Computers, Materials & Continua
DOI: 10.32604/cmc.2022.031485
Article

Tech Science Press

A Novel Optimized Language-Independent Text Summarization Technique

Hanan A. Hosni Mahmoud[1,*] and Alaaeldin M. Hafez[2]

[1]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[2]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
*Corresponding Author: Hanan A. Hosni Mahmoud. Email: hahosni@pnu.edu.sa
Received: 19 April 2022; Accepted: 30 May 2022

Abstract: A substantial amount of textual data is present electronically in several languages. These texts directed the gear to information redundancy. It is essential to remove this redundancy and decrease the reading time of these data. Therefore, we need a computerized text summarization technique to extract relevant information from group of text documents with correlated subjects. This paper proposes a language-independent extractive summarization technique. The proposed technique presents a clustering-based optimization technique. The clustering technique determines the main subjects of the text, while the proposed optimization technique minimizes redundancy, and maximizes significance. Experiments are devised and evaluated using BillSum dataset for the English language, MLSUM for German and Russian and Mawdoo3 for the Arabic language. The experiments are evaluated using ROUGE metrics. The results showed the effectiveness of the proposed technique compared to other language-dependent and language-independent summarization techniques. Our technique achieved better ROUGE metrics for all the utilized datasets. The technique accomplished an F-measure of 41.9% for Rouge-1, 18.7% for Rouge-2, 39.4% for Rouge-3, and 16.8% for Rouge-4 on average for all the dataset using all three objectives. Our system also exhibited an improvement of 26.6%, 35.5%, 34.65%, and 31.54% w.r.t. The recent model contributed in the summarization of BillSum in terms of ROUGE metric evaluation. Our model's performance is higher than the compared models, especially in the metric results of ROUGE_2 which is bi-gram matching.

Keywords: Text summarization: language-independent summarization; ROUGE

1 Introduction

The substantial amount of electronic data, in different languages, has increased the difficulty of mining useful information from it. It is difficult for people to read such huge articles information. Thus, it is essential to have a computerized summarization technique to deduce the important and prominent information rapidly. Computerized summarization techniques have been utilized for different fields such as web pages and online forms. For instance, the authors in [1] suggested a text token extraction to improve search results. The authors in [2], proposed a text token extracting approach for media analysis.

Language-independent summary extractors are language analysis applications. They target the generation of shorter text from single or multi-text documents while maintaining the meaning. Summarization techniques can be categorized according to the input, language, approach or the output as depicted in Fig. 1 [3,4]. The summarization can be performed on the input of a single text documents or multi text documents. A set of correlated text documents is utilized in the multi-text documents summarization. A one-text documents source will not show inconsistencies, however, in multi-text documents source conflict and redundancy can be found. Therefore, multi-text documents source summarization is more difficult than single source text documents [3–5]. Also, the summarization output can be nonspecific that discourses a huge community, or can be text token-based which emphasizes on specific subjects associated with the text token. This can be significant in categorizing the technique into indicative process [3,4].

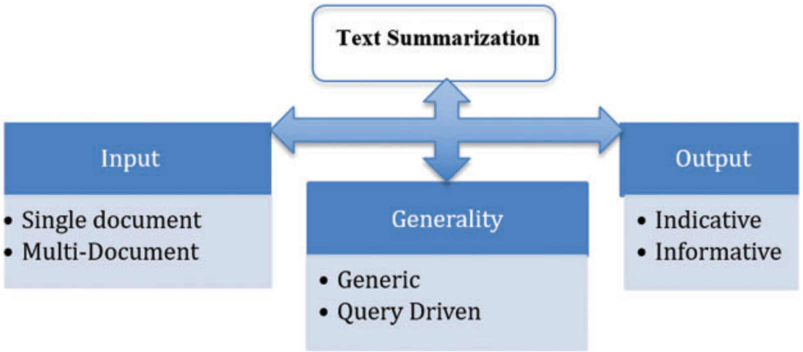

Figure 1: Summarization techniques parameters (input, language, output)

The summarization process can also be defined as extractive where the summarization output is made by choosing the main phrases based on linguistic features and statistical aspects to produce the Weighted-Sum based solutions [3–8]. While, abstracts depend on analyzing the text semantics using natural language processing techniques to produce new phrases that grasps the main ideas in the source text documents [3,4]. The abstracts synopses are more comprehensible and similar to summaries done by humans, but they need profound knowledge of the source text and also need parsers and text generators [6,7]. Deep learning and transfer learning can be utilized in abstract summarization. Deep learning can usually yield good results. Extract-summarization chooses the important phrases utilizing predefined features. The selected phrases are then combined to produce the summarization output. In multi-text documents, the redundancy problem is raised because phrases are mined from several text documents. Redundancy has to be handled in such cases. Also, restricted summarization needs to choose the best chosen summarization output and not the preeminent phrases. Therefore, multi text documents summarization will lead to a global optimization requirement [8–10].

The contributions of this paper are summarized as follows:

1. This paper proposes a language-independent extractive summarization technique.
2. The proposed technique presents a clustering-based optimization technique.

3. The clustering technique determines the main subjects of the text, while the proposed optimization technique minimizes redundancy, and maximizes significance.
4. Experiments are devised and evaluated for different languages to prove the independent feature of the model
5. The experiments are performed on datasets in the languages English, German, Russian and the Arabic languages.

This paper comprises the following: surveys of multi-text documents summarization techniques is presented in Section 2. Section 3 introduces problem definition and mathematical representation. Section 4 proposes the new methodology. In Section 5, results are demonstrated. Section 6 depicts conclusion and future work.

