850	0.3406	0.4287	0.4341	0.3040
3	0.4981	0.4803	0.3368	0.4624	0.4319	0.3013
4	0.5017	0.4731	0.3311	0.4390	0.4270	0.2980
5	0.5044	0.4851	0.3411	0.4671	**0.4480**	**0.3131**
6	0.4969	0.4731	0.3318	0.4585	0.4296	0.3008
7	**0.5175**	**0.4886**	**0.3433**	0.4502	0.4198	0.2923
8	0.4805	0.4694	0.3297	0.4569	0.4298	0.2996
9	0.5058	0.4864	0.3408	0.4492	0.4239	0.2948
10	0.5093	0.4866	0.3419	0.4654	0.4376	0.3054
11	0.4999	0.4777	0.3340	0.4463	0.4163	0.2907
12	0.5048	0.4800	0.3369	0.4435	0.4118	0.2869
13	0.5018	0.4787	0.3355	0.4561	0.4298	0.3005
14	0.5044	0.4813	0.3378	0.4476	0.4194	0.2917
15	0.5006	0.4964	0.3482	0.4118	0.3985	0.2764
16	0.5030	0.4813	0.3380	0.4440	0.4172	0.2898

Note: The best results are highlighted in bold.

Table 4. Correlation results between the proposed selection of measures and human judgments (Coverage) on the DUC02 dataset, considering the tuning of GA parameters

No.	**English (Train)**			**Spanish (Test)**		
	P	**S**	**K**	**P**	**S**	**K**
1	0.6467	0.6171	0.4483	0.6396	0.6058	0.4393
2	0.6469	0.6176	0.4488	0.6385	0.6047	0.4383
3	0.6449	0.6136	0.4455	**0.6412**	0.6072	0.4405
4	0.6443	0.6102	0.4429	0.6409	**0.6091**	**0.4419**
5	0.6481	0.6193	0.4499	0.6257	0.5963	0.4317
6	0.6466	0.6142	0.4461	0.6271	0.5983	0.4334
7	0.6459	0.6151	0.4465	0.6363	0.6025	0.4365
8	0.6483	0.6204	0.4511	0.6346	0.6012	0.4354
9	0.6479	0.6196	0.4502	0.6221	0.5932	0.4292
10	0.6439	0.6123	0.4444	0.6211	0.5950	0.4319
11	0.6462	0.6127	0.4451	0.6323	0.6023	0.4371
12	0.6465	0.6153	0.4469	0.6305	0.6032	0.4372
13	0.6462	0.6198	0.4505	0.6320	0.5999	0.4346
14	0.6467	0.6148	0.4465	0.6257	0.5986	0.4337
15	0.6457	0.6124	0.4448	0.6326	0.6032	0.4374
16	**0.6503**	**0.6220**	**0.4521**	0.6281	0.5976	0.4330

Note: The best results are highlighted in bold.

overfitting. Despite this, the best correlations are obtained from the parameters of the 5th experiment ($G = 200$, $N_{pop} = 200$, $K_{Tourn} = 2$, $P_c = 0.98$, $P_m = 0.00175$, and $\#Rst = 3$), obtaining 0.4094 on average (P: 0.4671, S: 0.4480, and K: 0.3131). Even obtaining the highest correlations (as shown in the 7th experiment) in the training set does not guarantee better performance in the test set.

However, we noticed higher correlation results by incrementing crossover probability and slightly reducing mutation probability. Finally, it is observed that the remaining experiments show lower correlation results.

Table 4 shows the correlation results between the proposed selection of measures and human judgments on the DUC02 dataset, considering the tuning of GA parameters shown in Table 1.

