



Building an Extractive Arabic Text Summarization Using a Hybrid Approach

Said Moulay Lakhdar and Mohamed Amine Chéraguï[✉] 

Mathematics and Computer Science Department,
Ahmed Draia University, Adrar, Algeria
moulaylakhdarsaid@yahoo.fr, m_cheraguï@univ-adrar.dz

Abstract. Nowadays, textual information in numerical format is hugely produced, which requires the development of tools such as Automatic Text Summarization to produce a condensed and relevant representation, thus helping the reader to decide whether the source document contains the information they are looking for or not. The Text Summarization has known these last decades an enormous progress, in particular works of Arabic language. The objective of this paper is to contribute more to these advances by proposing an Arabic text summarization tool (SumSAT), which adopts an extraction approach using hybridization between three techniques which are: Contextual exploration, indicative expression, and the graph method. Experimental results showed that the proposed approach achieved competitive and promising in the optic of generating a small and coherent summary.

Keywords: Arabic text summarization · Extractive based summarization · Contextual exploration method · Indicative expression method · Graph method

1 Introduction

Facing the exponential increase of textual resources, both on the various numerical supports and on the Internet (80% of the information that circulates on them is textual) [1], the development of tools that can manipulate this important volume such as automatic text summarization (ATS) has become crucial, since it can generate useful and relevant information by reducing the size of documents, thereby saving time and effort [2]. Radev et al. [3] defined a summary as “a text which is produced from one or more texts and conveys core information in the original texts; typically, it is no longer than half of the original text(s) and usually less than that.

In the literature, the automatic production of abstracts (or summary) is done using several methods and techniques that can be classified into two approaches, either by abstraction or extraction [4]. The first one (Abstraction approach) owes its origin to the work of van Dijk and Kintsch [5] from the fields of cognitive psycholinguistics and artificial intelligence, whose principle consists in producing the summary after comprehension, as humans do normally. This production process remains relatively difficult to compute, and text generation is still very imperfect. In the current situation, some methods use only very partial representations that reduce the original text, such as [6]: sentence reduction, sentence fusion, and sentence splitting.

However, in the extraction-based approach was essentially inspired by the approaches resulting from the information retrieval and the work of Luhn [7] and Edmundson [8]. The main purpose is to extract the most important or significant sentences in the original text and combining them to make a summary. Its objective is to produce the summary without going through deeper analysis, so the main task is to determine the relevance of these sentences according to one or more criteria (generally a statistical features) [9, 10].

The purpose of this paper is to present our Arabic text summarization tool SumSat. We adopt an extraction approach, where the originality of the work lies in making twofold contribution, the first in the pre-processing phase which consists in preparing the text for the summarization process, and the second in the processing phase where we have chosen a hybrid approach that combines three techniques: Contextual Exploration method, Indicative expression method, and Graph method.

The rest of this paper is organized as follows: Sect. 2 briefly describes related work on text summarization, especially in the Arabic language. In Sect. 3 we cover the general architecture and the methodology of SumSAT. In Sect. 4 we introduce the SumSat tool. The results of experiments on Arabic dataset are discussed in Sect. 5. The last section concludes the paper with pointers to future works.

2 Related Work

As a research topic, the text summarization is not recent; it dates back to the fifties (1950's) with Luhn's work [7], giving rise to a wide range of works and methods that can be classified into three categories: statistical (features extraction: TF/IDF, upper-case words, sentence length, similarity with the title, and sentence position in the document, etc.), linguistic (Rhetorical Structure Theory, Lexical chain, etc.) and Machine learning (Neural Network, VSM, etc.). If these works have a positive impact on the Text summarization for some languages such as English or French by outstanding achievements, works on the Arabic language are very few due mainly to its morphological and syntactic complexity. In this section, we give an overview of some works concerning the Arabic text summarization.

One of the earlier Arabic text summarization adopting an extractive methodology was LAKHAS. It was developed by Douzidia and Lapalme [11]. This system works on the journalistic text and uses several statistical features (sentence position, terms frequency, title words, and cue words). To evaluate its performance LAKHAS participated in the DUC 2004 (Document Understanding Conferences) where the result is translated into English and then evaluated using the ROUGE measure.