2 Related Work

Language-independent text summarization is categorized into two main methodologies: classical and deep learning techniques [11].

2.1 Classical Techniques

Classical extractive techniques are classified into two techniques. The first approach is the greedy algorithm which chooses a single phrase at a time. The second one is the global technique which examines the best summarization output in place of the best chosen phrases. Such optimization algorithms are NP-hard, and they need heuristics algorithms such as population [12–14]. Various algorithms are presented for fast summarization techniques.

In Greedy technique, a single phrase is selected using a pre-selected features to be incorporated in the summary. This technique is time-efficient and unpretentious, but has the limitation of not yielding an acceptable summarization because of data redundancy. Many algorithms are presented in this technique such deep learning techniques.

Statistical algorithms are usually utilized in language-independent text summarization using relevance score and statistical classifier [13]. In this techniques, they utilized features such as Term Frequency, phrase position and length to mirror the significance of a phrase in the source file [13–19]. Theses algorithms are utilized for multiple source text documents summarization. They are utilized also to improve the mixture of the important phrases and to exclude redundancy. The main problem is the lack of understanding the semantic of the text [14].

On the other hand, in machine learning methods, language-independent models are producing bi-arrangement output. Multiple texts and their extracts are utilized in the learning-phase data. Single phrase is categorized as a summarization output phrase using statistical and/or semantic set of features [20–23]. According to the authors in [23], machine learning techniques are well fit for the summarization of text documents. Studies have presented the efficiency of this technique [24], but, it need labeled data for supervised training, which is time-consuming.

Global summarization explores the best summarization output and not the phrases. This technique yields better output than stochastic technique, but it requires more computational complexity. Many algorithms are employed in this technique such as clustering approaches. The problem, with this approach, is the lengthy time required to obtain the summary.

In Graph, approaches, each text document is stored as a tree with a set of nodes, and set of arcs between the vertices. Each single phrase is represented as vertex, and each edge represents the connection between two related phrases. Each edge is assigned a weight that resembles the two vertices similarity. They utilize dice coefficient metric to represent the resemblance degree of any two phrases. An edge is defined when the resemblance degree is larger than the threshold [25–30]. This approach is used for text token summarization and requests that phrases are only selected from relevant sub-graph [29]. The main problem with graph techniques is the lack to understand the source text as it utilizes statistical metrics.

2.2 *Deep Learning Techniques*

Deep learning and neural text summarization achieve better results than classical methods [30–35]. Deep learning techniques involve a lengthy training phase. Extractive and abstractive deep learning techniques trail the same pipeline as follows:

I. Words are represented as continuous vector using Word2vec-alike algorithms.
II. Text documents are encoded utilizing word embedding techniques which encode and extract text features
III. The text representation is added as an input to regression simulations to rank phrases (extractive) or decoders to generate phrases (abstractive) [30].

Abstractive models focalize on salient features caption (meaning) and then create an abstract like human-written abstracts. Phrase-to-phrase recurrent neural network (RNN) architectures are the prevailing framework for abstract summarization [31]. In this model, an encoder symbolizes token in the input, the decoder generates the summary of the vector encoded. A beam search process is utilized to capture the best sequence to generate the summary [32].

As a summary, the main challenges for the Arabic summarization process, are the lack of multi-phrase sets for abstract summarization and the difficulty of the Arabic language. Another problem is that ROUGE metric is not a measure of relevance. The ROUGE metric uses exact word matching, while abstracts may rephrase some words into different ones with similar meanings. Also, abstracts may create fake facts, as 35% of abstracts summaries created by this technique face this problem. Other challenges such as phrase repetition and inaccuracy are also stated [35–40].

3 Problem Definition

The phases of the proposed system are as follows:

a) The first phase is the pre-processing phase which comprises tokenization, stop-tokens deletion of the related text documents sources to convert the source text into a unified one.
b) The second phase defines the informative features and extracts the new representation from each single phrase as a depiction of the significance score.
c) The third phase uses the C-Means clustering technique to classify the key themes in the source texts.
d) The next phase utilizes a multi-objectives optimization algorithm to maximize coverage and significance.
e) We evaluated the presented technique using BillSum [30], MLSUM [31], and Mawdoo3 [32] datasets, and the results revealed that our technique performs better than other peer systems.

In the following subsections, we are going to discuss the problem definition and representation of multiple text documents text summarization.

3.1 Problem Formulation

Text documents $D = (D_1,\ D_2\ ,\ \ldots D_m)$, where, m is the related text files count. A text document has a group of phrases $P\,(Di) =\ (P_1,\ P_2\ ,\ \ldots P_n)$, where n is the count of phrases in D_i. The objective is to generate a text summary $D' \subset D$ where, D' represents selected phrases from the set D.

The summarization objectives are defined as follows:

a) Relevance is defined as the choice of high score relevant and informative phrases from the set D.
b) Coverage is defined as the choice of phrases that cover important subtopics from the set D to include all the original information.
c) Redundancy ensures that the selected phrases do not contain redundant repeated information.
d) Length of the created summary, compared to the original text documents, should have enough length ratio, which should be defined in progress to optimize coverage and redundancy.

