

## A Dual Attention Encoder-Decoder Text Summarization Model

Nada Ali Hakami<sup>1</sup> and Hanan Ahmed Hosni Mahmoud<sup>2,\*</sup>

<sup>1</sup>Jazan University, Computer Science Department, College of Computer Science and Information Technology, Jazan, Saudi Arabia

<sup>2</sup>Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

\*Corresponding Author: Hanan Ahmed Hosni Mahmoud. Email: hahosni@pnu.edu.sa

Received: 20 April 2022; Accepted: 12 June 2022

**Abstract:** A worthy text summarization should represent the fundamental content of the document. Recent studies on computerized text summarization tried to present solutions to this challenging problem. Attention models are employed extensively in text summarization process. Classical attention techniques are utilized to acquire the context data in the decoding phase. Nevertheless, without real and efficient feature extraction, the produced summary may diverge from the core topic. In this article, we present an encoder-decoder attention system employing dual attention mechanism. In the dual attention mechanism, the attention algorithm gathers main data from the encoder side. In the dual attention model, the system can capture and produce more rational main content. The merging of the two attention phases produces precise and rational text summaries. The enhanced attention mechanism gives high score to text repetition to increase phrase score. It also captures the relationship between phrases and the title giving them higher score. We assessed our proposed model with or without significance optimization using ablation procedure. Our model with significance optimization achieved the highest performance of 96.7% precision and the least CPU time among other models in both training and sentence extraction.

**Keywords:** Text summarization; attention model; phrase significance

### 1 Introduction

Text summarization model is one of the natural language processing main applications. The advancement in information technology has yielded a speedy upsurge in the number of accessible texts. There is a necessity to cultivate and summarize this immense amount of content and capture the key context of people attention so that they can rapidly comprehend and peruse those texts. Recent studies divided text summarization into two techniques, namely extractive and abstractive models. The extraction model produces text summary by content extraction, whereas the abstract model produces the text summary by rewording the text content.

Convolution neural networks and deep learning models steered the research to the progress of both abstractive and extractive summarization models. For instance, sequence-to-sequence extraction



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

techniques [1,2] crack the order problem of sentences in the text and in the mined summary. The impression of the attention model is merged into the methodology in [3,4], and the experiments displayed better results than the earlier non-neural models. Recently, new techniques utilized pointer attention models have been published. These techniques extended pointer attention and coverage techniques to support learning of the newest deep neural agents [5–8]. These developments have yielded a noteworthy enhancement in the ranking of the assessment metrics. Though, in the text summaries produced by such models, there is quite considerable room for advances in terms of precision and decreasing repetition.

Natural language processing is a complex situation especially for eastern languages such as Arabic language. The complexity of Arabic language is due to its derivatives and its complex morphology [9,10]. Arabic language processing faces several challenges such as:

- I. Arabic language is greatly derivational, this affects language processing task such as stemming.
- II. There is no upper case concept in Arabic language. This affects the detection of proper names and abbreviations.
- III. The absence of lexical and language processing tools.

Most of the summarization techniques were directed to the English and other Latin languages, while few research were presented to eastern languages [11–14]. Also, previous techniques used a combined parameter of two objective functions creating an impure solution of multi-objectives optimization [15–19]. In phrase relevant score systems, the objective function included essential features such as phrase position and length. Clustering-based methodology was utilized to reduce redundancy. These methodologies failed to reflect the importance of the number of clusters which greatly enhance the summarization output coverage [20,21]. In our research, we propose a language-independent extractive summarization technique that utilizes clustering optimization technique. The introduced system undergoes a series of phases to choose the phrases that produce the summarization output.

Clustering techniques group related entities in one collection, while unrelated entities in other collections. Each entity denotes a phrase, and the cluster is composed of related phrases. The dice coefficient metric is utilized as the similarity metric of two phrases, where each single phrase is denoted by using inverse text documents frequency vector [22–27].

Agglomerative clustering technique is a bottom-up process, where each single phrase is defined as a cluster. Similar clusters are then merged using a stopping condition. Partitioned clustering technique is an up-down approach and begins with a single cluster that encloses all phrases, then divides it into multiple clusters. The k-means algorithm is used in such techniques. In partitioned clustering algorithms [28–36], they choose a single phrase from similar phrases to decrease redundancy. However, it produces incomprehensible result, because it does not capture contextual data [26].