AlSanie et al. [12] proposed one of the first Arabic text summarization system adopting Rhetorical Structure Theory (RST) where the idea is to create all Rhetorical Structure trees (RST-trees) that describe the structural organization of the source text, based on the relationships between the text segments. For this purpose, a set of eleven Arabic rhetorical relations and twenty-five cue sentences have been used. Finally, the system produces the summary by selecting the best tree. To evaluate the performance of their system, the authors have created their own corpus (from different Sources: technical article, newspaper articles, and books, on different fields: accounting,

technology, society ... etc.) with the corresponding summaries. The System gives good results in the case of small and medium-sized documents.

Sobh et al. [13] described a classification method to generate Arabic summaries. Based on two phases: first phase: extraction of the features, 11 features were defined: Sentence Weight, Sentence Length, Sentence Absolute Position, Sentence Paragraph Position, Sentence Paragraph Length, Sentence Similarity, Number of Infinitives, Number of Verbs, umber of Identified, Number of “Marfoa’at” and Is Digit. (some of these features require the use of a POS tag) all these features will be standardized. The second phase is the classification where the authors have combined two classifiers (Bayesian classifier and Genetic Programming classifier) to extract the summary sentences. For training and evaluation, a corpus was collected from the Ahram site. To measure performance, three measures were used, precision, recall and F-measure.

El-Haj and Hammo [14] developed two Arabic text summarization systems: Arabic Query-Based Text Summarization System and Arabic Concept-Based Text Summarization System. The first one AQBTS takes a text and a query (In the Arabic language), and generates a summary for the document following the query. The second ACBTS system takes a set of keywords representing a certain concept as input to the system. Both systems adopt two methods: Vector Space Model (VSM) and the cosine similarity measure to find the most relevant passages extracted from the Arabic document to produce a text summary.

Azmi and Al-Thanyyan [15] presented a system called Ikhtasir which combined between two techniques: the first one is RST (Rhetorical Structure Theory) to build the Rhetorical Structure tree for the text and then extracts the primary summary. The second was the Sentence scoring, which is applied to determine the importance of each sentence in the text by using a score and generate the final summary whose size is set by the user based on several words, percentage of original or the number of sentences. To evaluate Ikhtasir the authors used a set of Arabic texts (Ten different sample texts collected from the Saudi’s Ar-Riyadh daily newspaper web site) and three measures were used: precision, recall and F-measure.

Belkebir and Guessoum [16] proposed a Machine Learning-based approach to Arabic text summarization based on two steps. The first one aims to build a learning model by using adaptive boosting (AdaBoost) based on a set of statistical features (the number of common words between the sentence and the title, the first or the last sentence, The number of keywords in the sentence, the number of words in a sentence). In the second step, the model which was produced was tested by identifying whether a sentence is to be included in the summary. For training, the authors created their parallel corpus (<source, summary>) composed of 20 Arabic technology news articles with the summary that corresponds to them (The summaries were manually produced). For the evaluation, they used the FI-measure metric.

Al-Radaideh and Bataineh [17] developed a hybrid text summarization approach (extraction methodology). The approach combines domain knowledge, statistical features, semantic similarity, and genetic algorithms. In this approach, genetic algorithms are used to identify the optimal sentence combination for a summary based on maximizing informative scores and cohesion between sentences. The approach was tested on two corpora: KALIMAT corpus and Essex Arabic Summaries Corpus (EASC). To evaluate the performance the authors used ROUGE and F-measure.

Recently, Al-Abdallah and Al-Taani [18] described Arabic text summarization system using three techniques which are: Informative Scores, where it calculates a score for each sentence based on Title similarity, sentence length, and Sentence location. The second method, Calculate Semantic Scores, indicates the degree of similarity between two sentences by using the cosine similarity after that a similarity matrix for a document was building to know which sentences are useful to be picked based on semantic, finally, the matrix was converted to a DAG weighted graph. The third method was a meta-heuristic search algorithm called Firefly. The algorithm starts with a random set of candidate summaries, to evaluate the quality of each candidate summary, a fitness function is defined by multiplying Semantic Scores and Informative Scores, or each sentence in the summary candidate. After several iterations when the value of the fitness function does not change. the evolution stops and the summary with the highest score will generate. For the evaluation, the EASC corpus was used and to calculate the performance the choice was the ROUGE metric to determine the accuracy.

3 The General Architecture of Our ATS SumSAT

This section introduces the general architecture of our extractive Arabic text summarization system (SumSAT), based on a hybrid approach combining three techniques which are: Contextual exploration, Indicative expression, and the graph method. Figure 1 presents the various steps to generate a small and coherent summary.

