Faculty of Engineering and Technology
Master of Computing (MCOM)

Multi-Objective Optimization with K-medoids Clustering for Arabic Multi-Document Summarization

تلخيص المقالات العربية متعددة المصادر باستخدام طرق التجميع وخوارزميات التحسين متعددة الاهداف

Author:

Rana Alqaisi

Supervised by:

Dr. Wasel Ghanem
Mr. Aziz Qaroush

This Thesis was submitted in partial fulfilment of the requirements for the Master's Degree in Computing from the Faculty of Graduate Studies at Birzeit University, Palestine

June 2019

Arabic Multi-Objective Optimization with K-mediod Clustering for Multi-Document Summarization

By

Rana Alqaisi

Approved by the thesis committee

Dr. Wasel Ghanem, Birzeit University

Mr. Aziz Qaroush, Birzeit University

Dr. Abdel Salam Sayyad, Birzeit University

Dr. Ahmad Afaneh, Birzeit University

Discussion Date

20-April-2019

Abstract

Multi-document summarization is one of the most important applications of Natural Language Processing (NLP). It aims to create a shorter version from a set of related documents with preserving the main content and overall meanings. This will eliminate redundancy and preserve the time required to read the whole documents. Text Summarization (TS) is either abstractive or extractive. In extractive summarization, the summary is generated by selecting the most important sentences based on statistical and linguistic features. In contrast, abstractive summary contains novel sentences which don't appear in the source text.

In this thesis, we propose an extractive Arabic multi-document summarization approach that employs a clustering-based method and an evolutionary multi-objective optimization method. The proposed approach uses the k-medoids clustering algorithm with a silhouette method to identify the main topics appearing in the original set of documents, while the optimization process tries to select the set of sentences to generate a summary that contains the most important sentences with maximum coverage and minimum redundancy.

The proposed system has mainly three steps: scoring the sentences, identifying the topics that appear in the documents, then multi-objective optimization. In sentence scoring, both statistical features and semantic features are used to reflect the importance of each sentence in its local document. Moreover, the k-medoids clustering algorithm with a silhouette method is used to identify the main topics appearing in the original set of documents. Finally, the evolutionary algorithm is employed to generate a summary that contains the most important sentences with maximum coverage and diversity.

The performance of the proposed system is evaluated using TAC 2011 and DUC 2002 datasets. The experimental results are compared using ROUGE evaluation measure, which shows the effectiveness of our system compared to other peer systems. With TAC 2011, our system outperforms other peer systems with all ROUGE metrics, and we achieve an F-measure of 38.9%, 17.7%, 35.4%, and 15.8% for Rouge-1, Rouge-2, Rouge-L, and Rouge-SU4 respectively. Moreover, our system with DUC 2002 dataset achieves an F-measure of 47.1%, 23.7%, 47.1%, 20.4% for Rouge-1, Rouge-2, Rouge-L, and Rouge-SU4 respectively.

الملخص

يعد تلخيص النصوص متعددة الوثائق من أهم تطبيقات معالجة اللغات الطبيعية، والذي يهدف إلى إنشاء نسخة مصغرة من مجموعة الوثائق ذات الصلة مع المحافظة على المحتوى الرئيسي وتغطية المواضيع المختلفة التي تم ذكرها. والذي بدوره يلغي التكرار ويقلل الوقت اللازم لقراءة الوثائق كاملة. تلخيص النص إما ان يكون استخراجيا أو تجريديا. في التلخيص الاستخراجي ، يتم إنشاء الملخص عن طريق اختيار الجمل الأكثر أهمية بالاعتماد على الخصائص الإحصائية واللغوية. على النقيض من ذلك ، يتكون الملخص التجريدي من جمل جديدة لا تظهر في النص الاصلي والذي يحتاج بدوره لتقنيات اللغات الطبيعية لتوليد هذه الجمل.

هذا البحث يعنى بإنشاء نظام تلخيص آلي يعتمد على تقنيات التجميع وخوارزميات التحسين متعددة الأهداف. يستخدم النهج المقترح خوارزمية التجميع k-medoids مع طريقة silhouette لتحديد الموضوعات الرئيسية التي تظهر في مجموعة النصوص الأصلية، بينما خوارزمية التحسين تستخدم لتحديد مجموعة الجمل الافضل لإنشاء ملخص يحتوي على الجمل الاهم مع الحد الأقصى من التغطية للمواضيع المذكورة، والحد الأدنى من التكرار.

يتكون النظام المقترح بشكل أساسي من ثلاث خطوات: تقييم الجمل، تحديد الموضوعات التي ظهرت في الوثائق، ثم استخدام خوازميات التحسين متعددة الأهداف. في تقييم الجمل ، تستخدم الميزات الاحصائية والسمات الدلالية لتعكس أهمية كل جملة في المقالة التي تظهر فيها. علاوة على ذلك، يتم استخدام خوارزمية التجميع k-medoids مع طريقة silhouette لتحديد الموضوعات الرئيسية التي تظهر في مجموعة المستندات الأصلية. وفي النهاية، خوازميات التحسين متعددة الأهداف تستخدم لإنشاء ملخص يحتوي على أهم الجمل وأقصى قدر من التغطية والتنوع.

تم تقييم أداء النظام المقترح باستخدام مجموعات البيانات التالية: TAC 2011 وDUC 2002. ولقد تم مقارنة النتائج التجريبية باستخدام مقياس التقييم المعروف ROUGE، ولقد اظهرت النتائج فعالية االنظام المفترح مقارنة بالانظمة المشابهة. فلقد حصل النظام على F-measure بنسبة: 38.9%, 17.7%, 35.4%, 15.8% لهذه المقاييس على التوالي: Rouge-SU4, Rouge-L, Rouge-2, Rouge-1 مع مجموعة TAC 2011. بينما حصل النظام مع بيانات DUC 2002 لنفس المقاييس على النسب التالية: 20.4%, 47.1%, 23.7%, 47.1%.

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
TS	Text Summarization
MSA	Modern Standard Arabic
EA	Evolutionary Algorithm
HMM	Hidden Markov Mode
CLASSY	Clustering, Linguistics, and Statistics for Summarization Yield
RST	Rhetorical Structure Theory
CST	Cross-document Structure Theory
SDRT	Segmented Discourse Representation Theory
VSM	Vector Space Model
GA	Genetic Algorithm
MR	Mathematical Regression
FFNN	Feed Forward Neural Network
PNN	Probabilistic Neural Network
GMM	Gaussian Mixture Model
MDSCSA	Multi Document Summarization using Cuckoo Search Approach
AWN	Arabic Word Net
LSA	Latent Semantic Analysis
POS	Part-of-Speach
NER	Named Entity Recognition

TF-ISF	Term frequency-Inverse Sentence Frequency
TF-IDF	Term frequency-Inverse Document Frequency
MOEAs	Multi-Objective Evolutionary Algorithms
PSO	Particle Swarm Optimization
NSGAII	Non-dominated Sorting Genetic Algorithm-II
IBEA	Indicator Based Evolutionary Algorithm
SPEA2	Strength Pareto Evolutionary Algorithm2
MOCell	Multi Objective Cellular algorithm
MT	Machine Translation
HITs	Human Intelligence Tasks
hLDA	hierarchical Latent Dirichlet Allocation
AQBTSS	Query-Based Text Summarization System
ACBTSS	Arabic Concept-Based Text Summarization System
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
LSA	Longest Common Sequence
jMetal	Metaheuristic Algorithms in Java
DUC2002	Document Understanding Conference 2002
TAC2011	Text Analysis Conference 2011

Chapter 1 Introduction

1.1 Area of Study

The massive increase of information on the Internet has complicated the task of extracting useful information, e.g. the enormous number of articles which are daily posted to news websites. Each website covers the story from its preference and view, so users need to read all documents to cover the different aspects of a specific event or story. Moreover, online forums and social networks have become the most popular platform for users to share their experience with hotels, restaurants, and movies. For example, Tripadvisor.com is a website used to share reviews about hotels and restaurants, and identifying the helpful information within a reasonable time has become a tedious task [1]. Therefore, automated summarization systems that can help the user to identify the most important information within a short time has become a necessity. Automatic summarization systems have been applied for different domains including search engines, web pages, news, and all forms of online reviews.

Text Summarization (TS) is one of the most important applications of Natural Language Processing (NLP). TS aims to create a short version from one or more related text documents while preserving the content and overall meanings. Summarization methods can be classified based on different properties, features, and parameters such as input, output, language, and generality as shown in Figure 1.1 [2].