Semantic techniques, on the other hand, discover relations among different phrases. These techniques use phrases entailment and semantics relationships as well as reference and lexical [37–41]. Entailment is utilized to infer the phrase meaning from the meaning of another phrase. Phrases that are not understood from other phrases are encompassed in the summarization output. Lexical cohesion determines the essential phrases and their contributions to the summarization output using dice coefficient metric to diminish the redundancy. Semantic analysis is utilized to get the joint ones [37–39]. Datasets, such as Arabic WordNet, group synonyms into sets, and registers the various semantics duo in such sets. The authors in [39] utilized the AWN to develop queries and increase

the knowledge base of domains. Decision tree is then utilized to create the summarization output. This technique can generate a comprehensible non-redundant summary. The main problem is that constructing these resources has high time complexity.

The last techniques we are disclosing is the optimization-based ones. Multi-text documents are used in the summarization process as computational model of optimized length to yield a precise summarization output. The optimization includes maximizing coverage and diversity while minimizing redundancy, balance and coherence. Coverage ensures that all important aspects appear in the summary, while diversity decreases similar phrases presence in the summarization output. Coherence creates a coherent text course, while balance ensures the inclusion of the important aspects of the original texts [18–22].

This paper introduces the following:

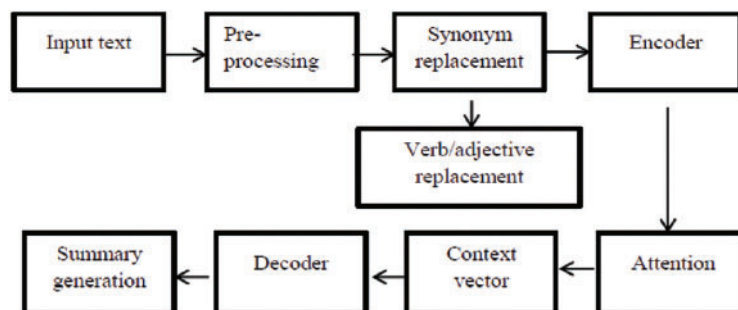
- a) Comparison of various tokenization techniques of summarization of multiple text documents.
- b) Proposing a combined objective to optimize both text-coverage and phrase significance objectives.
- c) Extensive experiments to display that our proposed technique performs better than other similar systems using precision and recall.

This paper is organized as follows: Section 2 presents the problem formulation, Section 3 proposes the global optimization technique, Section 4 displays the experimental results, while conclusion is depicted in Section 5.

## 2 Problem Formulation

The proposed system has several stages as follows (depicted in Fig. 1):

- a) Pre-processing phase to separate the input text into tokens in a process called tokenization.
- b) Description of the instructive features and definition of the sentence representation map.
- c) Feature clustering using The Interval Type-2 Fuzzy C-Means (ITTF) [20].
- d) Optimization technique to optimize the target summarization score.
- e) Evaluation and testing using the datasets BillArab, ArabS, MLAR1 and MLAR2.



**Figure 1:** The proposed encoder-decoder attention model

### 2.1 Preprocessing

Preprocessing is the course of formulating the texts documents prior to creating the text summaries. The main data is to get rid of the outsized noise embedded in a document. It is imperative to eradicate the noises that has no influence on the text.

### 2.1.1 The Document Preprocessing Stages

The document preprocessing is composed of four stages: splitting, combination, cleansing and tokenization. In this article, we employ the Tokenization Linguistic Toolkit (TLT) to implement the preprocessing process. Splitting the words and phrases is the initial stage in the preprocessing phase. In this stage, splitting of text is done using whitespace. Phrases are identified edge using punctuation marks as depicted in [Tab. 1](#).