3.2 Phrase Mathematical Representation

Phrase mathematical representation is the key task of the language processing procedures. It encodes phrases into vectors. Numerous algorithms were defined for phrase mathematical representations, such as word embedding [39]. In related text documents, each single phrase is defined by a real-valued bag-of-words. Let $T =\ (T_1,\ T_2\ ,\ \ldots T_m)$ defines all unique words in a text documents D, m is the count of non-similar words. Each signal phrase $P_j =\ (W_{j1},\ W_{j2}\ ,\ \ldots W_{jm)}$ is defined as a m-dimensional vector, W_{jk} is defined as the weight of word t_k in phrase P_j. The term's weight is computed utilizing inverse term frequency process (*ITF*) [36]. *ITF* is a distinct version of a process that is used to define significant content in a text documents in the corpus. *ITF* is normalized by dividing its value to the total text documents count enclosing the term k. ITF process syndicates term frequency with the inverse phrase frequency value, to compute the compound weight for each single term in each single phrase. *ITF* is utilized to identify the local importance of the word in each phrase. The *ITF* of t_k in phrase P_j is computed as follows:

$$ITF_j = ITF_{kj}\ \times log_2\left(\frac{N}{n_k}\right) \tag{1}$$

where, ITF_j indicates the occurrences count of the term t_k in the single phrase P_j. The symbol N denotes the total phrases' count, and n_k is the total number of phrases including the term tk. *ITF* metric performance is less than word embedding technique. However, word embedding techniques need huge structured data set and are essentially utilized in abstract summarization deep learning methods. Our proposed approach is an extractive technique which doesn't understand the phrase semantics.

4 The Proposed Language-Independent Text Summarization Technique

In this section, we are proposing a multi-documents Language-independent extractive summarization technique that engages a multi-objective optimization and clustering techniques. The proposed model extracts the most substantial phrases that represent the key topics of the original text while eradicating redundancy of the created text. Fig. 2 displays important phases of the model. The text contents are converted to a combined text document. In the second phase, phrases will be analyzed as bag vectors with the ITF algorithm. Then, a group of features will be selected to prompt the best score of each phrase. Clusters will be formed to detect the subjects that reside in the documents. In the last phase, the relevant summary is computed by an optimization technique that maximizes significance, coverage and also diversity.

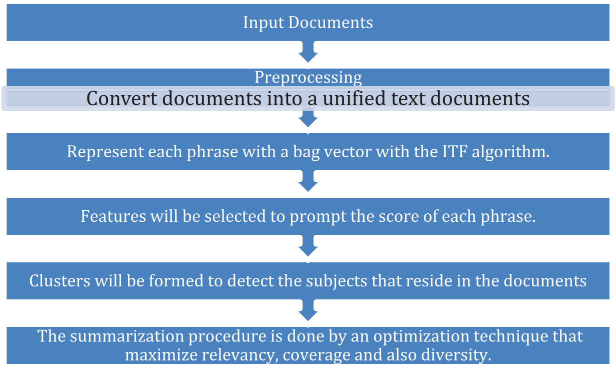
Input Documents
Preprocessing
Convert documents into a unified text documents
Represent each phrase with a bag vector with the ITF algorithm.
Features will be selected to prompt the score of each phrase.
Clusters will be formed to detect the subjects that reside in the documents
The summarization procedure is done by an optimization technique that maximize relevancy, coverage and also diversity.

Figure 2: Flow diagram of the proposed model phases. Phrases are analyzed as bag vectors and a group of features are selected to compute the best score of each phrase

4.1 Text Preprocessing Phase

The Preprocessing phase targets different language processing challenges by converting the source text documents to simplify the using of the other phases as calculating phrase like hood value. Preprocessing also targets the elimination of the words ambiguity and inconsistency for a better representation. This phase contains two processes namely: tokenization and normalization followed by stop-token removal [40]. We depend on recent studies to select the best preprocessing procedures. We also select CoreNLP language processing tools to aid in these preprocessing methods. We investigated experimentally the outcome of various tokenization processes on text summarization, without dealing with like typos and mistakes issues.

4.1.1 Tokenization

Tokenization targets the splitting of the text documents into smaller partitions such as phrases, and words and sometimes paragraphs [33]. This process is related to morphological study process. This process is usually difficult when handling rich languages that have complicated morphology such as Arabic language. Text tokenization is executed at the phrase and word levels to calculate the phrase score, which utilize the punctuation“.,!,;, and ?”. Tokenization is used to denote phrases as a set using space. We also investigate the semantic tokenization process using CoreNLP of punctuation marks delimiters. This approach is beneficial in regards of the presence of punctuation errors. Tab. 1 displays an input text with phrase tokenization via punctuation and semantic processes. Punctuation tokenization yields two phrases while semantic tokenization outputs one phrase.

Table 1: An input text with phrase tokenization via punctuation process (punctuation tokenization yields two phrases, while semantic tokenization outputs one phrase)

Input Text	أن مجلس العلماء سلط الضوء على الأنباء لإعداد كتيب للمراحل العلمية في العام الحالي، تواصل المدير مع وزارة التعليم العالي، والتي أكدت تلك الاخبار.
Punctuation tokenization	أن مجلس العلماء سلط الضوء على الأنباء لإعداد كتيب للمراحل العلمية في العام الحالي تواصل المدير مع وزارة التعليم العالي والتي أكدت تلك الاخبار
Normalization	أن مجلس العلمء سلط الضوء على الأنبء لإعداد كتيب للمراحل العلمية في العام الحال تواصل المدير مع وزارة التعليم العالي والتي أكدت تلك اخبار
Stop-token removal	مجلس العلمء سلط الضوء الأنبا إعداد كتيب مراحل العلمية العام الحالي تواصل المدير وزارة التعليم العالي أكدت تلك اخبار