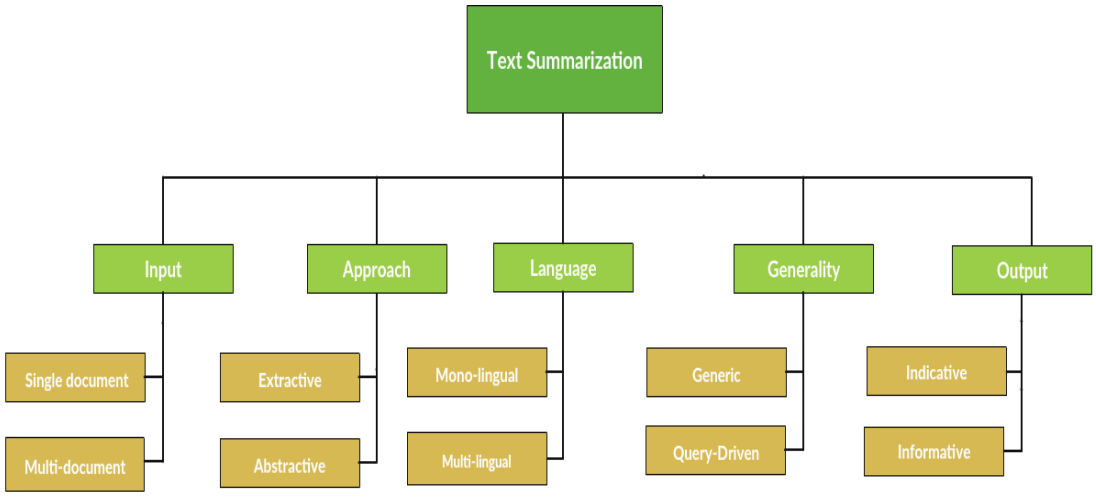

Figure 1.1: Categories of Text Summarization.

Accordingly, summarization system is classified based on the input into single document and multi-document summarization. As shown in Figure 1.2, single-document summarization tries to summarize a single document, while a set of related documents from different sources is processed in multi-document summarization. Therefore, single-document does not exhibit inconsistency problems, because it has only one author or group of authors who wrote it according to a common consensus. However, a set of problems in multi-document summarization is raised such as inconsistency, redundancy, and conflicting ideas by the different authors. As a result, dealing with multi-document summarization is harder than single-document [1].

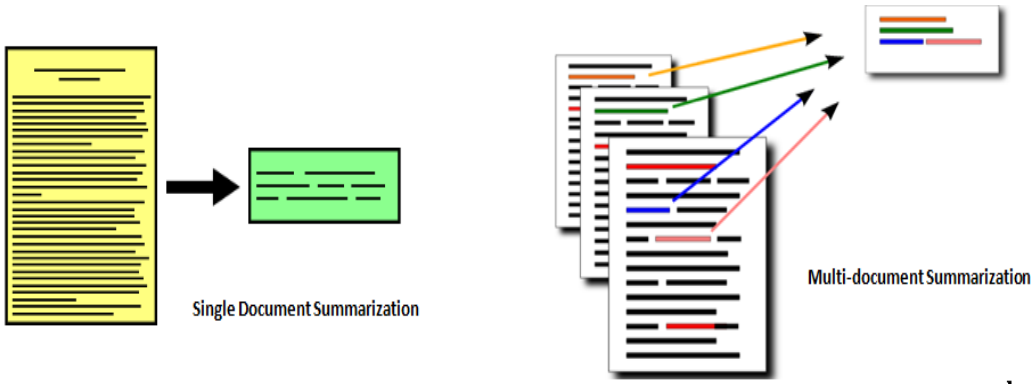

Figure 1.2: Single vs. Multi-document Summarization.

In addition, the generated summary may be extractive or abstractive. In extractive summarization, the summary is formed by selecting the important sentences based on statistical and linguistic features, then combining them. In contrast, the abstractive summary depends on understanding the text using the natural language processing techniques to generate novel sentences that hold the main ideas appeared in the source text. Despite that abstractive summaries are more readable and similar to human summaries, it needs deep knowledge of the text and lexical resources such as parsers and language generators. Therefore, researches focus mainly on extractive text summarization [3, 4].

Moreover, summarization system is classified as monolingual and multilingual based on the language of input documents. With monolingual summarization system, all documents have the same language, while in multilingual different languages can be seen in the input documents and the output summaries. Also, the summary may be generic and covers all topics in documents, or query-driven where it is generated based on the user's query. Finally, the output is an important parameter in classifying the summarization system into informative or indicative. Informative summaries cover the content of all topics appeared in the input documents, while indicative summaries represent a description of the input documents and include only meta-data [5].

1.2 Arabic Natural Language Processing

1.2.1 Arabic Language

The Arabic language is the largest Semitic language in terms of speakers. Arabic is spoken by more than 400 million people worldwide, and it is an official language of 22 countries. Moreover, Arabic language is considered one of the fastest-growing languages on the web [6].
Arabic language is written from right to left and contains 28 letters, and it has short vowels which appear as diacritical marks. Moreover, it has three different forms: classical Arabic, Modern Standard Arabic (MSA), and dialectal Arabic. The classical form is spoken daily by the prayers, while MSA which is derived from classical Arabic is seen in news media and formal speech. On the other hand, dialectal Arabic is spoken by ordinary people and has no written standards. Therefore, most of the work with Arabic NLP has been focused on MSA [5, 6].

1.2.2 Challenges of Arabic Natural Language Processing

Arabic Natural Language Processing (NLP) is considered much more complex than English language and other European languages. The main reason for this complexity is the nature of Arabic language which is highly derivational and has rich morphology. Therefore, Arabic NLP has many challenges that prevent the advance of research compared to other languages, which include the following [5, 6]:

- Arabic language is highly derivational and inflectional, this highly affects NLP task such as stemming and lemmatization. The Arabic language has roughly 10,000 roots and around 120 patterns for affixes.

- The absence of directs in written documents, where diacritics play an important rule in determining the word meaning and ease the task of tokenization and parsing the text. For example, the word (درس) can be read as (دَرَسَ) which is the past tense of the verb 'study', (دَرّسَ) which is the past tense of the verb 'teach', or (دَرْس) which is a noun that means 'lesson'.

- No capitalization in Arabic language which hardens the identification of proper nouns, titles, and abbreviations. This highly affects the task of named-entity recognition.
- The lack of resources such as lexicons and NLP tools.

1.3 Problem Statement

The big challenge in multi-document summarization is to extract the most important sentences that cover the main topics of the original source text while eliminating redundant information in the generated summary. Therefore, our research problem is summarized by the following questions:

1. To what extent the preprocessing can affect the summarization process?
2. How to cover the main topics mentioned in the original source text?
3. How to eliminate the redundancy in the generated summary?
4. How to handle with multi-document summarization problem as multi-objective optimization?

1.4 Contributions

In this research, an approach of three stages is proposed to produce the output summary. At first, a score is assigned to each sentence to express how is important. Then, a clustering technique called k-medoids was used to extract the main topics. Moreover, the silhouette method is used with k-medoids to determine the optimal number of topics. Finally, we formalize the multi-document summarization as a multi-objective optimization problem with a clear and accurate definition of three objective functions: coverage, sentence score, and diversity. This step is considered as an added value in Arabic multi-document summarization since all Arabic related approaches deal with redundancy and coverage as a single score represented as a weighted sum of these objectives. Moreover, using the score of sentences as an objective will spur sentences

that are important and not similar to other sentences to appear in the output summary.
We summarize our contributions in this thesis as follows:

1. We have studied the effect of using different tokenization and stemming methods in multi-document text summarization.

2. Our approach has been tested with the Arabic language only, but it is language independent. The approach perfectly works by changing the used tools during the feature extraction to the desired language.

3. Employing a combination of the most important statistical and semantic features with novel representation to score each sentence.

4. The usage of k-medoids to extract the main topics with the silhouette method to determine the optimal number of clusters.

5. Up to our knowledge, our approach will be the first one that handles the Arabic multi-document summarization as a multi-objective optimization problem.

6. Our method has been evaluated and tested on two well-known published datasets, and the evaluation results show that our approach outperforms other peer systems.