**Table 1:** The four stages of the preprocessing

The Arabic text	و ما يبدأ حدوثه هو أنه حول الشمس الصغيرة، و النجوم الصغيرة، تمتزج كل هذه العناصر و تلتف على شكل دوامة، و تُسرّع حركتها الطاقة الصادرة عن النجم، فتشكل جسيمات و رقيقات، و تُشكل دقائقاً صغيرة من الغبار، تُشكل صخوراً و أجراماً، و في نهاية المطاف تُشكل كواكباً و أقماراً
The translation	And what starts happening is that, around young suns, young stars, all these elements combine, they swirl around, the energy of the star stirs them around, they form particles, they form snowflakes, they form little dust motes, they form rocks, they form asteroids, and eventually, they form planets and moons.
After removing noises (in this case Diacritics)	و ما يبدأ حدوثه هو أنه حول الشمس الصغيرة، و النجوم الصغيرة، تمتزج كل هذه العناصر و تلتف على شكل دوامة، و تسرع حركتها الطاقة الصادرة عن النجم، فتشكل جسيمات و رقيقات، و تشكل دقائقاً صغيرة من الغبار، تُشكل صخوراً و أجراماً، و في نهاية المطاف تُشكل كواكباً و أقماراً
Word Segmentation (from first sentence only)	<ul style="list-style-type: none"> <li>• و</li> <li>• ما</li> <li>• يبدأ</li> <li>• حدوثه</li> <li>• هو</li> <li>• أنه</li> <li>• حول</li> <li>• الشمس</li> <li>• الصغيرة</li> </ul>
Phrase splitting (first three phrases)	<ol style="list-style-type: none"> <li>1. و ما يبدأ حدوثه هو أنه حول الشمس الصغيرة،</li> <li>2. و النجوم الصغيرة،</li> <li>3. تمتزج كل هذه العناصر و تلتف على شكل دوامة،</li> </ol>

### 2.1.2 Normalization

Characters in Arabic can have a set of variants due to Arabic marks. They look in alterations and are utilized instead of some letters as they take the same looks. This can mark different representation.

Word frequency, and text similarity can be compromised. Normalization makes the text consistent by replacing the variations of a character form by a unified form, or eliminating punctuation [33]. The process is performed as follows: (i) deleting marks, (ii) deleting foreign letters, (iii) deleting long marks and (iv) transforming various forms of characters and the alike characters.

### 2.1.3 Stop-Terms Deletion

Stop-terms, such as prepositions, are repeatedly happening words in Arabic language. They are used to connect the different parts of the phrases together. Stop-terms are non-informative in recognizing text themes. They are considered insignificant in some language processing like text reduction [14]. Eliminating Stop-terms reduces the text documents length and can enhance the performance because the metrics are computed on the term frequencies in the text documents.

## 2.2 Similarity Metrics

There are various metrics to compute the similarity measure between text phrases such as dice coefficient metric, Euclidean distance, and Jacc correlation [41]. However, dice coefficient metric is the most extensively used [42].

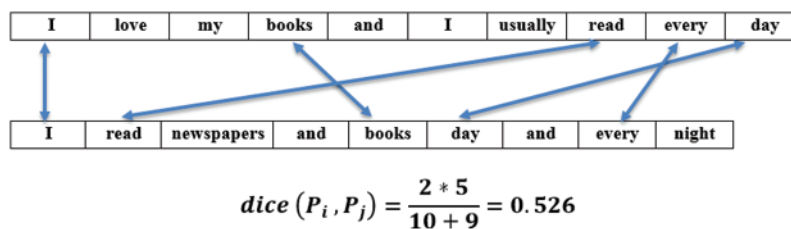
### 2.2.1 The Dice Coefficient

The dice coefficient metric is used to compute the similarity formula between two phrases  $P_i$  and  $P_j$  by computing their representation product vectors after it is divided by the number of elements in the vector. For any two phrases  $P_i$  and  $P_j$ , the dice coefficient metric (*dice*) is computed utilizing the following formula:

$$dice(P_i, P_j) = \frac{2 * P_{ij}}{|P_i| + |P_j|} \quad (1)$$

where,  $P_{ij}$  is the number of similar words in both sentences, while  $|P_i|$  is the length of the phrase  $P_i$  and  $|P_j|$  is the length of the phrase  $P_j$ .

Sentences and phrases with similar dice score (up to threshold range) will be grouped together. The group with the highest number of such sentences will be considered important and will be collected in the summary. An example of dice score between two sentences is depicted in Fig. 2.



**Figure 2:** An example of dice score between two sentences

### 2.2.2 Phrase Scoring

Following the preprocessing phase, features will be extracted for all phrases (one at a time) to calculate the score. Phrase score maximize the phrase significance as an objective. We choose four features for phrases: similarity, key phrases, location, and length. These features are extracted on both word and phrase level. Also, it is based on statistical and semantic methods. The features utilized in our proposed model are described below.