4.1.2 Phrase Length

Phrase length can compute the information contents in a phrase. Lengthy phrase is a metric of high information content. It totals the count of words in a phrase divided by the size of the lengthiest phrase, which is computed mathematically as:

$$length\,(P_i) = \frac{No.\ of\ terms\ in\ P_i}{No.\ of\ words\ in\ the\ longest\ phrase} \tag{2}$$

Numerous methods are proposed to figure the phrase score [39], the performance relates phrase-based voting techniques such as BordaFuse and CombANZ. We approve, for all the features, a linear sum of normalized scores to estimate each phrase value as follows:

$$Score\,(p_i) = h_1 * title\ similarity\,(p_i, t) + h_2 * keywords\,(p_i) + h_3 * location\,(p_i) + h_4 * length\,(p_i) \tag{3}$$

The weight h_i of a feature replicates its rank and therefore impact the calculation of the total rank. using results, we define ranks as follows: $h_1 = 3$, $h_2 = 2$, $h_3 = 1$, and $h_4 = \frac{1}{2.5}$.

4.2 Topics Selection for Clustering

Related text documents usually have the same group of subjects. To define these subjects, we will have employed clustering algorithm [31–36]. We selected a partitioning algorithm (C-Means). This method is extensively used to defeat the shortcomings of the K-means algorithm. C-Means employs nonlinear similarity measures for cluster's representation. (C_i) represents a point in the cluster with a minimum dissimilarity with other points (O_i) in the cluster. (C_i) is a point that is analogous in concept to centroids, but still is a point in the data set. The dissimilarity (C_i, O_i) is computed by utilizing any distance metric like Rectilinear distance d (Abs $(C_i - (O_i))$. The C-Means algorithm selects the cluster centroid that reduces the sum of the d from this centroid. If k is the count of clusters, P is the group of phrases in the texts represented using its ITF algorithm. C_j is the centroid of the cluster k_j. The C-Means process for phrase clustering is depicted by the following [32]:

A. Define K predefined clusters.
B. Select k-phrases, randomly, from P to be the initial centroids.
C. Assign each phrase $P_i \in$ P to the predefined cluster K_j with the most adjacent centroid using Rectilinear distance, $P_i \in K_j$, where $|C_i - P_i|$ is the minimum value for all K_j.
D. Re-compute the centroid C_j for the predefined K_j by selecting the phrases that reduce the sum w.r.t all other phrases, select C_j where:

$$\text{Sum of differences} = \sum C_j \sum s_i \in C_j,\ \text{where}\ |C_i - s_i|\ \text{is minimum}$$

E. Repeat steps A and B until the value of the centroids become constant.

The count of predefined k defines the count of various subjects that are present in the text files. Since k is predefined by the consumer, it can be time inefficient to acquire its minimum value. Therefore, we used the elbow algorithm to determine the optimal value of k. Elbow measure defines cluster cohesion, and how close are the enclosed objects. On the other hand, separation defines the mutual exclusion metric between clusters. For each phrase P_i, a(i) is defined as the mean distance between Pi and all other phrases in K_j. $\forall C \neq C_i$, define d(s_i, C) as the mean distance between P_i, and the objects O in cluster K_j. Then compute $d(s_i$, C) $\forall\ C \neq C_i$. $b\,(i)$ is the least distance. The elbow_measure of a phrase P_i is calculated as follows:

$$elbow\_measure\ \ (P_i) = \frac{b\,(i) - a\,(i)}{\max\,(b\,(i)\,, a\,(i))} \tag{4}$$

a) Defining the objective functions

We define multi objective functions that are used to rank the generated solutions. They are coverage and significance. Also, we include diversity objective to produce better summarization. The objectives are defined as follows:

- Coverage ensures that the summary includes all essential contents and important subjects that occur in the texts. Coverage is depicted as the similarity between phrases Pi where Pi is an element in both the Summary and the documents' topics. This similarity is defined by the centroids C of the generated clusters. Therefore, the function fcoverage is maximized as follows:

$$\text{fcoverage}\,(\text{X}) = \sum_{P_i \in Summary} \sum_{C_j \in \text{C}} - \text{dice}\ \left(P_i,\ C_j\right) \tag{5}$$

where, dice is defined as the dice coefficient metric, P_i represent the i$^{\text{th}}$ phrase, C_j is the centroid of the cluster K_j.

- Diversity goal is to decrease redundancy in the output summary by not repeating the same information using multiple phrases. We compute the redundancy using the similarity measure between phrases in the generated summary:

$$\text{fredundancy}\,(\text{X}) = \sum_{i=1}^{N1} \sum_{j=1}^{N1} \text{dice}\ \left(P_i\,, P_j\right) \tag{6}$$

where, dice is the dice coefficient metric and N1 is the count of phrases in the generated summary. The model attempts to minimize redundancy to produce a better summary.

Crossover is utilized to enhance diversity in the population [37–40]. Crossover models have been proposed, such as the binary crossover which utilizes probability density to simulate the crossover operator of the phrase representation [32].