1.5 Organisation of Thesis

The next chapters are organized as follows: Chapter 2 discusses the multi-objective optimization approaches, text summarization approaches, and features. Chapter 3 presents the proposed methodology, documents preprocessing, document representation, features extraction, and objective definitions and formalization. In chapter 4, data sets, evaluation measures, tools, and suggested experiments are illustrated. Finally, chapter 5 concludes the work and presents the current progress and future work.

Chapter 2 Background and Related Work

The first section of this chapter discusses the general idea of the optimization approaches, while the rest of the chapter illustrates the related work of text summarization approaches and used features in this domain.

2.1 Multi-Objective Optimization Approaches

Any optimization problem is formulated as follows: $maximize \;\; f(x) = f_1(x), f_2(x), ..., f_k(x)$, such that: $g_i(x) < b_i$ and $i = 1, 2, ..., p$, where x is a vector of decision variables, f_i's are the objective functions, and g_i represents the constraint functions of the problem. There are two main approaches to multi-objective optimization problem: the first one is the weighted sum approach which combines the objective functions into a composite function. This approach treats the multi-objective optimization problem as a single objective optimization problem where the composite function is the only objective to be optimized. This method is simple and can reach an optimum solution. However, when the problem has a set of optimal solutions an evolutionary algorithm (EA) is needed [7, 8].

The second approach to multi-objective optimization is the evolutionary algorithm. It is used to find a set of optimal solutions, which is called the pareto optimal set. This set represents solutions that are non-dominated by each other. For example, the decision of buying a car has two main contradictory objective functions which are cost and comfort. It is clear from Figure 2.1 that it is impossible to improve one without making the another worse. Thus, a set of optimal solutions exists at different preferences of the decision maker. In real life problems, an evolutionary algorithm, which retrieves a set of optimal solutions, is preferred more than

the weighted sum approach, because the final solution is a trade-off between the conflicting objectives [7].

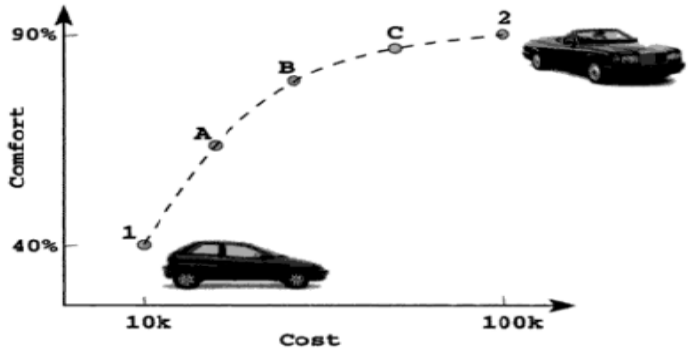

Figure 2.1: The pareto front of buying a car with cost and comfort objectives.

2.2 Text Summarization Approaches

Since the middle of the 20th century, the automatic text summarization problem has been flourishing so that we can see a variety in the available approaches. Each approach has its own advantages and limitations which are discussed in this section.

2.2.1 Statistical based Approach

Statistical methods including the Hidden Markov Model (HMM), K-mixture Model, Expectation Maximisation, and Vector Space Model are used by many summarization systems to extract the relevant sentences [9]. Alguliev and Aliguliyev [10] have proposed a system that ranks the sentences based on similarity with document's title, then sentences that have a score higher than a predefined threshold are included in the summary. Also, statistical models can be used to eliminate irrelevant sentences or words before the summarization process. For example, Schlesinger et al. [11] have used the Hidden Markov Model as an elimination technique. HMM contained two states corresponding to summary and non summary sentences, and they used a naive Bayesian approach to test the probability of a sentence to be included in the summary or not. Moreover, Morita et al. [12] have introduced a system called 'query-snowball' to generate a query-based summary from a set of related documents. They have formulated the problem as

a maximization problem that maximizes the total score of words included in the summary. Statistical methods can be used for single and multi-document summarization. Also, it can be used to enhance the selection of important sentences or the elimination of redundant sentences. However, it fails to understand the text since it only depends on statistical measures [9].

2.2.2 Graph based Approach

In this approach, each document is represented as directed graph G=(V, E), where V represents the set of vertices, and E is the edge between two vertices. Each sentence of the document is a node (vertex) in the graph, and an edge connects two sentences if there is a relation between them. The weight of the edge corresponds to the similarity between two sentences. The cosine similarity is widely used to measure the relation between two sentences, and an edge exists between two nodes if their similarity is greater than a predefined threshold [13, 14].

This approach has been adopted by many researchers. For instance, Radev et al. [15] have used this approach and generated the summary by selecting sentences that are strongly connected to many other sentences. Also, another approach is proposed by Al-Tanni and Al-Omour [16], where the document is represented as a weighted directed graph, and the cosine similarity with TF-IDF feature is used to rank the sentences. The summary is generated by selecting the sentences with the shortest path from the first node to the last node in the graph.

The document's sub-graphs represent the different topics covered in the document. This property can be helpful for both query-based and generic-based summaries. For query-based summaries, sentences are only connected from pertinent sub-graph, while for generic summaries sentences are selected from each sub-graph for best coverage [2]. However, the graph-based approach fails to understand the text since it depends only on statistical measures. Thus, the generated summary may be incomplete. This approach can be used for both single and multi-document summarization.

2.2.3 Cluster-based Approach

This approach is used to group similar objects in one cluster, while dissimilar ones into different clusters. Each object represents a sentence, and the cluster is a set of related sentences. The

cosine similarity is widely used to measure the similarity between two sentences, where each sentence is represented as TF-IDF vector [13, 14]. Clustering approaches can be classified as agglomerative, and partitional based on the initial state. Agglomerative clustering is a bottom-up approach and it represents each sentence as a cluster then tries to merge similar clusters until stopping criteria. On the other hand, partitional clustering starts with one cluster that contains all sentences, then tries to divide it into different clusters. The k-means is considered the common partitional clustering algorithm [14].

Clustering approach has been widely used for Arabic multi-document summarization. For instance, Fejer and Omar [17] have proposed a combined clustering method to group the documents into clusters. Then the key-phrases are extracted from each cluster so that this will help in identifying the most important sentences. Haboush et al. [18] have proposed a system that uses the word roots as a weight instead of the word itself. The number of sentences that has the same root will be in the same cluster. Therefore, it is possible to determine the weight of each word in the cluster. Then the system calculates the score of each sentence, and top-ranked sentences are chosen to form the summary.

In addition, Judith et al. [19] have proposed a system called CLASSY (Clustering, Linguistics, and Statistics for Summarization Yield) to generate single and multi-document summaries of machine translated documents. The system has five steps: preparation of raw texts, trimming of sentences, scoring, redundancy elimination, and sentence organizing. The results are not satisfactory as the English language, since trimming of sentences based on linguistic, and the translated text highly depends on the quality of the translation machine.

On the other hand, Radev et al. [20] have proposed a system called MEAD. The system is a centroid-based method to score sentences based on sentence's features including sentence position, sentence similarity with the centroids, and sentence length. Sarkar [21] has presented a system for multi-document summarization. The system clusters sentences using the histogram similarity. Then it uses clustering algorithms to identify the sub-topics from the input set of related documents, and finally selects the representative sentences from clusters to form the summary. He proposed two approaches to order the clusters based on the size or the importance of the clusters. Cluster importance is determined by counting the important words in each clus-

ter after removing the stop words, ordering the clusters, then selecting the top-n clusters based on their importance, and finally selecting one sentence from each cluster to generate the final summary. Indeed, the representative sentences are determined by many criteria: randomly, long sentences are preferred, based on similarity to the centroid of input documents, or using the local and global importance. The results show that ordering clusters based on the importance and selecting sentences based on local and global importance outperform other approaches.

Systems that use clustering have many issues that affect the quality of the generated summary including the number of clusters, how to order them, how to select sentences, and finally how to merge the selected sentences to form the summary. However, little researches consider all these issues together. The authors in [17] have proposed to rank sentences based on the key phrases. One sentence from similar sentences which has the highest score is included in the summary, while the authors in [22] selected the top 5 sentences from each cluster based on its TF-IDF score. On the other hand, a system [23] has been proposed that has exactly two clusters, and clusters are ranked based on the distance between its objects, and sentences from the biggest clusters are chosen. Another system has been implemented [21] where the number of clusters relies on the size of the input collection, and clusters are ordered by counting the words in each cluster. Moreover, this system ranks the sentences based on local and global importance of a sentence, then one representative sentence from each cluster is selected to form the summary. Moreover, El-Ghannam and El-Shishtawy have used the key phrases to score sentences and documents, and they have proposed two techniques Sen-Rich that selects the maximum richness sentences, and Doc-Rich that selects sentences from centroid document [17].