### 2.2.3 Similarity Between Phrases and the Title

Similarity depicts the overlapping among titles and phrases in the text documents. If a phrase has words that exist in the title, then it is considered a key phrase. This is founded on the theory that a researcher selects the label to reveal the topic of the text documents. Also, the phrase that shares main words with the title, will have higher rank. Therefore, we define the similarity of words with title as:

$$\text{Title summary } (P_i, t) = \text{sim}(P_i \rightarrow, t') + Kp(t) \cap Kp(P_i) \quad (2)$$

where,  $P_i$  is the ITF of a phrase  $P_i$ ,

$t'$  is the ITF of all titles,

$Kp(t)$  is the Key-words of the titles,

$Kp(P_i)$  is the Key-words of the phrase ( $P_i$ ),

$Kp(t) \cap Kp(P_i)$  is the intersection value.

This value is divided by the greatest intersection value for normalization,  $\text{sim}(P_i \rightarrow, t')$  is the similarity score between  $P_i$  and the titles  $t$  calculated by the dice coefficient metric.

### 2.2.4 Phrase Position

The Position of a phrase indicates its significance irrespective of the text documents subject. Leading phrases, particularly the first phrase, are considered key phrases and should be incorporated in the summary. This is based on the theory that states that the most significant phrases occur very early in the text documents [38]. For instance, the first phrase in a text documents is the most imperative phrase [31]. Typically, the phrase location is scored as follows [29]:

$$\text{location}(p_i) = \begin{cases} 4, & \text{first phrase in the text document} \\ 3, & \text{last phrase in the text document} \\ 2, & \text{first or last phrase in the any paragraph} \\ 1, & \text{a phrase in the first or last in any paragraph} \\ \frac{1}{\sqrt{i^2 + j^2}}, & \text{a phrase } i \text{ in any paragraph } j \end{cases}$$

### 2.2.5 Selection of Keywords

Keywords are a list of relevant words that create a compressed text of the focal subject in the associated texts. Proper noun can be a single key-word or multi key-word. The presence of Key-words in a phrase enhances its importance [5,6]. This feature score is computed by calculating their number in a phrase. Normalization is performed by splitting the feature rank by the count of keywords in all the text documents, mathematically is calculated as follows:

$$\text{Keywords}(P_i) = \frac{\text{No. of Keywords in } P_i}{\text{Total number of keywords}} \quad (3)$$

## 3 Global Optimization

Multiple text document summarization is formulated as a global optimization of multiple objective function and tests the quality of the produced summary. The proposed objectives are coverage, phrase relevance and diversity. We will maximize significance and minimize redundancy to produce more precise summary. Unfortunately, maximizing coverage and significance may yield a drop in the

diversity objective value. Therefore, an optimum solution, that can be drawn from these objectives concurrently, cannot be found. Our proposed model will run simultaneous optimization of these paradoxes. The Multiple Objective Optimization (MUL) technique appears to be the best way to handle this contradiction. MUL optimizes more than one objective to solve the problem. It offers non-dominating solutions to the problem. MUL can be defined as:

$$\text{maximize/minimize } F(x) = [f_1(x), f_2(x), \dots, f_m(x)] \text{ s.t. } x \in X \quad (4)$$

$X$  is defined as a group of objective vectors ( $x$ ) and  $m$  is the maximum count of the objectives to be optimized. MUL models were proposed to answer such challenges [42]. In our proposed model we used the Sorting Genetic non-dominated Algorithm-II (SGA-II) [14]. SGA-II is a popular multiple objective approach. SGA-II is characterized by the speedy non-dominated sorting technique, as well as speedy distance estimation technique, and the simplicity of its crowded association operator. In SGA-II algorithm, the population is partitioned into sorted fit-fronts  $F_1, F_2, \dots, F_m$ . A fit answer covers a group of suggested answers with the same aptness. The best answers will be included in the best-list of fit-fronts. The crowding distance (CD) computes the distance of two of best-list solutions. CD is used to discriminate solutions of the same fit-front. The model selects the best solutions with dissimilar non-dominate levels and better fit-near fitness. On the other hand, the one with a greater CD will be selected from the best Weighted-Sum-front. The main steps of SGA-II are: encoding of the candidates, defining initial solutions, defining the objective functions and generating the text summarization.

The main steps of SGA-II are detailed below:

- Encoding of the candidates
- In summarization, candidates define the set of phrases to produce the text summary. The proposed language-independent model utilizes binary optimization, where solutions (so called chromosomes) are denoted by binary coded vectors (BCV).