- Significance is calculated by the phrase score and describes the importance phrases in the generated summary. We intend to maximize the significance as an objective, that support the phrases with higher rank to be enclosed. We compute the significance objective as follows:

$$f_{\text{significance}}\ \ (\text{X}) = \sum_{Pi \in \text{Summary}} Score\,(P_i\ ) \tag{7}$$

Finally, we can say that the proposed model is an optimization problem defined as following:

$$f_{model}\,(x) = \max\left(f_{\text{coverage}}\ ,\ f_{\text{significance}},\ \frac{1}{f_{\text{redundancy}}}\right) \tag{8}$$

b) Generate text summarization

The optimization procedure produces a number of maximum Weighted-Sum solutions. Thus, we have to choose the best solution utilizing user requirements. We use a majority voting algorithm to aggregate the solutions [38]. To produce the final solution, we implement the majority voting algorithm over every dominated solution. The summary is generated by selecting the phrases that gain majority as solutions as depicted in Tab. 2. When the majority output is lengthy, we will delete phrases with low scores. The process is depicted in Tab. 3.

Table 2: Majority voting algorithm (The summary is generated by selecting the phrases that gained majority)

Weighted-Sum optimal solutions	0	1	1	1	1	1	1	0	1
	0	1	0	1	0	1	1	1	1
	0	1	0	1	1	0	1	0	0
	0	1	0	1	1	0	1	1	1
	0	0	1	1	0	0	1	0	0
Majority	0	1	0	1	1	0	1	0	1

Table 3: Multi-objective optimization algorithm using majority voting

```
Algorithm: Perform-Optimization (Input: Candidates Count, generation size max)
Start
1. Candidate count is set to the number of documents, max is set to the largest number of phrases
   in the candidate documents.
2. Initialize the candidates to the first phrase of each document (initial population)
3. Compute the objective value for each candidate
4. Allocate rank using Weighted-Sum sort algorithm
5. Generate the offspring
6. Apply binary selection among the offspring
7. Apply crossover
8. Calculate objective values of the offspring populations
9. for i = 0 to max
       for each candidate and offspring in the population
       {Allocate rank using Weighted-Sum algorithm
       generate the dominate fin-fronts
       calculate CD between the fin-fronts
       end for
       Select fin-fronts with the highest CD
       Produce next solution
       Apply binary selection
       Apply cross overCalculate objective function of the children
       Merge the candidates and their offspring
end for
10. Output: optimal solution
End
```

5 Experiments

In this section, we are evaluating the efficacy of our summarization model by describing the conducted experiments. Datasets are described in Section 5.1. Evaluation metrics are discussed in Section 5.2. The results are reported and discussed in Section 5.3. Finally, we compared our model with other related models in Section 5.4.

5.1 The Dataset

To evaluate our model, we utilized the public datasets BillSum [30], MLSUM [31], and Mawdoo3 [32]. These datasets are summarized for English, Arabic, German and Russian languages. Tab. 4 presents the datasets statistics. Where, reference sets are set of reference documents summarized by experts and used as ground truth.

Table 4: The datasets statistics (English, Arabic, German and Russian languages)

	BillSum (English) [30]	Mawdoo3 (Arabic) [32]	MLSUM (German) [31]	MLSUM (Russian) [31]
Number of documents	767	1050	502	600
Number of sentences	87,540	100,500	68,023	80,90
Number of words	699,630	505,675	340,021	450,320
Number of different words	19,500	22,210	12,000	13,000
Number of reference sets	67	76	50	60
Documents per reference sets	10 (average)	10(average)	8 (mean)	9 (average)
Number of summarized output	3	4	3	4

5.2 Evaluation Metrics

We utilize the ROUGE evaluation metric [39] for performance evaluation of our proposed model. ROUGE is an acknowledged metric and is the official metric for computerized evaluation of text-document summarization. ROUGE measures the superiority of the output by calculating similarity score of the output *vs.* the ground truth summarized by a human. Similarity is calculated by including all overlapping terms using *Ngram*, term sequences (*ROUGE_L*) and term pairs (*ROUG_SU*). In our performance evaluation, we utilize the metrics: *ROUGE_N* (N= 1, 2, 3 and 4). The *ROUGE_N* metric matches *Ngram* of the generated text summary and reference summary and then calculates the count of matches. It is computed as follows:

$$ROUGE\_N = \frac{\sum_{S\, \in\, summary(R)} \sum_{Ngram\, \in\, S} Count\ (matched\ NGram)}{\sum_{S\, \in\, summary(R)} \sum_{Ngram \in S} Count\ (NGram)} \tag{9}$$

5.3 Experimental Results

The first experiment reveals the performance of the pre-processing techniques in text document summarization. The datasets are tokenized punctually via CoreNLP tool. Tabs. 5 and 6 present the outcome of the pre-processing on the summarization process. They display the results of tokenization alone and tokenization using punctuation. It is obvious from Tab. 5, that the best performance was achieved when the punctuation is utilized in the tokenizer. For the English language, the technique accomplished F-measure of 0.474, 0.304, 0.208, and 0.172, on the average, for ROUGE_1, ROUGE_2, ROUGE_3, and ROUGE_4, respectively. The F-measure results for the other languages are detailed in Tab. 5. In detail, the punctuation tokenizer is very stable given that the data is correctly punctuated, while the tokenizer alone is better for data without punctuations. The used datasets are fully punctuated with full marks. They describe the reason that punctuation tokenizer has better performance than the alone tokenizer. Since the optimization procedure utilizes random variables to control the crossover operators, we will not depend on single run observation. We will compute the mean F-measure of 20-Independent runs. Also, we will employ the same concepts in comparison with similar systems using the same dataset. MLSUM for German and Russian and Mawdoo3 for the Arabic language.