This approach is widely used in multi-document text summarization since similar sentences from different documents are grouped into the same cluster. Thus, the selection will be one sentence from many similar ones, as a result, this will reduce the redundancy. However, it generates an unreadable summary, since it is based on statistical measures and cannot capture contextual information [14].

2.2.4 Discourse Theory

Discourse theory is represented by a set of approaches to produce more informative and representative summaries by describing the relations between text units. These approaches include Rhetorical Structure Theory (RST), Cross-document Structure Theory (CST), and Segmented Discourse Representation Theory (SDRT). RST describes the main aspects of the text and the relations between sentences. It represents the coherent text as a tree of a nuclear node which represents an important proposition, and satellite which is considered as additional information. On the other hand, CST describes the semantic connection among units of related texts. It is widely used in multi-document summarization, and it represents the coherent text as a graph. Also, SDRT allows attachment between non-adjacent discourse units and for multiple attachments to a given discourse unit, and it represents the discourse structures as an acyclic graph [24]. Cardoso and Pardo [24] have proposed an automatic summarization system based on two semantic discourse models: Rhetorical Structure Theory (RST) and Cross-document Structure Theory (CST). They have shown that multi-document summarization process is enhanced and the generated summary is more informative when they combined the RST and CST models.

Similarly, Alsanie [25] has proposed an Arabic text summarization system based on Rhetorical Structure Theory (RST). The system determines the elementary units and defines the rhetorical relations between the extracted units, then it generates all possible RS-trees. Finally, the system selects the most suitable RS-tree. The RS-tree presents important information at high levels (near) the root. Thus, it is possible to generate summaries with various granularities. The first level for short summaries, whereas longer summaries are constructed by considering more levels e.g. two levels. The system has reported good results for small and medium-sized articles.

Azmi and Al-Thanyyan [26] have proposed a two-pass system. It generates the summary using the RST and scores each sentence in the generated summary. Then, the final summary is formed by selecting sentences that maximize the overall score. The algorithm for the rhetorical parsing is based on Alsanie's work [25], while important sentences are determined by their positions, similarity to the title, and if it contains numerical values. The system combines the advantages of RST-based systems and the different scoring schemas.

Furthermore, Mathkour [27] has used the rhetorical structure theory to classify Arabic security documents. The proposed model parses the paragraphs from the documents to form the RST tree. The importance of each paragraph is determined by examining the promotion of the tree root. Also, Ibrahim et al. [28] have proposed a hybrid model that uses the RST to discover the most important paragraphs, and vector space model (VSM) to rank the paragraphs based on the cosine similarity.

In addition, the SDRT model has adopted for the first time by Keskes [24]. He has proposed a semantically driven approach that uses the SDRT model to analyze the Arabic discourse. The document is represented by a acyclic graph, which addresses explicit and implicit Arabic discourse relations. Discourse relations allow the linking of adjacent as well as non-adjacent units within the SDRT framework. This approach is based on the linguistics process so that the generated summary is cohesive. However, this approach fails to deal with multi-document issues such as redundancy elimination.

2.2.5 Machine Learning based Approach

In this approach, text summarization is considered as a binary classification problem, where a set of documents and their extractive summaries are used as a training set, and each sentence is classified as a summary sentence or non-summary based on statistical features [2].

Abdel Fattah et al. [29] have investigated the effect of different features combinations in text summarization models. Ten features have been used to train the genetic algorithm (GA) and the mathematical regression (MR) models to obtain a suitable combination of feature weights. Then, all features are used to train feed forward neural network (FFNN), probabilistic neural network (PNN) and Gaussian mixture model (GMM) to form the summary for each model. Their results indicated that GMM outperforms other models. Also, Belkebir and Guessoum have proposed [30] a supervised approach using Adaboost with a set of statistical features which are the sentence length, sentence position, similarity with the title, and the number of key phrases. They compare their approach with the multi-layer perceptron and j48 decision tree. Results indicate that the model outperforms the multilayer perceptron and j48 decision trees.

Recently, Al-Radaideh and Bataineh [31] have proposed a hybrid approach that uses the domain knowledge along with statistical features and genetic algorithm to select the best summary for Arabic single documents. They have manually created the knowledge for Arabic political domain by determining the list of keywords and phrases related to that domain. They have evaluated their approach over KALIMAT corpus and Essex Arabic Summaries Corpus. The results are promising with political documents summaries.

According to [32], machine learning approaches are well suited for single document more than multi-document summarization. Moreover, experiments have shown that unsupervised methods outperform supervised methods for multi-document summarization since it is based on a single feature such as the presence of topic words or graph methods.

2.2.6 Lexical and Semantic based Approaches

The aim of these approaches is to find relations between sentences. Many techniques exist in the state of art, including textual entailment, semantic clustering, co-reference, and lexical chains and semantic. Text entailment has used to determine if a sentence can infer the meaning of another one. Therefore, only sentences that are not inferred by any other sentences are included in the summary.

Little researches have been done in this area for Arabic text summarization. The authors in [33] have used the lexical cohesion to determine the important sentences and ignore the others. Then, the cosine similarity is used to reduce the redundancy. Also, the root and semantic relations between senses of words are used in [34] to extract the common words. These approaches are good, but as mentioned above, Arabic NLP resources are limited. Therefore, the detection of semantic and lexicon relations becomes harder. Also, ontologies are used to capture the semantic information of a specific domain. Arabic Word Net (AWN) is a form of ontologies, that groups synonym words into sets, and records the different semantic relations into these sets. Moreover, the authors in [35] use the AWN to expand the user's query and adding the knowledge base of a specific domain, then the decision tree algorithm is used to generate the summary.

A recent approach is proposed by Al-Sabahi et al. [36] that uses linear algebra along with

semantic features of the text to overcome the limitations of Latent Semantic Analysis (LSA) approaches. They have reduced the dimension of the LSA by utilizing the part of speech (PSO). Also, they have added the weight of four adjacent sentences to consider the word order and syntactic relations, while calculating the input matrix. Indeed, they have proposed a new LSA method for selecting the sentences, where the term description is appended to the sentence description for each topic.

This approach greatly works when an ontology is available; however, these ontologies are not available for all domains, and constructing them manually is a time-consuming task.

2.2.7 Multi-Objective Optimization Approach

Multi-document summarization is considered by many researchers as a multi-objective optimization problem, where a set of objectives are considered to produce a good summary, including maximum coverage, minimum redundancy (maximum diversity), coherence, and balance. Coverage means that a summary should contain all important aspects that appear in the documents, while diversity aims to reduce the similar sentences in the output summary. On the other hand, coherence aims to generate a coherent text flow. Moreover, balance means that a summary should have the same relative importance of different aspects in the original documents [37, 38, 39, 40, 41].

This approach has been adopted by many researchers with different algorithms. Also, they consider the different number of objectives with expressing the objective functions in different forms. Recent studies use clustering techniques to minimize the redundancy, while the coherence has been solved by many approaches such as ordering sentences based on key phrases [42], measuring the mutual information between the adjacent terms or sentences [37, 43], and transforming synonymous words into basic words to improve the readability [44]. Also, Oufaida et al. [45] have used clustering techniques and discriminant analysis method (minimum redundancy and maximum relevance) to rank terms based on its discriminant power, and top sentences are selected to form the summary.

Recently, multi-objective evolutionary algorithms (MOEAs) have attracted a lot of researches by their ability to approximate a set of Pareto solutions (non-dominated solutions) [46]. Any

optimization problem is formulated as shown in the following equation:

$$maximize \quad f(x) = f_1(x), f_2(x), ..., f_k(x) \tag{2.1}$$

subject to:

$$g_i(x) < b_i, i = 1, 2, ..., p \tag{2.2}$$

Where x is a vector of decision variables (free variables), f_i is the ith objective functions, and g_i represents the ith constraint function of the problem.