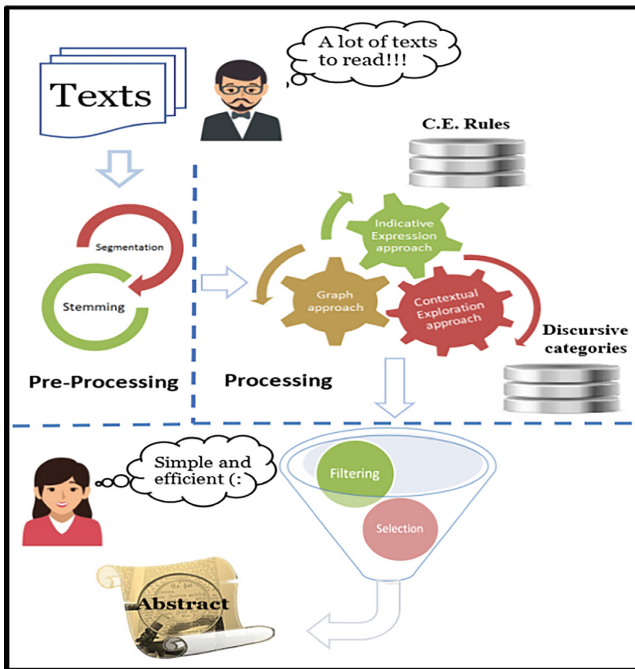


Fig. 1. The main process of our Arabic Summarization system SumSAT.

3.1 Step 1: Pre-processing

This step involves performing several basic operations to prepare the document or text for processing, including segmentation, elimination of stop words and stemming.

Segmentation is a fundamental step in automatic text processing. Its purpose is to divide a text into units of a specific type that we have previously defined and identified, in our case is the sentence. The method used to divide a text is based on the contextual exploration method, where the input is a plain text in the form of a single text segment. The segmentation starts with detecting the presence of indicators, which are punctuation marks («.», «;», «!», «?», «.»). If there is an indicator, segmentation rules will be applied to explore the contexts (before and after) to ensure that additional indicators are present and that certain conditions are met. In the case of an end of a sentence, this decision is converted into the action of segmentation of the text into two textual segments. By repeating this operation on the resultant segments, we obtain a set of textual segments which placed next to each other, which form the input plain text.

It is important to mention that in our segmentation the dot «.» cannot be always considered as an indicator of a sentence end; i.e., cases like abbreviation, acronym or a number in decimal, where particular rules can be added.

Stemming. This operation consists of transforming, eventually agglutinated or inflected word into its canonical form (stem or root) [19] (Table 1).

Table 1. Example of extracting a root from Arabic words.

Word	English Translation	Root
"مدرسة"	School	
"درس"	Lesson	درس
"مدرس"	Teacher	(D+R+Q)
"دارس"	Student	

In our case, we need the results of the stemming in the graph method to define the most important sentences. To generate these roots, we use the Full-Text Search technique, which allows us to generate the roots of words composing the sentences and eliminate the stop words. This technique also generates other features such as ranking (rank value) to classify the found sentences in order to filter the relevant ones according to their scores.

3.2 Step 2: Processing

Since we adopt an extractive methodology, the main task is to evaluate each sentence in the document to determine the importance of each them (sentence) and select the most relevant ones, to generate the most coherent and meaningful summary at the end. For this purpose, we have set up a hybrid approach combining three methods: the

contextual exploration (main method), the indicative expression and graph method (secondary methods). The secondary methods will scramble on the result of the principal method to give better results or provide a solution in the case that contextual exploration is not efficient.

Contextual Exploration Method. Allows access to the semantic content of a text, without the need for deep syntactic analyses [20]. Sentences are classified into hierarchical semantic categories (Hypothesis, Objective, Definition, etc.). This method has been chosen to produce a consistent summary and to offer users the possibility to choose the summary by point of view, where the information to be summarized is classified into discursive categories. The contextual exploration (CE) module receives a segmented text as input (the result of the segmentation module). The first task is to detect the presence of some linguistic indicators in each sentence. Once an indicator is found, all contextual exploration rules related to that indicator will be set to find additional clues and to verify the conditions required by that rule. If all conditions are verified, an annotation action, determined by the exploration rule, is performed on the sentence exactly where the linguistic indicator is placed.

For our approach, we have defined 13 discursive categories; each category has its complementary clues (See Fig. 2).