Table 5: The mean F-Measure of ROUGE_N of tokenization combined with punctuation removal for four languages

Data set	Tokenization combined without punctuation				The proposed tokenization combined with punctuation removal			
F1-measure of 20 runs	ROUGE_1	ROUGE_2	ROUGE_3	ROUGE_4	ROUGE_1	ROUGE_2	ROUGE_3	ROUGE_4
BillSum (English)	0.314	0.214	0.148	0.102	0.474	0.304	0.208	0.172
(Arabic)	0.302	0.202	0.136	0.09	0.462	0.292	0.196	0.160
MLSUM (German)	0.291	0.191	0.125	0.079	0.451	0.280	0.175	0.152
MLSUM (Russian)	0.25	0.17	0.114	0.063	0.42	0.24	0.164	0.141

The second experiment reveals the performance of maximizing the objective named significance score. Tab. 6 demonstrates the results of maximizing coverage function and diversity function only. Tab. 7 demonstrates the results of maximizing both coverage and diversity, as well as significance. The results, for the English language, prove that maximization increases the efficiency of the results considerably with a mean improvement of 38.5%, 34.8%, 23.4% and 18.1% respectively, as depicted in Tabs. 6 and 7. This increased efficiency stems from the significant features that are included in the significance function such as phrase position, which are not included in either the coverage function or the diversity function.

Table 6: The mean F-Measure of ROUGE_N of tokenization (alone) with and without significance objective

Data set	Tokenization alone combined with punctuation without Significance Objective				Tokenization alone with coverage, diversity and significance			
F1-measure of 20 runs	ROUGE_1	ROUGE_2	ROUGE_3	ROUGE_4	ROUGE_1	ROUGE_2	ROUGE_3	ROUGE_4
BillSum (English)	0.324	0.220	0.158	0.142	0.484	0.334	0.218	0.179
Mawdoo3 (Arabic)	0.312	0.209	0.146	0.129	0.472	0.299	0.209	0.170

(Continued)

Table 6: Continued

Data set	Tokenization alone combined with punctuation without Significance Objective				Tokenization alone with coverage, diversity and significance			
MLSUM (German)	0.299	0.198	0.135	0.089	0.461	0.289	0.185	0.159
MLSUM (Russian)	0.291	0.179	0.124	0.073	0.429	0.251	0.174	0.151

Table 7: The mean F-Measure of ROUGE_N of punctuation tokenization with and without significance objective

Data set	Tokenization and punctuation with coverage and diversity (without significance)				Tokenization and punctuation with coverage, diversity and significance			
F1-measure of 20 runs	ROUGE_1	ROUGE_2	ROUGE_3	ROUGE_4	ROUGE_1	ROUGE_2	ROUGE_3	ROUGE_4
BillSum (English)	0.354	0.260	0.178	0.162	0.499	0.354	0.238	0.199
Mawdoo3 (Arabic)	0.342	0.249	0.156	0.139	0.482	0.311	0.229	0.180
MLSUM (German)	0.311	0.214	0.145	0.099	0.459	0.294	0.195	0.169
MLSUM (Russian)	0.303	0.189	0.134	0.086	0.434	0.263	0.187	0.160

5.4 Comparing with Other Related Works

We conducted an objective study comparing our proposed model against similar language-independent extractive document text summarization published models. The comparison is depicted in Tab. 8 which delivers a comparison of performance of our proposed model against other similar models on BillSum dataset utilizing the ROUGE metric. The results indicate that our model outperforms all other models using BillSum dataset. Our system exhibits an improvement of 26.6%, 35.5%, 34.65%, and 31.54% w.r.t. to the compared models using BillSum in terms of Rouge-N from 1 to 4, respectively. It is found that the proposed model's accuracy is higher than compared models, from the result of the ROUGE_2 which is bi-gram matching.

Table 8: A comparison of performance of our proposed model against other similar models on Mawdoo3 (Arabic) dataset

Models	Results			Improvement of our proposed punctuation tokenization combined with punctuation w.r.t. other models		
	Recall	Precision	F-measure	Recall	Precision	F-measure
Model-1 [40]	0.48	0.42	0.428	+31.4%	+50%	+60.3%
Model-2 [35]	0.48	0.49	0.88	+21%	+23%	+21.2%

(Continued)

Table 8: Continued

Models	Results			Improvement of our proposed punctuation tokenization combined with punctuation w.r.t. other models		
	Recall	Precision	F-measure	Recall	Precision	F-measure
Our proposed model using tokenization and punctuation with coverage and diversity (without significance)	0.46	0.44	0.488	+27.8%	+45.5%	+44.5%
Our proposed model using tokenization and punctuation with coverage, diversity and significance	0.86	0.88	0.899	-	-	-

5.5 Computational Time Comparison

In this section, we are establishing the computational time required to construct summaries comparing our proposed model (using different parameters including tokenization and punctuation with coverage, diversity and significance) against state of the art models Model-1 [40] and Model-2 [35]. From the database Mawdoo3 (Arabic) (see Fig. 3). As depicted from the figure our model has an advantage in computational time of extracting a phrase in the final summary in seconds *vs.* Number of phrases in thousands in the Mawdoo3 (Arabic) Reference sets.

Experiments establish that the cost of extracting N phrases in the final summary has an upper bound of $O\ (N\ Log\ (N))$. Therefore, training time is $O\ M\ (N\ Log\ (N))$. We computed the CPU time that our model consumed to form the final summarization text, as depicted in Fig. 4.

5.6 Discussion

The experiments depicted the performance of maximizing the significance score, coverage function and diversity function. The results demonstrated that maximizing both coverage and diversity and increased the efficiency of the results with an average accuracy of 90%. The comparative study of our proposed model against state of the art summarization models using BillSum dataset is depicted using the ROUGE metric. The results indicated that our model outperform all other models with an increase of 35.5 using the ROUGE_2 metric, which is bi-gram matching.