Unlike previous correlation results, the best correlations are obtained when the parameters of the 4th experiment were employed ($G = 100$, $N_{pop} = 200$, $K_{Tourn} = 2$, $P_c = 0.98$, $P_m = 0.00175$, and $\#Rst = 3$), obtaining 0.5640 on average (P:

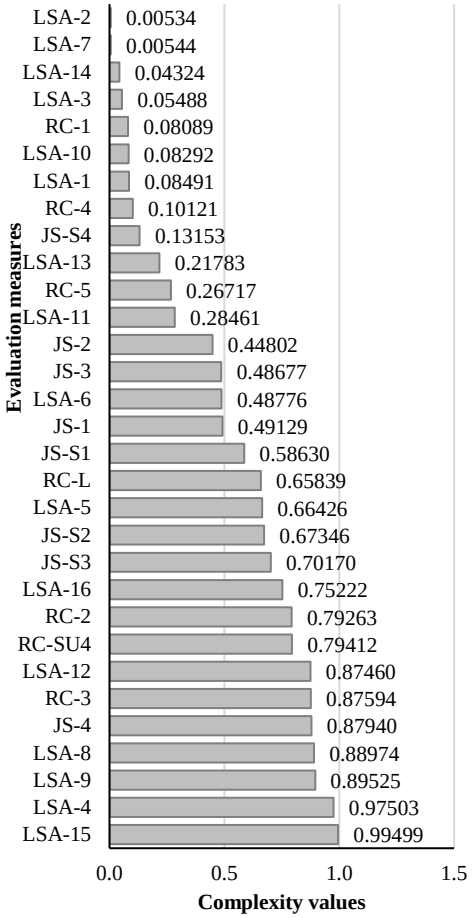

Fig. 5. Complexity values of the proposed selection of measures

0.6409, S: 0.6091, and K: 0.4419). Moreover, we noticed similar results when the parameters of the 3rd experiment were used, obtaining 0.5630 on average.

On the other hand, we have observed that the proposed GA reaches higher correlation results in the training set when we improve exploration and intensification over the next experiments. However, the performance in the test set is reduced. That is, it produces overfitting. Therefore, focusing on exploration over exploitation is necessary because while selecting measures would be more specific, it is not generalizable.

Regarding complexity levels of content measures, Fig. 5 shows an overall ranking of complexity values obtained by the proposed GA to state-of-the-art evaluation measures. In particular, we have observed that the LSA-2, 7, 14, 3, 10, 1, and RC-1 have obtained the lowest complexity values in the ranking, which are lower than 0.1.

This lets us assume that these measures evaluate a small portion of summaries because their corresponding source documents have complexities lower than 0.1. Additionally, the following measures have obtained similar complexities: RC-4, JS-S4, LSA-13, RC-5, and LSA-11.

On the other hand, we noticed that the SIMetrix divergence measures, LSA-6, 5, and RC-L, have obtained higher complexity values, which in turn may indicate that these measures are more frequently selected to evaluate summaries. In other words, the complexity values of their corresponding source documents have achieved similar results.

Finally, it is worth mentioning that the remaining measures have obtained complexities near the highest complexity value possible (1.0). This is because some measures were selected because their source documents obtained similar complexities. Moreover, we have noticed that LSA-4 and LSA-15 were chosen not to evaluate any summary. It means that the proposed GA tends to exclude some measures, assigning them very high or low complexities implicitly.

From the above results, we have chosen the best correlation results (4th and 5th experiment) to compare the performance between the proposed selection of measures and state-of-the-art measures. Table 5 shows this comparison of DUC01 and DUC02 datasets translated into Spanish. The purpose of such a comparison is to highlight the importance of how the proposed selection of measures may be affected across a different language that is not English.

Moreover, how individual measures (*e.g.*, JS_1) and SECO-SEVA [13] may vary in other languages. In addition, we have considered a baseline selection of measures (Avg), averaging the Pearson, Spearman, and Kendall correlations of individual measures to obtain their complexity values.