For instance let  $X_i = X_{i,1}, X_{i,2}, \dots, X_{i,N}$ , where  $X_i$  is a BCV and  $N$  is the BVC size, which is defined as the count of phrases in the related-document set. For instance, in a text of eight phrases,  $X_i = [0, 1, 1, 0, 1, 0, 0, 1]$  means that according to the solution  $X_i$ : phrases 1, 4, 6, and 7 are not contained within in the summary, while phrase 2, 3, 5 and 8 are enclosed.

### 3.1 Selection of Initial Solutions

Evolution processes usually start with initial solutions (called population),  $PP = X_1, X_2, \dots, X_g$ , where,  $g$  is the population size. The initial solution of the candidates is produced randomly from predefined ranges. The  $n^{\text{th}}$  populate of the  $g^{\text{th}}$  initial solution is a random number between 0 and 1 [15]. The transformation rule is:

$$X(g, n) = \begin{cases} 1, & \text{if } \text{rand}(g, n) \leq \tanh(X(g, n)) \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where,  $\tanh$  is the hyper-bolic tangent function [35].

And the objective function is defined as:

$$f(X) = \sum_{P_i \in \text{Summary}} \sum_{C_j \in C} \text{dice}(P_i, C_j) \quad (6)$$

where,  $C_j$  is the centroid of the document  $j$

### 3.2 Create Summarization

The global process generates several solutions. Therefore, we need to decide on the optimum solution employing maximization constraints. We use a maximization technique to obtain the solution [42]. To create such solution, we apply the maximization technique over the extracted solutions. The text summary is created by electing the phrases with the maximum Dice score and the maximum global function values as declared in Tab. 2. When the summary is overlong, we will omit the lowest Dice value phrases.

**Table 2:** The global maximization technique of several statements in various documents

Global function	S1 in D1	0.52	S2 in D1	0.72	S3 in D1	0.32	S4 in D1	0.91
	S1 in D2	0.53	S2 in D2	0.73	S3 in D2	0.33	S4 in D2	0.93
	S1 in D3	0.42	S2 in D3	0.62	S3 in D3	0.36	S4 in D3	0.82
	S1 in D4	0.53	S2 in D4	0.73	S3 in D4	0.37	S4 in D4	0.92
	S1 in D5	0.6	S2 in D5	0.63	S3 in D5	0.41	S4 in D5	0.61
Average global function number	0.52		0.686		0.358		<b>0.836</b>	

## 4 Experimental Results

The initial simulation discloses the pre-processing performance. Tokens are generated from the selected datasets using Tokenization Linguistic Toolkit (TLT). Tabs. 3 and 4 display the performance of the preprocessing phase. They show tokenization technique vs. splitting with punctuation technique. It is noticeable, from Tab. 3, that the best precision is completed when the splitting. For the Arabic language, of four different datasets, the proposed method has better Dice coefficients with tokenization and splitting. The splitting tokenization is very robust as depicted by the results.

**Table 3:** The mean dice coefficient of tokenization with splitting for four different datasets

Dataset	The mean dice coefficient of tokenization	ROUGE of tokenization with splitting
BillArab	0.819	0.979
ArabS	0.807	0.967
MLAR1	0.791	0.951
MLAR2	0.75	0.97

**Table 4:** Expected time to excerpt a sentence in the ultimate summary in seconds in the BillArab Arabic dataset

Number of phrases in thousands	Our proposed model with significance optimization	Our proposed model (without significance optimization)	Model-1 [30]	Model-2 [31]
200	0.162	0.132	0.232	0.332
400	0.321	0.291	0.391	0.491
600	0.394	0.364	0.464	0.564
800	0.443	0.413	0.513	0.613
1000	0.479	0.449	0.549	0.649