Also, the CPU time of our model has an advantage in extract acting time of a phrase in the final summary *vs.* other compared models.

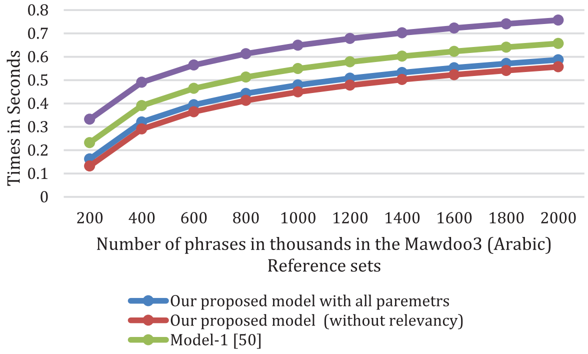

Figure 3: Average time to extract a phrase in the final summary in seconds *vs.* Number of phrases in thousands in the Mawdoo3 (Arabic) Reference sets

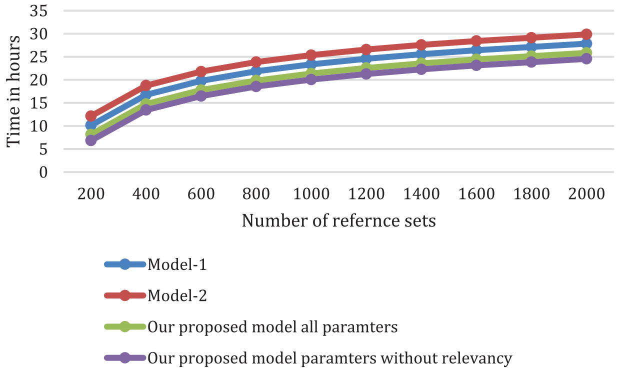

Figure 4: The comparison of the cost of average CPU time for training phase for different models in hours

6 Conclusion

In conclusion, we formulated the multi-language-independent text summarization process as an objective optimization process (maximize multiple objectives simultaneously). The proposed model employs four phases: the first phase is the preprocessing process followed by feature extraction and clustering, while the last phase is the multi-objectives simultaneous optimization. Sentences are modeled in a unified form through preprocessing such as tokenization, stop word removal and normalization. Statistical features are selected and used for significance scoring for each phrase. Topics of the related documents are defined using centroid clustering. The last phase generates an optimal summary using a multi-objectives optimization evolutionary method, maximizing significance and minimizing redundancy. Results verify the efficacy of our model over the state-of-art models by measuring the ROUGE metric.

Still we have some limitations as follows (i) Sentence score is computed experimentally, it could be computed through Genetic Algorithm, and (ii) We did not include coherency of the output, we could have included it to the objectives to be optimized.

Acknowledgement: We would like to thank Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R113), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R113), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Qumsiyeh and Y. Ng, “Searching web text documents using a classification approach,” *International Journal of Web Information Systems*, vol. 12, no. 1, pp. 83–101, 2019.

[2] A. Pal and D. Saha, “An approach to automatic language-independent text classification using simplified lesk algorithm and wordnet,” *International Journal of Control Theory and Computer Modeling (IJCTCM)*, vol. 3, no. 4/5, pp. 15–23, 2020.

[3] 4.M. Al-Smadi, S. Al-Zboon, Y. Jararweh and P. Juola, “Transfer learning for arabic named entity recognition with deep neural networks,” *IEEE Access*, vol. 8, pp. 37736–37745, 2020.

[4] M. Othman, M. Al-Hagery and Y. Hashemi, “Arabic text processing model: Verbs roots and conjugation automation,” *IEEE Access*, vol. 8, pp. 103919–103923, 2020.

[5] H. Almuzaini and A. Azmi, “Impact of stemming and word embedding on deep learning-based arabic text categorization,” *IEEE Access*, vol. 8, pp. 27919–27928, 2020.

[6] M. Alhawarat and A. Aseeri, “A superior arabic text categorization deep model (SATCDM),” *IEEE Access*, vol. 9, pp. 24653–24661, 2020.

[7] S. Marie-Sainte and N. Alalyani, “Firefly algorithm based feature selection for arabic text classification,” *Journal of King Saud University Computer and Information Sciences*, vol. 32, no. 3, pp. 320–328, 2020.

[8] T. Sadad, A. Rehman, A. Munir and T. Saba, “Text tokens detection and multi-classification using advanced deep learning techniques,” *Knowledge-Based Systems*, vol. 84, no. 6, pp. 1296–1908, 2021.

[9] M. El-Affendi, K. Alrajhi and A. Hussain, “A novel deep learning-based multilevel parallel attention neural (MPAN) model for multidomain arabic sentiment analysis,” *IEEE Access*, vol. 9, pp. 7508–7518, 2021.

[10] Y. LeCun, Y. Bengio and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436, 2020.

[11] S. He, Z. Li, Y. Tang, Z. Liao, F. Li *et al.,* “Parameters compressing in deep learning,” *Computers Materials & Continua*, vol. 62, no. 1, pp. 321–336, 2020.

[12] W. El-Kassas, C. Salama, A. Rafea and H. Mohamed, “Automatic text classification: A comprehensive survey,” *Expert Systems Applications*, vol. 5, no. 3, pp. 123–131, 2021.