According to the obtained results, the proposed selection of measures has achieved the highest Spearman and Kendall correlation results on the DUC01 dataset, obtaining 0.4480 and 0.3131,

Table 5. Comparison between the proposed selection of measures and state-of-the-art evaluation measures

Measure	DUC01			DUC02		
	P	S	K	P	S	K
Proposed	0.4390	0.4270	0.2980	0.6409	0.6091	0.4419
Proposed	**0.4671**	**0.4480**	**0.3131**	0.6257	0.5963	0.4317
SECO-SEVA	0.4619	0.4250	0.2964	**0.6472**	0.6120	0.4452
Avg (baseline)	0.4401	0.4042	0.2800	0.6069	0.5691	0.4113
JS_1	**0.4674**	0.4361	0.3040	0.6185	**0.6189**	**0.4506**
$JS{-}S_1$	0.4648	0.4338	0.3019	0.5452	0.6086	0.4402
LSA-3	0.4528	0.4182	0.2906	0.6426	0.6069	0.4402
LSA-15	0.4575	0.4232	0.2944	0.6454	0.6106	0.4434
LSA-11	0.4418	0.4087	0.2838	0.6402	0.6028	0.4373

respectively. Moreover, its performance on the Pearson correlation (0.4671) shows closeness to the highest result (0.4674). In general, these results suggest that the proposed selection improves automatic evaluation even if summaries and source documents are not in English. Moreover, it is worth mentioning that if we only consider the baseline approach, the results would be far from the best evaluation measures.

However, the performance of the proposed selection of measures is competitive with the best measures on the DUC02 dataset, obtaining P: 0.6409, S: 0.6091, and K: 0.4419. Compared to SECO-SEVA and JS_1, it shows proximity on Pearson, Spearman, and Kendall correlations. Although its performance is lower, it shows more stability because all correlations show proximity to the best results.

6 Conclusions and Future Works

In this paper, we propose a selection of content measures to evaluate summaries without human references using a Genetic Algorithm (GA). In addition, 12 text complexity indexes were used to measure source documents' degree of ease or difficulty (see Section “Text Complexity Indexes”). Moreover, we have proposed a GA that seeks to assign complexity values to content measures. The employed fitness function measures the correlation between the optimized selection of measures and human judgments.

According to the results obtained from several experiments, the proposed selection of measures achieves the best correlations on the DUC01 dataset, improving the evaluation of text summaries without human references. Despite the proposed selection of measures shows lower correlations to the best individual measures, it still shows competitive performance, showing proximity to the highest correlation results.

This also suggests that the selection of measures captures the strengths of several individual content measures. For this reason, we propose as future work the inclusion of other evaluation measures (*e.g.*, MoverScore [55] or BERTScore [56]). In addition to this, we seek the inclusion of other text complexity indexes that may help to measure other aspects of complexity of source documents.

Finally, it is worth highlighting that using other GA operators and parameters may improve the process of exploration and intensification of the GA. Moreover, it would be useful using the proposed selection of measures to improve other NLP tasks such as ATS, and Text Classification.

References

1. **Ermakova, L., Cossu, J. V., Mothe, J. (2019).** A survey on evaluation of summarization methods. Information processing & management, Vol. 56, No. 5, pp. 1794–1814. DOI: 10.1016/j.ipm.2019.04.001.
2. **El-Kassas, W. S., Salama, C. R., Rafea, A. A., Mohamed, H. K. (2021).** Automatic text summarization: A comprehensive survey. Expert systems with applications, Vol. 165, p. 113679. DOI: 10.1016/j.eswa.2020.113679.