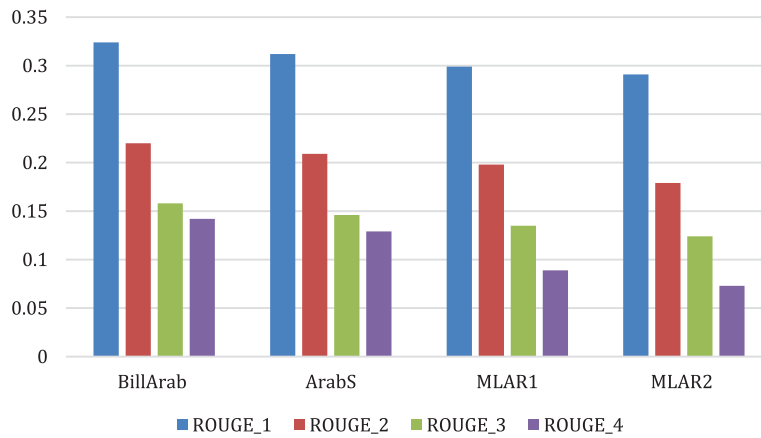
(Continued)



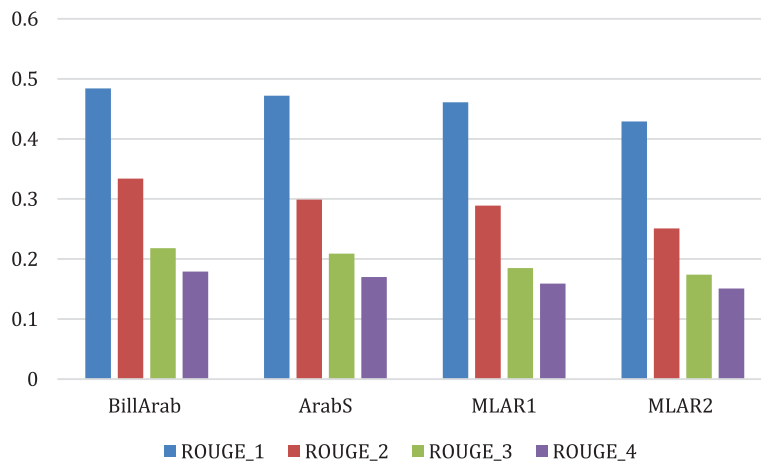
**Table 4:** Continued

Number of phrases in thousands	Our proposed model with significance optimization	Our proposed model (without significance optimization)	Model-1 [30]	Model-2 [31]
1200	0.508	0.478	0.578	0.678
1400	0.532	0.502	0.602	0.702
1600	0.553	0.523	0.623	0.723
1800	0.571	0.541	0.641	0.741
2000	0.587	0.557	0.657	0.757

The next experiment computes the F-measure of the global objective function. Fig. 3 displays the optimization function. Fig. 4 depicts the performance of optimizing the significance. The experimental results, for the BillArab dataset, optimized the competence of the experimental results significantly with an increase of 29.3%, 30.8%, 31.4% and 28.1% respectively.



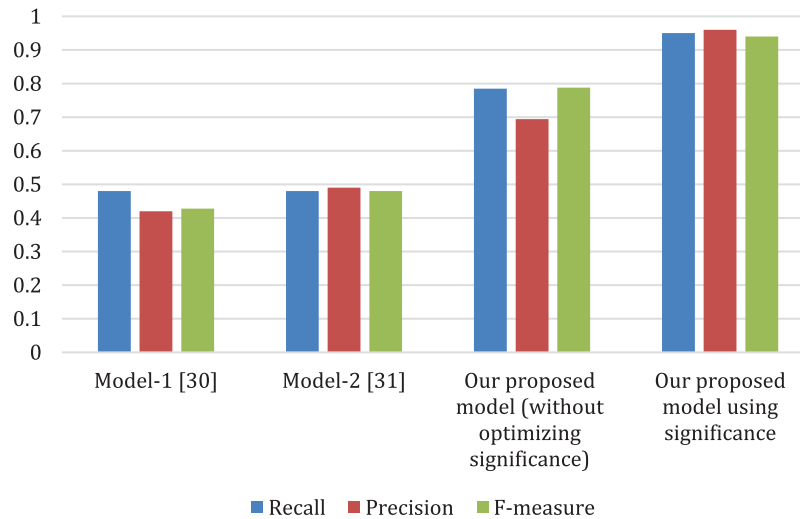
**Figure 3:** The F-measure of ROUGE\_N of optimization function