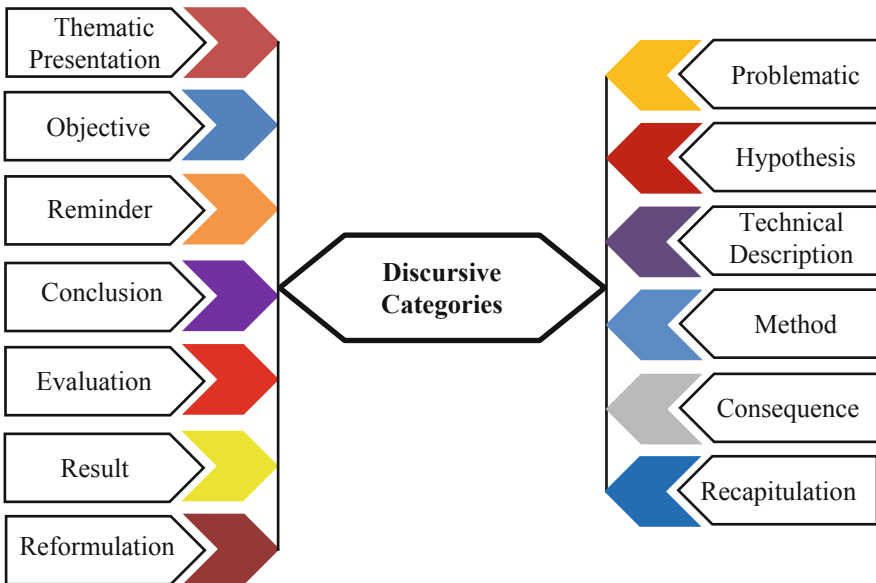


Fig. 2. The discursive categories defined for SumSAT.

Example: The following example illustrates an application of our method to select sentences that contains information about the discursive category “conclusions and results”. One of the rules associated with this category is as follows (Fig. 3):

```

<Rule NameRule="RConclusion" Task="Summary" Point_of_View = "conclusion">
  <Conditions>
    <Indicator Search_Space="sentence" Value = "form_conclusion"/>
    <clue Search_Space="." Value = "ClueConclusion" Context = "After"/>
  </Conditions>
  <Actions>
    <Annotation Annotation="Conclusion"/ >
  </Actions>

```

Fig. 3. Example of a rule describing a discursive category.

The rule, delimited by the tag (<Rule> and </Rule>), consists of two parts:

- Condition part: delimited by (<Conditions> and </Conditions>): It groups information about the indicator (delimited by <Indicator and />) associated with an information category, and information about the additional clues (<clue and/>) that are associated with it.
- Actions part: delimited by (<Actions> and </Actions>): Action to be done, after verifying the existence of additional clues and the required conditions.

Where:

- NameRule: the name that identifies the rule.
- Task: The task this rule performs since contextual exploration can be used for annotation and summary generation, as it can be used for segmentation.
- Point of View: Represents the category name of the information retrieved.
- Search_space: Space or context, where the additional clue is located; whether the search is done in the phrase itself or the paragraph.
- Value: It is the name of the file where the indicators are stored, or the name of the file where the clues are stored, associated with this category of information.
- Context: Specifies whether the search for additional clues should be done before or after the indicator.

Consider the following sentence to be annotated (applying the above-mentioned rule) (Fig. 4):

فعلى سبيل المثال، أظهرت الدراسات الحيوانية أن نبتة "روزماري" تقي من سرطان الثدي، وأن الكركم يحمي من بعض أنواع الأورام.
 For Example, **animal studies** have shown **that** rosemary protects against breast cancer and that turmeric proctects against some types of tumours.

Fig. 4. Example of a contextual exploration rule.

In this sentence, it can be said that the complementary clue (أن) is present after the indicator (أظهرت الدراسات). Therefore, the action to be taken is indicated in the actions part (delimited by <Actions> and </Action>); so, this sentence assigned the value 'Conclusion' to indicate that it contains information concerning a result or conclusion.

In some cases, the information in the form of a discursive category cannot be detected or not present in the document for summarizing. In this case the performance of the contextual exploration method will be compromised. To reduce the deficiencies of our Arabic text summarization system, we have associated with the method mentioned above (CE) two statistical methods, which are: the indicative expression and the graph method, in order to give the user the possibility to choose a default summary (general or specific field).