3. **Rojas-Simon, J., Ledeneva, Y., Garcia-Hernandez, R. A. (2022).** Evaluation of text summaries based on linear optimization of content metrics. Springer, Vol. 1048. DOI: 10.1007/978-3-031-07214-7.
4. **Jones, K. S., Galliers, J. R. (1995).** Evaluating natural language processing systems. An Analysis and Review. Vol. 1083. DOI: 10.1007/BFb0027470.
5. **Cabrera-Diego, L. A., Torres-Moreno, J. M., Durette, B. (2016).** Evaluating multiple summaries without human models: A first experiment with a trivergent model. In: Métais, E., Meziane, F., Saraee, M., Sugumaran, V., Vadera, S. (eds), Natural Language Processing and Information Systems. NLDB 2016. Lecture Notes in Computer Science, Springer, Cham, Vol. 9612. DOI: 10.1007/978-3-319-41754- 7_8.
6. **Mani, I., House, D., Klein, G., Hirschman, L., Firmin, T., Sundheim, B. M. (1999).** The TIPSTER SUMMAC text summarization evaluation. Ninth Conference of the European Chapter of the Association for Computational Linguistics, Vol. 1, No. 1, pp. 77–85.
7. **Steinberger, J., Ježek, K. (2009).** Evaluation measures for text summarization. Computing and Informatics, Vol. 28, No. 2, pp. 251–275.
8. **Lloret, E., Plaza, L., Aker, A. (2018).** The challenging task of summary evaluation: an overview. Language Resources and Evaluation, Vol. 52, pp. 101–148. DOI: 10.1007/s10579-017-9399-2.
9. **Lin, C. Y. (2004).** ROUGE: A package for automatic evaluation of summaries. Text Summarization Branches Out, pp. 74–81.
10. **He, T., Chen, J., Ma, L., Gui, Z., Li, F., Shao, W., Wang, Q. (2008).** ROUGE-C: A fully automated evaluation method for multi-document summarization. 2008 IEEE International Conference on Granular Computing, pp. 269–274. DOI: 10.1109/GRC.2008.4664680.
11. **Louis, A., Nenkova, A. (2013).** Automatically assessing machine summary content without a gold standard. Computational Linguistics, Vol. 39, No. 2, pp. 267–300. DOI: 10.1162/COLI_a_00123.
12. **Cabrera-Diego, L. A., Torres-Moreno, J. M. (2018).** Summtriver: A new trivergent model to evaluate summaries automatically without human references. Data & Knowledge Engineering, Vol. 113, pp. 184–197, DOI: 10.1016/j.datak.2017.09.001.
13. **Rojas-Simón, J., Ledeneva, Y., García-Hernández, R. A. (2021).** Evaluation of text summaries without human references based on the linear optimization of content metrics using a genetic algorithm. Expert systems with applications, Vol. 167, p. 113827, DOI: 10.1016/j.eswa.2020.113827.
14. **Conroy, J. M., Dang, H. T. (2008).** Mind the gap: Dangers of divorcing evaluations of summary content from linguistic quality. Proceedings of the 22nd International Conference on Computational Linguistics, Manchester: Association for Computational Linguistics, 2008, pp. 145–152.
15. **Ellouze, S., Jaoua, M., Belguith, L. H. (2016).** Automatic evaluation of a summary's linguistic quality. In: Métais, E., Meziane, F., Saraee, M., Sugumaran, V., Vadera, S. (eds), Natural Language Processing and Information Systems. NLDB 2016. Lecture Notes in Computer Science, Vol 9612. DOI: 10.1007/978-3-319-41754-7_39.
16. **Ellouze, S., Jaoua, M., Hadrich-Belguith, L. (2017).** Mix multiple features to evaluate the content and the linguistic quality of text summaries. Journal of computing and information technology, Vol. 25, No. 2, pp. 149–166. DOI: 10.20532/cit.2017.1003398.
17. **Ellouze, S., Jaoua, M., Belguith, L. H. (2017).** Machine learning approach to evaluate multilingual summaries. Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source

Types and Genres, pp. 47–54. DOI: 10.18653/v1/W17-1007.