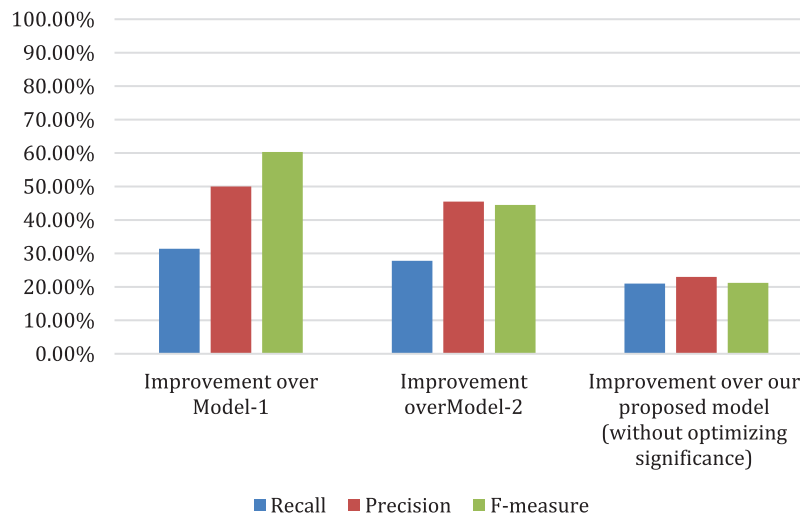
**Figure 4:** The performance of optimizing the significance objective

#### 4.1 Performance Comparison

We have done a performance comparison study between our model and other two models representing state of the art in summarization in Arabic language. The performance comparison is displayed in Fig. 5, including precision, recall and F-measure [23]. Fig. 6 displays the improvement of our model with significance optimization over Model-1 [30], Model-2 [31] and over our proposed model without significance optimization. Our model with significance optimization outperforms other models with respect to precision, recall and F-measure.



**Figure 5:** Performance comparison



**Figure 6:** Improvement of our model with significance optimization versus other model

#### 4.2 CPU Time Comparison

We performed comparison of the CPU time that is needed to excerpt one sentence in the summary. We compared our proposed model with significance optimization against Model-1 [30], Model-2 [31],

and our proposed model without significance optimization. We used the database BillArab of Arabic language, as depicted in Tab. 4. From the Table, our proposed model has less CPU time to excerpt one sentence in terms of seconds form thousands of sentences.

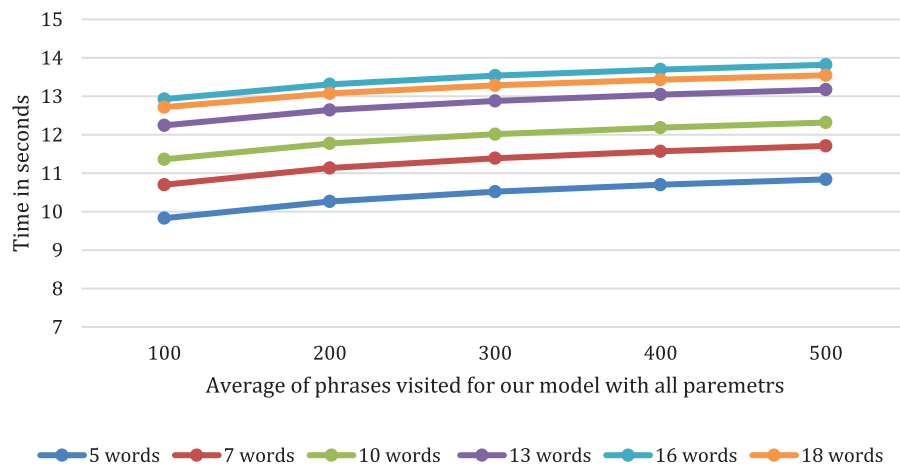
Tab. 5 depicts the average training time for different models.

**Table 5:** Average training time for different models in hours

Number of phrases in thousands	Model-1 [30]	Model-2 [31]	Our proposed model with significance optimization	Our proposed model without significance optimization
200	10.136	12.136	8.136	6.836
400	16.756	18.756	14.756	13.456
600	19.779	21.779	17.779	16.479
800	21.850	23.850	19.850	18.550
1000	23.360	25.360	21.360	20.060
1200	24.567	26.567	22.567	21.267
1400	25.571	27.571	23.571	22.271
1600	26.431	28.431	24.431	23.131
1800	27.136	29.136	25.136	23.836
2000	27.852	29.852	25.852	24.552

### 4.3 Cost of Summarization Request

The response time of a summarization request were computed as mean phrase count that summarization model has to visit. For 1,000 random summarization requests; the mean count of phrases visited per request is computed. Fig. 7 depicts the results: one each for each phrases of five words and up, to phrases of 18 words. It can be recognized that when answering a summarization request, the model will traverse only a small portion of the dataset.