Indicative Expression Method. In this method, the weight of each sentence depends on some specific indicators or expressions used by the author. These indicators differ according to the field covered because the choice of text units depends on the subject matter [11]. For example, the following expressions: 'this present paper', 'in this paper we propose', 'in conclusion', can be considered relevant to a scientific topic. This method is selected to offer the possibility of generating a summary of a general order, or a specific field; sport, culture, economy, etc., by identifying sentences that contain indicators. These indicators are determined according to the field of the text to be analyzed using the following formula:

$$\text{Score}_{\text{cue}}(S) = \begin{cases} 1 & \text{if } S \text{ corresponds to an Indicator} \\ 0 & \text{else} \end{cases} \quad (1)$$

Graph Method. The generation of the summary, using the graph method, consists of selecting the most representative phrases of the source text, since it attributes to the sentences a relevance score or similarity measure by calculating the number of intersection terms [21, 22]. These terms are the result of the stemming process performed in the pre-processing process.

Suppose that we have a Document composed of six sentences (P1, P2, P2,..., P6). After applying stemming for each sentence, the total number of terms shared with all the others is given in the table below (Table 2):

Table 2. Sentences weights.

Phrases	P1	P2	P3	P4	P5	P6
Total number of Stems (Roots) shared with all other Sentences	9	8	7	3	6	5

Modelling this problem for the summary is like considering: The document as an undirected graph “ $G = (N, E)$ ”, the sentences as nodes (N_i) of this graph, the intersections of the sentences as edges (E_j) of this graph, the total number of intersecting terms (stems or roots), of a sentence with all the others, as a weight of the node representing this sentence. Finally, to generate the summary we use the Greedy algorithm (Table 3 and Fig. 5).

Table 3. Matrix for representing sentence intersections.

	P1	P2	P3	P4	P5	P6
P1	0	0	1	1	1	1
P2	0	0	1	1	0	1
P3	1	1	0	0	1	0
P4	1	1	0	0	0	1
P5	1	0	1	0	0	1
P6	1	1	0	1	1	0

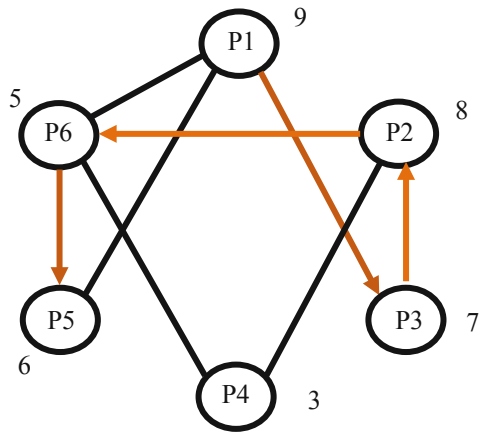


Fig. 5. Pathway followed using the Greedy algorithm.

The followed path is represented by brown arrows on the graph, and the final summary will be composed of the sentences that correspond to the visited nodes. If the final summary is limited to only four sentences: the list of selected sentences is P1, P3, P2, and P6. This list of sentences appears in the summary in the same order as the sentences appear in the source document: P1, P2, P3, and P6.

3.3 Step 3: Filtering and Selection

The generation of the summary must take into consideration the user’s requirements, and the compression ratio to determine the relevant phrases to be selected. The final summary is made up of all phrases that fulfill the following conditions:

- Sentences that belong to the discursive categories, or the selected domains (chosen by the user);
- And/or the Sentences that appear in the list of nodes visited by the graph method (the case of the default summary);
- The number of sentences is limited by the summary rate, introduced by the user;
- The appearance order of the sentences in the summary must respect the order of these sentences in the source text.

To generate a dynamic summary, a link is established between the summary sentences and their corresponding phrases in the source text.

4 Presentation of SumSat

SumSAT (Acronym of Summarization System for Arabic Text) is a web application system that runs on web browsers. Its execution is local to the IIS server (Internet Information Server), of Windows. The interaction between our system and Microsoft SQL Server is done by queries (T-SQL transactions). SumSAT is introduced to the user through a GUI, based on HTML5, ASP, C#, and Silverlight (Figs. 6 and 7).

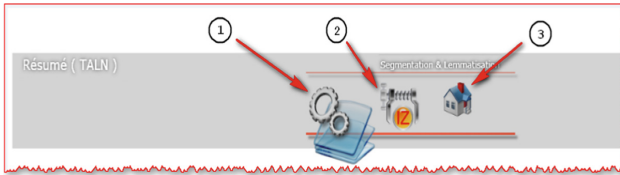


Fig. 6. GUI main menu.