Multi-objective evolutionary algorithms are widely used in multi-document summarization. The authors in [47] have used the genetic algorithms and swarm intelligence to minimize the divergence between the probability distributions of n-grams in the source documents and the summary. However, the researchers have employed the Branch-and-bound algorithm, Particle Swarm Optimization(PSO) [48, 49], binary differential evolution algorithm [50], and Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [51] to maximize the coverage and minimize the redundancy, while the length of the summary is a constraint. Moreover, multi-objective evolutionary algorithm with Tchebycheff decomposition is used to maximize the coverage and diversity [52]. Whereas, Multi-Objective Artificial Bee Colony has been adopted by the authors in [53] to minimize the redundancy with including only relevant text units in the summary. In addition, the authors in [54] maximize coverage, cohesion, and readability using the Cuckoo search approach (MDSCSA). A recent approach has been proposed by Al-saleh and Menai which uses the Ant Colony optimization algorithm to maximize the summary coverage. Their approach has been evaluated using the 2011 MultiLing Pilot dataset for both Arabic and English language [55]. Equations that describe the objective functions proposed by each system are shown in Table 2.1.

Table 2.1: Different objective functions proposed in the state-of-art.

Paper #	Objective Functions	Equations
[47]	Divergence	$KL(P\|\|Q) = \sum_{g} p_{\mathrm{P(w)}} \log_2 \frac{p_{\mathrm{P(w)}}}{p_{\mathrm{Q(w)}}}$
[48]	Relevance, Redundancy	$\text{maximize} \quad f_{\mathrm{a}} = a \cdot f_{\mathrm{cos}} + (1-a) \cdot f_{\mathrm{NGD}}$ $f_{\mathrm{cos}} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} [sim_{\mathrm{cos}}(\vec{D}, \vec{si}) + sim_{\mathrm{cos}}(\vec{D}, \vec{sj}) - sim_{\mathrm{cos}}(\vec{si}, \vec{sj})] x_{\mathrm{ij}}$ $f_{\mathrm{NGD}} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} [sim_{\mathrm{NGD}}(\vec{D}, \vec{si}) + sim_{\mathrm{NGD}}(\vec{D}, \vec{sj}) - sim_{\mathrm{NGD}}(\vec{si}, \vec{sj})] x_{\mathrm{ij}}$
[50]	Coverage, Redundancy	$\text{maximize} \quad f(X) = w \cdot f_{\mathrm{cov}}(X) + (1-w) \cdot f_{\mathrm{red}}(X)$ $f_{\mathrm{cov}} = \sum_{i=1}^{n-1} sim(si, O) x_{\mathrm{i}}$ $f_{\mathrm{red}} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} [1 - sim(si, sj)] x_{\mathrm{i}} x_{\mathrm{j}}$
[51]	Coverage, Diversity	$f_{\mathrm{coverage}}(X) = G \cdot \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} sim(s_{\mathrm{i}}, s_{\mathrm{j}}) x_{\mathrm{i}} x_{\mathrm{j}}$ $f_{\mathrm{diversity}}(X) = G \cdot \sum_{i=1}^{k} \sum_{j=1}^{n} sim(c_{\mathrm{i}}, s_{\mathrm{j}}) x_{\mathrm{i}}$
[53]	Relevance, Redundancy	$f(X) = max(f_{\mathrm{red}}(X), f_{\mathrm{cov}}(X))$ $f_{\mathrm{red}} = \frac{1}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} sim(s_{\mathrm{i}}, s_{\mathrm{j}}) \cdot Y_{\mathrm{ij}} \cdot \sum_{i=1}^{n} x_{\mathrm{i}}}$ $f_{\mathrm{cov}} = \sum_{i=1}^{n} sim(s_{\mathrm{i}}, o) x_{\mathrm{i}}$
[54]	Coverage, Cohesion, Readability	$f(S) = f_{\mathrm{coverage}} + f_{\mathrm{cohesion}} + f_{\mathrm{readability}}$ $f_{\mathrm{coverage}} = Sim(s_{\mathrm{i}}, O) \qquad i = 1, 2, ..., n$ $f_{\mathrm{cohesion}} = 1 - Sim(s_{\mathrm{i}}, s_{\mathrm{j}}) \qquad i \neq j = 1, 2, ..., n$ $f_{\mathrm{readability}} = Sim(s_{\mathrm{i}}, s_{\mathrm{i}}) \qquad i \neq j = 1, 2, ..., n$
[55]	Covergae	$S = max(\sum_{s_{\mathrm{k}}} (c_{\mathrm{k}}, z_{\mathrm{k}}))$ $s.t. = \sum_{s_{\mathrm{k}}} (l_{\mathrm{k}}, z_{\mathrm{k}}) \leqslant L$

The results of this approach are very promising compared to other approaches. Moreover, there are little researches conducted on the Arabic language that use this approach.

2.3 Features Employed in Text Summarization

Selecting relevant sentences highly depends on the used features to represent words, sentences, paragraphs, and documents. Table 2.2 shows a brief description of all features used in text summarization and compares them from different aspects such as level and category [3, 4, 56, 57]. Based on the level, text summarization features are categorized into word, sentence, paragraph, and graph level. Also, the category is an important parameter to classify the summarization features into statistical or semantic.

Choosing and modeling the best features highly affect the selection of relevant sentences. As a result, the summarization system quality and performance. All features mentioned in the table below can be used in Arabic language except the upper case feature, since no capitalization in Arabic. Also, these features are used with both single and multi-document summarization problems. However, some modifications may be needed with multi-document summarization such as the similarity measures. It needs to be modified to represent the similarity across the different articles. For example, similarity to title feature is modified to be the similarity to all titles of the multi-documents.

Table 2.2: A brief description of the features adopted in text summarization systems [3, 4, 56, 57].

Feature Name	**Brief Description**	**Level**	**Category**
Word/Term Frequency	The number of times that term Ti occurs in a given document D	Word-level	Statistical
Term frequency/Inverse sentence frequency TF/ISF	Relate term frequency in a sentence to the number of times that the same term occurs along all sentences.	Word-level	Statistical
Gain	Modification on IDF formula to consider terms with medium frequency.	Word-level	Statistical

Upper case	Words that contain one or more upper case letters (e.g. proper names, highlighted words)	Word-level	Statistical
Fonts effects/style	Existence of Bold, Italic, and Underline	Word-level	Statistical
Proper noun	A noun that names a particular person, place, or thing, and mainly begin with capital letters, no matter where they occur within a sentence.	Word-level	Semantic
Word co-occurrences	Chance of two or more terms forming text in the sentence and appear alongside each other in the same, manner and position.	Word-level	Semantic
Lexical similarity	Measures the degree in which the words sets of two given sentences are similar (same meaning) or other semantic relation.	Word-level	Semantic
Key-phrases	A short list of important terms that provide a condensed summary of the main topics of a document.	Word-based	Semantic
Thematic features	The words that are most frequent in the document.	Word-level	Semantic
Similarity with title	Similarity or overlapping between a given sentence and the document title.	Word-level	Statistical
Cue words	Words that serve as explanation words such as in summery and for example.	Word-level	Statistical

Positive keywords	Words that are used to emphasize or focus on a special idea such as have outstanding, and support for.	Word-level	Semantic
Sentence length	Count the number of words in the sentence (can be used to classify the sentence as too short or too long).	Sentence-level	Statistical
Sentence inclusion of numerical data	Existence of numerical data in the sentence.	Sentence-level	Statistical
Sentence centrality	The similarity or the overlapping between a sentence and other sentences in the document without any semantic treatment	Sentence-level	Statistical
Sentence-to-Centroid Cohesion	The similarity between the sentence and the vector representing the centroid of the document.	Sentence-level	Statistical
Iterative query score	The ratio of the total count of sentences retrieved from the iterative query (top frequent words) and the total number of iterations.	Sentence-level	Statistical
Word similarity among paragraphs	Count words in a paragraph that occur more frequently in other paragraphs sentences	Paragraph-level	Statistical
Paragraph Feature	Location of the paragraph, e.g. first 10 paragraph and last 5 are the most important.	Paragraph-level	Statistical

Paragraph Position	Relate the position of a sentence to the paragraph.	Paragraph-level	Statistical
Text rank	Extract important keywords from the entire document along with its weight using a graph-based model.	Graph-level	Semantic
Bush path of the node	The number of links that connect the node to other nodes (sentences) on the map (the number of sentences that are similar to a particular sentence).	Graph-Level	Statistical
Aggregate similarity	Summing weights (similarities) of the links that connect a node (sentence) to other nodes (Bushy Path).	Graph-level	Statistical

According to [4, 57] TF, TF-ISF, Named Entity, and Proper Nouns are the best combination of features that could be used in text summarization.