18. **García-Calderón, M. Á., García-Hernández, R. A., Ledeneva, Y. (2019).** Providing order to the handwritten TLS task: A complexity index. Journal of Intelligent & Fuzzy Systems, Vol. 36, No. 5, pp. 4621–4631. DOI: 10.3233/JIFS-179013.
19. **Alaei, A., Pal, U., Nagabhushan, P. (2012).** Dataset and ground truth for handwritten text in four different scripts. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 26, No. 4, p. 25, DOI: 10.1142/S0218001412530011.
20. **Ptak, R., Żygadło, B., Unold, O. (2017).** Projection–based text line segmentation with a variable threshold. International Journal of Applied Mathematics and Computer Science, Vol. 27, No. 1, pp. 195–206. DOI: 10.1515/amcs-2017-0014.
21. **Saabni, R., Asi, A., El-Sana, J. (2014).** Text line extraction for historical document images. Pattern Recognition Letters, Vol. 35, pp. 23–33. DOI: 10.1016/j.patrec.2013.07.007.
22. **Valy, D., Verleysen, M., Sok, K. (2017).** Line segmentation for grayscale text images of khmer palm leaf manuscripts. 2017 Seventh International Conference on Image Processing Theory, Tools and Applications, IEEE, pp. 1–6. DOI: 10.1109/IPTA.2017.8310097.
23. **Villegas, M., Puigcerver, J., Toselli, A. H., Sánchez, J. A., Vidal, E. (2016).** Overview of the imageCLEF 2016 handwritten scanned document retrieval task. CLEF Working Notes, Vol. 1609, pp. 233–253.
24. **Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., Alaei, A. (2013).** ICDAR 2013 handwriting segmentation contest. 2013 12th International Conference on Document Analysis and Recognition pp. 1402–1406. DOI: 10.1109/ICDAR.2013.283.
25. **García-Castro, L. R. (2013).** Suma de visitas de pueblos de la Nueva España, 1548-1550. Universidad Autónoma del Estado de México.
26. **Sánchez Juárez, S. A. (2016).** Amoxcalli. Un análisis sobre la dimensión ontológica de los códices en los archivos, bibliotecas y museos. Icofom Study Series, Vol. 44, pp. 57–68. DOI: 10.4000/iss.676.
27. **Arvanitopoulos, N., Süsstrunk, S. (2014).** Seam carving for text line extraction on color and grayscale historical manuscripts. 2014 14th International Conference on Frontiers in Handwriting Recognition, IEEE. pp. 726–731. DOI: 10.1109/ICFHR.2014.127.
28. **García-Calderón, M. Á., García-Hernández, R. A., Ledeneva, Y. (2018).** Unsupervised multi-language handwritten text line segmentation. Journal of Intelligent & Fuzzy Systems, Vol. 34, No. 5, pp. 2901–2911,. DOI: 10.3233/JIFS-169476.
29. **Arivazhagan, M., Srinivasan, H., Srihari, S. (2007).** A statistical approach to line segmentation in handwritten documents. Document recognition and retrieval XIV, Vol. 6500, pp. 245–255. DOI: 10.1117/12.704538.
30. **Torres-Moreno, J. M., Saggion, H., Cunha, I. D., SanJuan, E., Velázquez-Morales, P. (2010).** Summary evaluation with and without references. Polibits, Vol. 42, pp. 13–20.
31. **Gutierrez-Vasques, X., Mijangos, V. (2019).** Productivity and predictability for measuring morphological complexity. Entropy, Vol. 22, No. 1, p. 48. DOI: 10.3390/e22010048.
32. **Williamson, G. L., Fitzgerald, J., Stenner, A. J. (2013).** The common core state standards' quantitative text complexity trajectory: Figuring out how much complexity is enough. Educational Researcher, Vol. 42, No. 2, pp. 59–69. DOI: 10.3102/0013189X12466695.
33. **Haspelmath, M., Sims, A. D. (2010).** Understanding morphology. 2nd ed. London, United Kingdom: Hodder Education, an Hachette UK Company, pp. 384. DOI: 10.4324/9780203776506.
34. **Sidorov, G. (2019).** Syntactic n-grams in computational linguistics. Cham, Switzerland:

Springer International Publishing, pp. 125-125. DOI: 10.1007/978-3-030-14771-6.