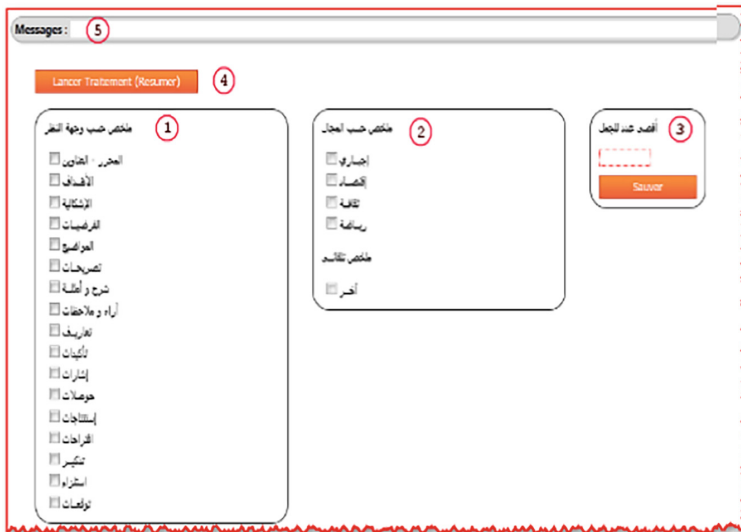


Fig. 7. GUI generation of summary.

5 Evaluation and Results

SumSAT’s summary generation is based on a hybrid approach where the discursive annotation constitutes its main task. The generated summary is based on the concept of point of view. Therefore, the relevance of a sentence depends on the presence of surface linguistic markers referring to a discursive category. The evaluation of the summary generation process consists of the evaluation of the discursive annotation task made by SumSAT.

The objective of this evaluation is to know the percentage of sentences correctly annotated by the system, compared to the total number of annotated sentences, and compared to the total number of manually annotated sentences (reference summaries). This can be expressed by measuring:

The Precision Rate: The number of correct discursive categories, detected by the system, compared to the total number of discursive categories detected by the system.

The Recall Rate: The number of correct discursive categories, detected by the system, compared to the total number of discursive categories presented in the reference summary.

The precision and recall rates are calculated as follows:

$$\text{Precision}(\%) = (a/b) * 100 \quad (2)$$

$$\text{Recall}(\%) = (a/c) * 100 \quad (3)$$

Where:

- a: Number of automatically assigned correct annotations.
- b: Number of automatically assigned annotations.
- c: Number of manually assigned correct annotations.

For this purpose, we have constructed corpora composed of twenty-five documents, and their corresponding summaries (The reference summaries are manually compiled by two experts). For each of the selected documents, we have proceeded to the generation of summaries, by discursive categories. The evaluation consists of applying the metrics, to criticize and conclude based on the results obtained.

The results of the calculated rates, as well as the precision and recall results, are illustrated in Tables 4, 5 and 6 and by representative graphs (Figs. 8, 9 and 10). These results are calculated for all the selected documents in the corpora, and each of the discursive categories adopted by SumSAT. For all categories, the precision rate is higher than 66%, except for four of them (hypothesis, Recapitulation, Reminder, Prediction), which have a precision rate between 40% and 50%. Similarly, the recall rate is higher than 66%, except for the three categories that have a recall rate between 30% and 50% (Prediction, Definition, and Reminder). This shows that SumSAT has promising results which can be improved, despite the difficulties of generating coherent summaries.

Precision rate: These results show that much more work needs to be done on refining surface markers to maximize this rate. In technical terms, it is necessary to work on two parameters. The first parameter, related to regular expressions, detects discursive markers (indicators and additional clues). The second parameter is linguistic (the good choice of these discursive markers).

Recall rate: The results show that the work which can contribute to improving these results will be linguistic, especially the collection of discursive markers to enrich linguistic resources.

It is important to mention that the obtained results are influenced by the divergence of the texts from the point of view of style, discursive and argumentative strategies, and the covered topic. This means that the surface markers, for some categories, are rarely the same from one text to another. Similarly, the indicators are sometimes weak and cannot refer to a discursive category. Moreover, the additional clues are sometimes equivocal.