[13] M. Gomez, M. Rodríguez and C. Pérez, “Comparison of automatic methods for reducing the weighted-sum fit-front to a single solution applied to multi-text documents text classification,” *Knowledge-Based Systems*, vol. 10, no. 1, pp. 173–196, 2019.

[14] A. Widyassari, S. Rustad, G. hidik and A. Syukur, “Review of automatic language-independent text classification techniques & methods,” *Journal King Saud University of Computer Information Sciences*, vol. 19, no. 1, pp. 133–142, 2020.

[15] M. Lins, G. Silva, F. Freitas and G. Cavalcanti, “Assessing phrase scoring techniques for tokenization text classification,” *Expert Systems Applications*, vol. 40, no. 14, pp. 755–764, 2019.

[16] Y. Meena and D. Gopalani, “Efficient voting-based tokenization automatic text classification using prominent feature set,” *IETE Journal of Reasoning*, vol. 62, no. 5, pp. 581–590, 2020.

[17] A. Al-Saleh and M. Menai, “Automatic arabic text classification: A survey,” *Artificial Intelligence Review*, vol. 45, no. 2, pp. 203–234, 2020.

[18] V. Vijayan, K. Bindu and L. Parameswaran, "A comprehensive study of text classification algorithms," *Advances in Informatic*, vol. 2, no. 1, pp. 1109–1113, 2019.

[19] A. Al-Saleh and M. Menai, "Solving multi-text documents classification as an orienteering problem," *Algorithms*, vol. 11, no. 7, pp. 96–107, 2019.

[20] V. Patil, M. Krishnamoorthy, P. Oke and M. Kiruthika, "A statistical approach for text documents classification," *Knowledge-Based Systems*, vol. 9, pp. 173–196, 2020.

[21] H. Morita, T. Sakai and M. Okumura, "Query pnowball: A co-occurrence-based approach to multi-text documents classification for question answering," *Information Media Technology*, vol. 7, no. 3, pp. 1124–1129, 2021.

[22] D. Patel, S. Shah and H. Chhinkaniwala, "Fuzzy logic based multi text documents classification with improved phrase scoring and redundancy removal technique," *Expert Systems Applications*, vol. 194, pp. 187–197, 2019.

[23] M. Safaya, A. Abdullatif and D. Yuret, "ADAN-CNN for oensive speech identication in social media," *Expert Systems*, vol. 4, no. 3, pp. 167–178, 2020.

[24] W. Antoun, F. Baly and H. Hajj, "ArabADAN: Transformer-based model for arabic language understanding," *Information Technology*, vol. 4, no. 3, pp. 124–132, 2020.

[25] M. Fattah and F. Ren, "GA MR FFNN PNN and GMM based models for automatic text classification," *Computer Speech Languages*, vol. 23, no. 1, pp. 126–144, 2021.

[26] A. Qaroush, I. Farha, W. Ghanem and E. Maali, "An efficient single text documents arabic language-independent text classification using a combination of statistical and semantic features," *Knowledge-Based Systems*, vol. 7, no. 2, pp. 173–186, 2019.

[27] V. Gupta and G. Lehal, "A survey of language-independent text classification tokenization techniques," *Journal Emerging Technology Web*, vol. 2, no. 3, pp. 258–268, 2020.

[28] I. Keskes, "Discourse analysis of arabic text documents and application to automatic classification," *Algorithms*, vol. 12, no. 3, pp. 187–198, 2019.

[29] X. Cai, W. Li and R. Zhang, "Enhancing diversity and coverage of text documents summaries through subspace clustering and clustering-based optimization," *Information Science*, vol. 279, pp. 764–775, 2021.

[30] BillSum database https://www.tensorflow.org/datasets/catalog/billsum.

[31] MLSUM database: https://github.com/huggingface/datasets/tree/master/datasets/mlsum.

[32] Arabic database: https://github.com/mawdoo3.com.

[33] W. Luo, F. Zhuang, Q. He and Z. Shi, "Exploiting relevance coverage and novelty for query-focused multi-text documents classification," *Knowledge-Based Systems*, vol. 46, pp. 33–42, 2019.

[34] A. Zhou, B. Qu, H. Li and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evolutionary. Computation*, vol. 1, no. 1, pp. 32–49, 2020.

[35] M. Sanchez, M. Rodríguez and C. Pérez, "Tokenization multi-text documents text classification using a multi-objective artificial bee colony optimization approach," *Knowledge-Based Systems*, vol. 159, no. 1, pp. 1–8, 2020.

[36] P. S. Lakshmana and S. Kalpana, "UNL based document summarization based on levels of users," *International Journal of Computer Applications*, vol. 66, no. 2, pp. 167–177, March 2020.

[37] G. Lins, F. Silva, A. Freitas and G. Cavalcanti, "Assessing phrase scoring techniques for extractive text summarization," *Expert System Application*, vol. 40, no. 14, pp. 5755–5764, 2013.

[38] S. Mangairkarasi and S. Gunasundari, "Semantic based text summarization using universal networking language," *International Journal of Applied Information Systems*, vol. 3, no. 1, pp. 34–45, 2019.

[39] W. El-Kassas, C. Salama, A. Rafea and H. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Systems Application*, vol. 3, no. 1, pp. 123–132, 2021.

[40] M. Sanchez-Gomez, A. Vega-Rodríguez and J. Pérez, "Comparison of automatic methods for reducing the weighted-sum fit-front to a single solution applied to multi-text documents text summarization," *Knowledge Based Systems*, vol. 1, no. 2, pp. 123–136, 2019.