Chapter 3 Multi-Objective Optimization with K-medoids Clustering for Multi-Document Summarization

Most of the available approaches have targeted the English language and other European languages, while little works have been introduced in Arabic language. Also, all the used summarization approaches illustrated in the previous chapter have many limitations. Statistical and graph-based approaches depend on statistical measures, so they fail to understand the meaning behind the text. In contrast, lexical and semantic approaches can handle linguistic features. However, these approaches highly depend on the ontologies, which are not available for all domains, and constructing them manually is a time consuming task.

For clustering-based approach, it is widely used with multi-document summarization to eliminate the redundancy. However, clustering techniques have many issues that affect the quality of the generated summary including the number of clusters, how to order them, how to select sentences, and finally how to merge the selected sentences to form the summary. These parameters are rarely considered together by researchers.

Regarding multi-objective optimization approach used for Arabic multi-document summarization, all systems deal with the contradictory objectives using the weighted sum approach. Also, the sentence's score is ignored in such systems, while it plays an important role to spur sentences that are important and not similar to other sentences to appear in the output summary.

Therefore, we propose an extractive Arabic multi-document summarization approach that employs clustering-based method and an evolutionary multi-objective optimization method. In this

chapter, the different stages of the proposed summarization system will be illustrated.

3.1 Proposed Methodology: An Overview

This section presents our methodology used for Arabic multi-document text summarization. The system integrates the multi-objective optimization approach with clustering techniques to extract the most important sentences that cover the main topics of the original source text while eliminating the redundant information from the generated summary. Figure 3.1 shows the stages of the proposed system. At first, a set of preprocessing steps is applied to represent the original text into a suitable form for text summarization such as tokenization, normalization, stop words removal, and stemming. Then, the documents are tokenized into sentences and each sentence is represented by a vector space model (VSM). The next stage is to express the importance of each sentence using a scoring based schema. The scoring based schema can handle the importance of a sentence, but it fails to eliminate the redundancy in the output summary or to capture the different topics appeared in the original text. Therefore, scoring is not sufficient to generate a good summary, which has the following characteristics: 1) coverage: summary has to cover the main topics appeared on the original source text; 2) diversity: summary should eliminate the redundant information. Therefore, the multi-objective optimization approach is used to optimize score, coverage, and diversity altogether. The next sections will discuss each stage in more details.

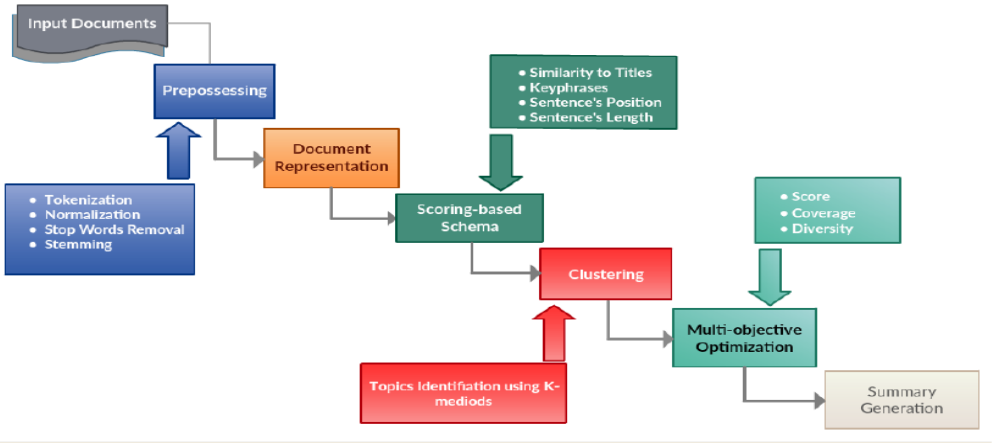

Figure 3.1: Flow of the main framework stages.

3.2 Preprocessing

Preprocessing aims to transform the original text documents into a unified form that facilitates working in the coming stages such as feature extraction. As shown in Figure 3.2, this stage includes tokenization, normalization, stop words removal, and stemming. Preprocessing tasks are used to reduce the ambiguity of words, and to increase the effectiveness of the proposed system.

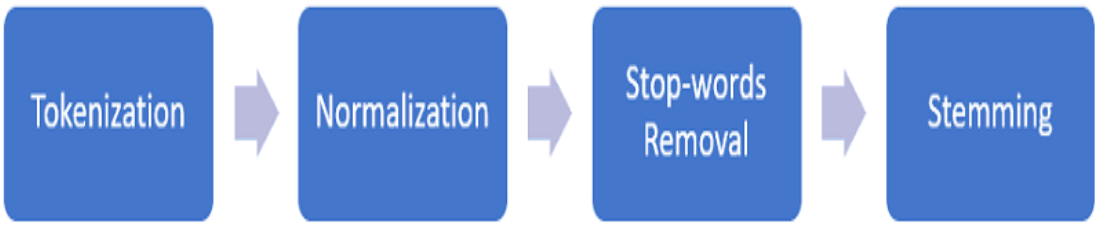

Figure 3.2: Preprocessing stages.

3.2.1 Tokenization

Tokenization is the process of splitting the document into small units such as paragraphs, sentences, and words [58]. This task is highly related to the morphological analysis. Therefore, it is a non-trivial task, even it is more complex when dealing with languages which have rich and complex morphology such as Arabic.

Text tokenization is used to split the input text into tokens. Tokens are the minimal syntactic unit which may be a word, part of a word, multi-word expression, or punctuation mark. The tokenizer is responsible for defining the word boundaries such as white spaces or punctuation marks. Also, it has to determine if the word has a stem or clitic. A clitic is similar to affix, but it serves syntactic functions such as negation, definition, conjunction or preposition. Therefore, the tokenizer is responsible for defining word boundaries, demarcating clitics, multi-word expressions, abbreviations, and numbers [58].

Many solutions for Arabic tokenization has been proposed, including punctuation marks and semantic-based systems. Punctuation marks based systems are suitable when the text is written with correct usage of punctuation marks. Also, it is simple and faster compared to the semantic-based systems. However, semantic-based systems are more efficient when the text lacks from punctuation marks, and it considers the morphological information during the tokenization.

A rule-based system was proposed by Attia [58], the system has three different levels based on the depth of the linguistic analysis involved. It determines the main tokens based on white spaces and punctuation marks, while morphological information is needed to determine the sub-tokens. This information is provided either deterministically by a morphological transducer, or in-deterministically by a token guesser. The output of this stage is a list of tokens that will be the input for further processing. Recently, many researchers such as [55] used the Stanford CoreNLP toolkit [59] for text segmentation. This toolkit is used widely in the research which is a Java-based framework that contains most of the NLP steps such as tokenization, Named Entity Recognition (NER), morphological analysis (stemming), and part-of-speech tagging (POS), sentence splitting and other another annotates (gender, sentiment). In text summarization, the input documents are split into sentences and each sentence into tokens.

In our research, tokenization is performed at two levels: at the sentence level based on punc-

tuation marks "**!?,.**", and at the word level using the white space, both levels are performed using java code. Indeed, we studied the effect of using Stanford CoreNLP for tokenization instead of relying on punctuation marks. The following figure shows an example from the input documents and its tokenized version using punctuation marks and Stanford tokenizer:

Figure 3.3: Text tokenization example.

It is clear from Figure 3.3 that the same input may be tokenized differently based on the used tokenizer. Using punctuation marks tokenizer, the input is tokenized into two sentences, while using Stanford tokenizer, one sentence is produced. Stanford tokenizer is suitable when the text is written as a long line without punctuation marks.

3.2.2 Normalization

Normalization aims to handle the different style in writing by transforming words that are differently written in a common form. For example, different authors can write 'مدرسة' or 'مدرسه', while both mean 'School', therefore normalization replaces 'مدرسة' by 'مدرسه'.

This stage aims to normalize the different forms of one letter into one form e.g. the normalization of 'أ ، آ ، إ' to 'ا'. Also, the normalization of 'ي' to 'ى', and 'ة' to 'ه'. Moreover, the symbol of shadda is replaced by duplicating the letter e.g. 'درّس' is transformed into 'دررس'. Also, diacritics are deleted from the text. In addition, altatweel is

removed e.g. 'البـــحر' is transformed into 'البحر' [6].