Table 4. SumSat evaluation (01)

Category	Precision (%)	Recall (%)
Objective	73,68	82,35
hypothesis	42,03	70
Conclusion	77,78	70
Explanation	88,57	95,38
Consequence	77,27	70,83

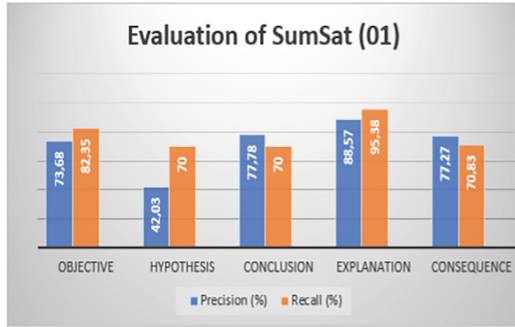


Fig. 8. Graphical representation of SumSat’s evaluation results (01).

Table 5. SumSat evaluation (02)

Category	Precision (%)	Recall (%)
Definition	66,67	32,67
Confirmation	97,5	82,98
Problematic	66,67	66,67
Reminder	50	44,02
Recapitulation	50	88,24

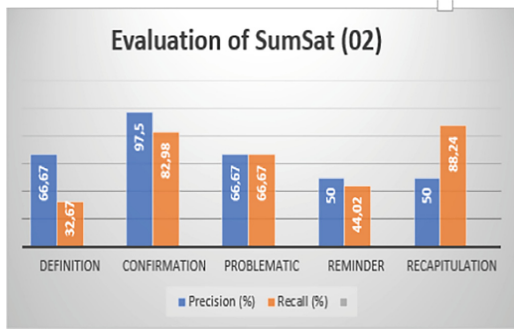


Fig. 9. Graphical representation of SumSat’s evaluation results (02).

Table 6. SumSat evaluation (03)

Category	Precision (%)	Recall (%)
Author, Title & Subtitle	91,94	91,94
Thematic	85,71	66,67
Prediction	50	50
Finding & opinion	90	69,26
Enunciation	94,94	91,85

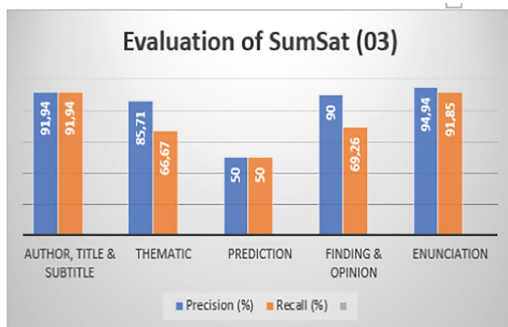


Fig. 10. Graphical representation of SumSat’s evaluation results (03).

6 Conclusion and Future Work

In this paper we have developed a hybrid Arabic text summarization system, combining two approaches: symbolic (by Contextual Exploration) and numerical (by the indicative expression method and the graph method). During the different steps of the development process, we were confronted with several problems related mainly to the nature of the Arabic language itself. In pre-processing, the incorrect use of punctuation marks (author's style) induces segmentation errors, and as a result, the relevance of phrases is incorrect, which gives an incoherent summary. The second problem is the quality of the stemming, the tool we used for this operation presented some limitations, hence the importance of choosing a performing Arabic stemmer, to ensure that the graph method gives better results.

In the processing step, one of the difficulties met, and which influences the performance of the system, is the manual search for linguistic indicators, to enrich the list of discursive categories. This task costs time and resources, which has reduced the list of the information offered by SumSAT. Also, we found that the representative sentences with a high weight may not be selected because of the restrictions on incrementing the list of visited nodes when the transition is made only between the adjacent ones (Graph model method).

As future work, we can improve the quality of the summary generated by our SumSAT system, especially in the processing step, as well as for graph method we use to generate the summary by making a modification, such that the greedy algorithm gives the advantage to the representative nodes, without being limited by the transitions between the adjacent summits. Also, the integration of a tool for identifying surface linguistic markers in documents is a good way to enrich the system's linguistic resources.