35. **Chisholm, E., Kolda, T. G. (1999).** New term weighting formulas for the vector space method in information retrieval. United States. DOI: 10.2172/5698.
36. **Wall, M. E., Rechtsteiner, A., Rocha, L. M. (2003).** Singular value decomposition and principal component analysis. A practical approach to microarray data, Springer, pp. 91–109. DOI: 10.1007/0-306-47815-3_5.
37. **Shannon, C. E. (1948).** A mathematical theory of communication. The Bell system technical journal, Vol. 27, No. 3, pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.
38. **Kullback, S. (1997).** Information theory and statistics. Courier Corporation.
39. **Lin, J. (1991).** Divergence measures based on the Shannon entropy. IEEE Transactions on Information theory, Vol. 37, No. 1, pp. 145–151. DOI: 10.1109/18.61115.
40. **CCSSO (2010).** Supplemental information for appendix a of the common core state standards for english language arts and literacy: New research on text complexity. www.achievethecore.org/text-complexity.
41. **LDE (2021).** Guide for determining text complexity: Kindergarten through grade 12 overview. Baton Rouge. https://www.louisianabelieves.com/docs/default-source/teacher-toolbox-resources/guide---how-to-determine-text-complexity-grades-k-12.pdf?sfvrsn=7.
42. **Smith, E. A., Senter, R. J. (1967).** Automated readability index. AMRL-TR, pp. 1–14.
43. **Coleman, M., Liau, T. L. (1975).** A computer readability formula designed for machine scoring. Journal of Applied Psychology, Vol. 60, No. 2, pp. 283–284. DOI: 10.1037/h0076540.
44. **Manning, C., Schutze, H. (1999).** Foundations of statistical natural language processing. MIT press. DOI: 10.1145/601858.601867.
45. **Conroy, J. M., Schlesinger, J. D., Rankel, P. A., O'Leary, D. P. (2010).** Guiding CLASSY toward more responsive summaries. TAC 2010.
46. **Rankel, P. A., Conroy, J. M., Schlesinger, J. D. (2012).** Better metrics to automatically predict the quality of a text summary. Algorithms, Vol. 5, No. 4, pp. 398–420. DOI: 10.3390/a5040398.
47. **Mitchell, M. (1998).** An introduction to genetic algorithms. MIT press.
48. **Syswerda, G. (1989).** Uniform crossover in genetic algorithms. ICGA, Vol. 3, No. pp. 2–9.
49. **Eshelman, L. J. (1991).** The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. Foundations of genetic algorithms, Vol. 1, pp. 265–283, DOI: 10.1016/B978-0-08-050684-5.50020-3.
50. **Rathee, S., Ratnoo, S. (2020).** Feature selection using multi-objective CHC genetic algorithm. Procedia Computer Science, Vol. 167, pp. 1656–1664, DOI: 10.1016/j.procs.2020.03.376.
51. **Lin, C. Y., Hovy, E. (2002).** Manual and automatic evaluation of summaries. Proceedings of the ACL-02 Workshop on Automatic Summarization, pp. 45–51. DOI: 10.3115/1118162.1118168.
52. **Over, P., Dang, H., Harman, D. (2007).** DUC in context. Information Processing & Management, Vol. 43, No. 6, pp. 1506–1520, DOI: 10.1016/j.ipm.2007.01.019.
53. **Braun, S., Vasilyev, O., Iskender, N., Bohannon, J. (2021).** Does summary evaluation survive translation to other languages? preprint arXiv:2109.08129. pp. 2425–2435. DOI: 10.18653/v1/2022.naacl-main.173.
54. **Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C. M., Eger, S. (2019).** MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 622.

pp. 563–578. DOI: 10.48550/arXiv.1909.02622.

55. **Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., Artzi, Y. (2019).** Bertscore: Evaluating text generation with Bert. DOI: 10.48550/arXiv.1904.0967.
56. **Phillips, I. T., Chhabra, A. K. (1999).** Empirical performance evaluation of graphics recognition systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 9, pp. 849–870. DOI: 10.1109/34.790427.

Article received on 16/04/2024; accepted on 04/07/2024.
**Corresponding author is Jonathan Rojas-Simón.*