In our work, AraNLP which is a Java-based library is used to normalize the input text. Figure 3.4 shows a sample of output produced by the normalization step:

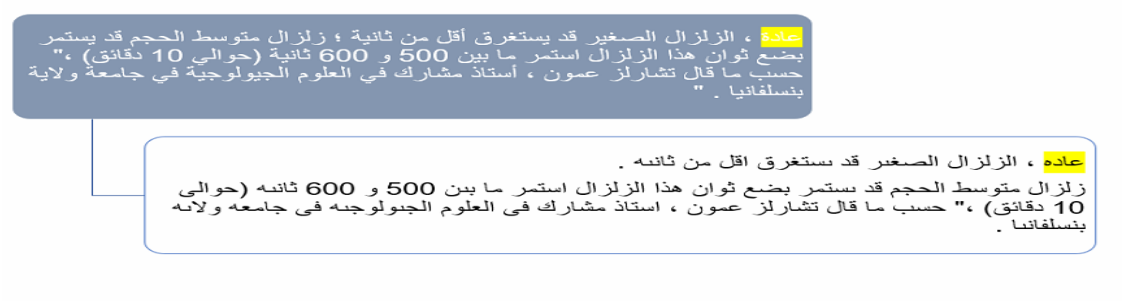

Figure 3.4: Text normalization example.

It is clear from the above Figure 3.4, how normalization replaces the ' ة ' to ' ه ' in the word 'عادة'.

3.2.3 Stop Words Removal

Stop words are words that appear frequently to connect the different units of text, but it doesn't hold any meaning in natural language processing analysis such as prepositions e.g. 'قبل ، بعد', pronouns such as ' انتم ،نحن', and conjunctions e.g. 'لكن ، ليس ، أنه' .

Stop words are removed since they are non-informative and increase the size of features to be processed. The list of stop words may be generic or application dependent so that there is no single list of stop words used by all NLP tools. With generic stop words, still the feature vector may be large since in each domain there is a set of words that occurred frequently but are not important for text analysis. In contrast, application dependent lists can reduce the feature vector, but it is not an easy task to create these lists.

Also, many approaches don't remove stop words in order to support phrase matching. Usually, researchers prepare his own list based on the collected dataset [6]. In this research, the general stop-words list [60], and Khoja's stop-words list [61] are combined and used. Figure 3.5 shows the output after the stop words removal step is applied to the input text.

Figure 3.5: Stop Words removal example.

3.2.4 Stemming

Stemming is the process of reducing a word to its word stem [62]. This task is very important for text summarization where sentences may have several forms of a particular word. Stemming process maps them to a common form (stem). Therefore, this will help a lot to reduce the redundancy where similar sentences may have the same words with different derivational forms [62].

Two main stemming approaches are used in Arabic language [63]. Light stemming is the first approach, where word's affixes are removed. On the other hand, morphological analyzers try to extract more complete forms using vocalization variation and derivation patterns. Two main types of analyzers exist including root-based stemmers and lemma based stemmers. Root-based stemmers try to find the abstract root of a word. This method reduces the dimension of document features space; however, words may lose their meanings e.g. 'مكتبة' which is a word referred to a place; however, stemming maps it to 'كتب', so it loses its merit. Lemma-based stemmers depend on morphological analysis and vocabulary usage to find a suitable root for a word such as Buckwalter [63].

In this study, we use both lemma-based and root-based stemmers to compare which is better in text summarization. Figure 3.6 shows the output of the stemming stage of the different stemming types.

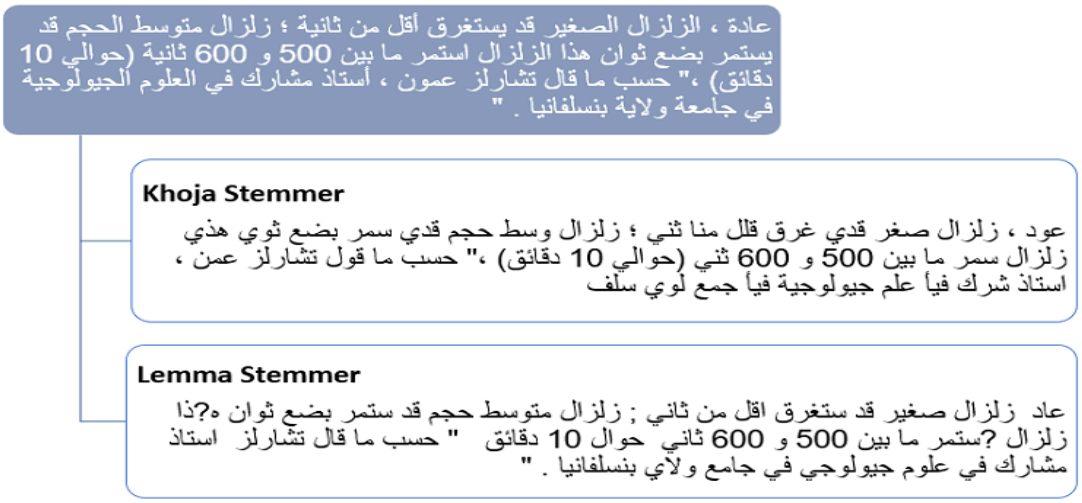

Figure 3.6: Stemming example.

It is clear from Figure 3.6 that lemma stemmer depends on the morphological analysis to find a suitable root for a word e.g. the name of a country 'بنسلفانيا' is kept unchanged, while in Khoja stemmer it is replaced by 'سلف' .

3.3 Document Representation

Preprocessing stages reduce differences between the writing styles, remove stop words, and reduce the feature vector dimension. Therefore, text matching and similarity measures perform better. In this stage, we discuss how the documents are represented.

Given a set of related documents D $= d_1, d_2, ..., d_i$, where i represents the number of documents. Tokenization step segments these documents into paragraphs, and then these paragraphs are tokenized further into sentences S $= s_1, s_2, ..., s_n$. Each sentence is represented as a bag of words (vector of words) using the vector space model. The vectors' dimension is determined by the distinct terms and phrases that appear in all documents. Different methods can be used to weight the terms; however, TF-ISF is the most common one [64]. The TF is used to measures the local importance of the term in a given sentence, while the ISF is used to measure the global importance among all sentences, the TF-ISF weight of term i in sentence j is calculated

as follows:

$$W_{\text{ij}} = tf_{\text{ij}} \times log_2 \frac{N}{n_{\text{i}}} \tag{3.1}$$

where:

tf_{ij} = number of occurrences of term i in sentence j.

N = total number of sentences.

n_{i} = number of sentences containing term i. The output of this stage is a vector of words with TF-ISF weights, Figure 3.7 shows how the vectors of features appear.

الإنترنت	عاده	الصغير	يستغرق	أقل	ثانية	.	زلزال	متوسط	الحجم	يستمر	يضم	توان	استمر
0	0.421534	0.198189	0.185235	0.192494	0.182896	0	0	0	0	0	0	0	0
0	0	0	0	0	0.182896	0	0.657223	0.044502	0.044502	0.044502	0.044502	0.044502	0.092774

Figure 3.7: Example showing how vectors of features are represented.

3.4 Summarization Approach

The proposed system has mainly three steps which are shown in Figure 3.8: 1) scoring the sentences, 2) identifying the topics appeared in the documents, then 3) multi-objective optimization.

Figure 3.8: The proposed system steps.

Using statistical features alone might not provide a rich information summary, because they

don't take into consideration the meaning and might cause some redundancy in the generated summary. On the other hand, relying on semantic features alone will not capture very important statistics like TF-ISF. Therefore, to handle these shortcomings, a combination of these types were used.

3.4.1 Scoring the sentences

In this research, we use the most important features to score sentences where this score is used later as an objective function to be maximized. The selection and formulation of the features are based on deep analysis of related work [3, 4, 57, 65], some hypothesis, our observations, and set of experiments. Using statistical features alone might not provide a rich information summary, because they don't take into consideration the meaning and might cause some redundancy in the generated summary. On the other hand, relying on semantic features alone will not capture very important statistics like TF-ISF. Therefore, to handle these shortcomings, a combination of these types were used including:

- **Similarity to titles:** The title is a very important sentence that document tries to explain and expand, therefore a very important feature to consider is the similarity between the sentence and the title. Moreover, in multi-document summarization, each document is written by a different author with a different title, so the similarity is measured between a sentence and a vector that combines all titles.