References

1. Lamsiyah, S., El Alaoui, S.O., Espinasse, B.: Résumé automatique guidé de textes: État de l'art et perspectives. In: Proceedings of the 15th Conférence en Recherche d'Information et Applications, France (2018)
2. Modaresi, P., Gross, P., Sefidrodi, S., Eckhoff, M., Conrad, S.: On (commercial) benefits of automatic text summarization systems in the news domain: a case of media monitoring and media response analysis. CoRR, [arXiv:1701.00728](https://arxiv.org/abs/1701.00728) (2017)
3. Radev, D., Hovy, D., McKeown, K.: Introduction to the special issue on summarization. *Comput. Linguist.* **28**(4), 399–408 (2002)
4. Dalaland, V., Malik, L.: A survey of extractive and abstractive text summarization techniques. In: Proceedings 6th International Conference on Emerging Trends in Engineering and Technology, pp. 109–110 (2013)
5. Kintsch, W., Van Dijk, T.A.: Toward a model of text comprehension and production. *Psychol. Rev.* **85**(5), 363–394 (1978)
6. Genest, P.E., Lapalme, G.: Fully abstractive approach to guided summarization. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Korea, pp. 354–358 (2012)

7. Luhn, H.P.: The automatic creation of literature abstracts. *IBM J. Res. Dev.* **2**(2), 159–165 (1958)
8. Edmundson, H.P.: New methods in automatic extracting. *J. Assoc. Comput. Mach.* **16**(2), 264–285 (1969)
9. Mohamed, A.A.: Automatic summarization of the Arabic documents using NMF: a preliminary study. In: *Proceedings of the 11th International Conference on Computer Engineering and Systems (ICCES)*, Egypt, pp. 235–240 (2016)
10. Oufaida, H., Noualib, O., Blache, P.: Minimum redundancy and maximum relevance for single and multi-document Arabic text summarization. *J. King Saud Univ. Comput. Inf. Sci.* **26**(4), 450–461 (2014)
11. Douzidia, F.S., Lapalme, G.: Lakhas, an Arabic summarization system. In: *Proceedings of 2004 Document Understanding Conference (DUC 2004)*, Boston, pp. 128–135 (2004)
12. AlSanie, W., Kotsis, G., Taniar, D., Bressan, S., Ibrahim, I.K., Mokhtar, S.: Towards a suitable rhetorical representation for Arabic text summarization. In: *iiWAS*, vol. 196, pp. 535–542. Austrian Computer Society (2005)
13. Sobh, I., Darwish, N., Fayek, M.: An optimized dual classification system for Arabic extractive generic text summarization. In: *Proceedings of the 7th Conference on Language Engineering*, pp. 149–154 (2006)
14. El-Haj, M., Hammo, B.: Evaluation of query-based Arabic text summarization system. In: *Proceeding of the IEEE International Conference on Natural Language Processing and Knowledge Engineering*, pp. 1–7 (2008)
15. Azmi, A., Al-thanyyan, S.: Ikhtasir a user selected compression ratio Arabic text summarization system. In: *Proceeding of International Conference of Natural Language Processing and Knowledge Engineering (NLP-KE 2009)*, pp. 1–7 (2009)
16. Belkebir, R., Guessoum, A.: A supervised approach to Arabic text summarization using AdaBoost. In: Rocha, A., Correia, A.M., Costanzo, S., Reis, L.P. (eds.) *New Contributions in Information Systems and Technologies. AISC*, vol. 353, pp. 227–236. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16486-1_23
17. Al-Abdallah, R.Z., Al-Taani, A.T.: Arabic text summarization using firefly algorithm. In: *Proceedings of the IEEE Amity International Conference on Artificial Intelligence*, Dubai, pp. 61–65 (2019)
18. Al-Radaideh, Q.A., Bataineh, D.Q.: A hybrid approach for Arabic text summarization using domain knowledge and genetic algorithms. *Cogn. Comput.* **10**, 651–669 (2018)
19. Rouibia, R., Belhadj, I., Cheragui, M.A.: JIDR: towards building hybrid Arabic stemmer. In: *Proceeding of the 1st IEEE International Conference on Mathematics and Information Technology (ICMIT)*, Algeria, pp. 183–190 (2017)
20. Alrahabi, M., Ibrahim A.H., Desclés, J.P.: Semantic annotation of reported information in Arabic. In: *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference*, Florida, pp. 263–268 (2006)
21. Al-Taani, A.T., Al-Omour, M.M.: An extractive graph-based Arabic text summarization approach. In: *Proceedings of the International Arab Conference on Information Technology*, Oman, pp. 158–163 (2014)
22. Alami, N., Meknassi, M., Ouatik, S.A., Ennahahi, N.: Arabic text summarization based on graph theory. In: *Proceedings of the 12th International Conference of Computer Systems and Applications (AICCSA)*, Marrakech, pp. 1–8 (2015)