**Figure 7:** Average response time of a summarization request for our proposed model

## 5 Conclusion

Computerized summarization is considered a highly compound natural language processing module, principally for the Arabic complicated language. Generating a text summarization model for Arabic language, using an encoder-decoder attention system with dual attention process. In the dual attention process, the attention model creates data from the encoder, with the dual attention, the system can excerpt and create rational content, and the fusion of the two attention phases creates precise text summaries. The enhanced attention mechanism gives high score to text repetition increasing Phrase score and also capture the relationship between phrases and the title giving them higher score. We trained the model on hundreds of Arabic texts and thousands of sentences from four Arabic datasets including BillArab [20], ArabS [21], MLAR1 [22] and MLAR2 [22]. We evaluated our model with different parameters and included significance as the most important parameter through an ablation procedure. We compared our model versus state of the art models, and it outperformed them in precision and in CPU time for training and extracting a significant sentence towards the final summary.

**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] 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] E. Ijjina and K. Chalavadi, "Human action recognition using genetic algorithms and convolutional neural networks," *Pattern Recognition*, vol. 59, no. 5, pp. 199–212, November 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. 112–119, 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] N. S. Shirwandkar and S. Kulkarni, "Extractive text summarization using deep learning," in *2018 Fourth Int. Conf. on Computing Communication Control and Automation (IC3ube)*, Pune, India, IEEE, 2018.
- [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] A. Kornilova and V. Eidelman, *BillArab Database*. NY, USA: Kaggle, 2019. [Online]. Available: <https://www.tensorflow.org/datasets/catalog/billsum>.
- [21] A. Hafez, H. Ahmed, *ArabS Database*. LA, USA: Kaggle, 2020. [Online]. Available: <https://github.com/huggingface/datasets/tree/master/datasets/mlsum>.
- [22] M. Chou, *Arabic Database MLAR1 and MLAR2*. LA, USA: Kaggle, 2020. [Online]. Available: <https://github.com/mawdoo3.com>.
- [23] 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.
- [24] H. Morita, T. Sakai and M. Okumura, "Query snowball: A co-occurrence-based approach to multi-text documents classification for question answering," *Information Media Technology*, vol. 7, no. 3, pp. 1124–1129, 2021.
- [25] 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.
- [26] M. Safaya, A. Abdullatif and D. Yuret, "ADAN-CNN for offensive speech identification in social media," *Expert Systems*, vol. 4, no. 3, pp. 167–178, 2020.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] I. Keskes, "Discourse analysis of arabic text documents and application to automatic classification," *Algorithms*, vol. 12, no. 3, pp. 187–198, 2019.
- [32] 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.

- [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] A. P. M. Pal and D. Saha, "An approach to automatic language-independent text summarization using simplified lesk algorithm and wordnet," *International Journal of Control Theory and Computer Modeling (IJCTCM)*, vol. 3, no. 4/5, pp. 15–23, 2013.
- [37] D. Radev, H. Jing, M. Stys and D. Tam, "Centroid-based summarization of multiple documents," *Information Processing and Management*, vol. 40, no. 6, pp. 919–938, 2019.
- [38] H. Silber and K. McCoy, "Efficiently computed lexical chains as an intermediate representation for automatic text summarization," *Computational Linguistics*, vol. 28, no. 4, pp. 487–496, 2019.
- [39] K. Sarkar, "Automatic single document text summarization using key concepts in documents," *Journal of Information Processing Systems*, vol. 9, no. 4, pp. 602–620, 2013.
- [40] M. Aparicio, P. Figueiredo, F. Raposo, D. Martins, M. De Matos *et al.*, "Summarization of films and documentaries based on subtitles and scripts," *Pattern Recognition Letters*, vol. 73, pp. 7–12, 2016.
- [41] S. Mangairkarasi and S. Gunasundari, "Semantic based text summarization using universal networking language," *International Journal of Applied Information Systems*, vol. 3, no. 1, pp. 120–129, August 2012.
- [42] W. Putro, G. Pratamasunu, Z. Arifin and D. Purwitasari, "Coverage diversity and coherence optimization for multi-text documents language-independent summarization," *Applied Information Systems*, vol. 8, no. 1, pp. 1–10, 2015.