  Cosine similarity and keyphrases matching will be used as similarity measures which are calculated as follows:

$$cosinesimilarity(\vec{s_{\mathrm{i}}}, \vec{t}) = \frac{\vec{s_{\mathrm{i}}} \cdot \vec{t}}{|\vec{s_{\mathrm{i}}}| \times |\vec{t}|} \tag{3.2}$$

  where $\vec{s_{\mathrm{i}}}$ is the sentence's vector, and $\vec{t}$ is a vector that combines all titles. The cosine similarity is computed by performing the inner product of the two vectors, and this value is normalized using the length of vectors. However, cosine similarity doesn't distinguish terms, so keyphrases matching is used to express the existence of important terms, and is

calculated using the following equation:

$$\text{keyphrases}(\vec{s_i}, \vec{t}) = \frac{\text{No. of keyphrases in } \vec{s_i}}{\text{Total number of keyphrases in } \vec{t}} \tag{3.3}$$

These similarity measures are based on exact matching (statistically), therefore Arabic WordNet is used to find the synonymous for each term to match them semantically. Therefore, a vector of the word and its synonymous is created and the matching is based on this vector instead of the word itself. Finally, the similarity to titles feature is calculated as follows:

$$\text{SimilaritytoTitles}(\vec{s_i}) = \text{cosinesimilarity}(\vec{s_i}, \vec{t}) + \text{keyphrases}(\vec{s_i}, \vec{t}) \tag{3.4}$$

- **keyphrases:** Keyphrases are important words that reflect the topics of a document such as proper nouns and numbers, this feature is calculated by counting the number of keyphrases that appear in a sentence, and this feature is normalized by the total number of keyphrases in all documents.

$$\text{keyphrases}(\vec{s_i}) = \frac{\text{No. of keyphrases in } \vec{s_i}}{\text{Total number of keyphrases}} \tag{3.5}$$

- **Sentence's position:** This score ranks sentences based on their positions. The formulation of this feature is based on deep analysis of related work [3, 4, 57], and some hypothesis which can be modelled as follows:

  - The first sentence in the first and the last paragraphs expresses rich information and has the highest priority to appear in the output summary.
  - The first sentence in any paragraph other than the first and last paragraphs is important too.
  - Less score for sentences appear in the first and last paragraphs other than the first sentence.
  - The least score for remaining sentences based on their locations.

Therefore, the score of sentence i in paragraph j is computed as follows:

$$position(s_{\text{i}}, p_{\text{j}}) = \left\{ \begin{array}{ll} 3, & \text{First sentence-First pargraph} \\ 2, & \text{First sentence-Last pargraph} \\ 1, & \text{First sentence any pargraph} \\ \frac{1}{\sqrt{i}}, & \text{Other sentences in first or last pargraph} \\ \frac{1}{\sqrt{i+j^2}}, & \text{other sentences} \end{array} \right\}$$

The first sentence in the first paragraph has the highest score which equals '3'. Then, Less score that equals '2' is assigned to the first sentence in the last paragraph. Also, the first sentence in all paragraphs are considered important and has a score equals '1'. After that, the sentences appeared in the first or last paragraph other than the first sentence are assigned a score proportional to its position, the least score is for other sentences based on its position and its paragraph number. The square root is used in the last two scores to smooth the output values. Also, the division is used to produce smaller values when the sentences are less important. Finally, the output is divided by 3, so these values are normalized.

- **Sentence's length:** This feature counts the number of terms that appear in a sentence normalized by the length of the longest sentence, it is computed as follows:

$$\text{length}(\vec{s_{\text{i}}}) = \frac{\text{No. of terms in } \vec{s_{\text{i}}}}{|\text{longest sentence}|} \tag{3.6}$$

According to [4], which compares the performance of different sentence-based voting methods such as BordaFuse, CombMNZ, expCombANZ, etc., we adopt a weighted linear sum of normalized features score to evaluate each sentence in the document as shown in Equation-3.7.

$$\text{Score}(\text{s}_\text{i}) = w_1 \cdot \text{SimilaritytoTitles} + w_2 \cdot \text{keyphrases}(\vec{s_{\text{i}}}, \vec{t}) + w_3 \cdot \text{position}(\text{s}_\text{i},\text{p}_\text{j}) + w_4 \cdot \text{length}(\vec{s_{\text{i}}}) \tag{3.7}$$

The weight of each feature reflects its importance and effect on the total score. Therefore,

based on statistical measures like the mean and standard deviation. Also, a set of experiments is conducted to achieve close average and equally likely effect of the different features in the total score. As a result, w_2 is set to 3, w_4 is set to $\frac{1}{2.5}$, w_1 and w_3 are set to 1. Figure 3.1 summarizes the range of values for each feature before and after the weighting.

Table 3.1: The range of values for proposed features.

Feature Name	**Before weighting**	**Weighting factor**	**After weighting**
Similarity to Titles	$0 - 0.8$	1	$0 - 0.8$
Keyphrases	$0 - 0.6$	3	$0 - 1.8$
Sentence's position	$0 - 1$	1	$0 - 1$
Sentence's length	$0 - 1$	$\frac{1}{2.5}$	$0 - 0.4$

3.4.2 Topics Identification by Clustering

Each input document has a set of topics, in order to identify these topics we employ a clustering-based method. Clustering is used to group similar sentences in one cluster, dissimilar ones into different clusters, therefore each cluster represents a topic. As mentioned in chapter 2, a set of clustering techniques can be used in text summarization, but we chose k-medoid [66] clustering algorithm, which is a partition based clustering algorithm. This algorithm is widely used to overcome the weaknesses of k-means particularly the sensibility for noise points (outliers), also the medoid is a sentence not a number. The k-medoids starts by initializing the number of clusters. Since the best number of clusters is not known and varies between documents. Therefore, the silhoutte method [66] is used to determine the optimal number of clusters. Determining the optimal number of clusters which represents the main topics in the documents is a missing feature in all previous systems that use clustering in summarization.

K-medoids algorithm chooses a medoid (representative item) for each cluster at each iteration, the medoid is an object in a cluster that minimizes:

$$\sum_{j=C_{\mathrm{i}}} d(i,j) \tag{3.8}$$

The object i is the medoid of a cluster C_{i}, which minimizes the distance (d) between it and other objects in that cluster. The k-medoids is summarized by the following step [66]:

1. Randomly k-objects are chosen to be the initial cluster medoids.

2. Assign each object to a cluster with the closest medoid.

3. Recalculate the k-medoids using equation 3.8.

4. Repeat step 2 and 3 until medoids become fixed.

K represents the optimal number of clusters, this value is very important in summarization which indicates the number of different topics that exist in the original set of documents. The silhouette is suggested by Kaufman and Rousseeuw to determine the best cluster for each object, and assess the quality of the obtained clusters. For each object i, let $a(i)$ be the average distance from object i to all other objects in C_{i} cluster. On the other hand, for every $C \neq C_{\mathrm{i}}$, let $d(i, C)$ be the average distance from object i to all other objects in that cluster. After computing this value for all $C \neq C_{\mathrm{i}}$, $b(i)$ represents the minimum and its cluster is called the neighbour of i. The output of this stage is a set of clusters each one expresses a topic, and each cluster is represented by its medoid.
The number $s(i)$ is calculated using the following equation:

$$\frac{b(i) - a(i)}{max(b(i), a(i))} \tag{3.9}$$

This value measures how the object i fits to C_{i} cluster or its neighbour cluster. A negative value means that object is misclassified, a value equals to zero indicate that a neighbour cluster is more suitable for object i. If it is close to one, it means that object i fits well in its cluster. The average value of $s(i)$ for all objects in a cluster is called average silhouette width of that cluster. Moreover, the mean of $s(i)$ for all objects is called average silhouette width for the entire data set and is denoted by $\bar{s}\ (k)$, where k represents the number of clusters. Choosing k which maximizes $\bar{s}\ (k)$ represents the optimal number of clusters [66].

3.4.3 Multi Objective Optimization

Our summarization system aims to extract the most important sentences that cover the main topics of the original source text while eliminating the redundant information from the gener-

ated summary. Therefore, multi-document summarization problem can be formalized as multi-objective optimization problem with three objectives which are: coverage, diversity (less redundancy), and score. As mentioned in chapter 2, researchers differ in how they define and formalize these conflicting objectives. In this work, we searched most of the available work in this area, and reached a common definition for each objective [37, 38, 39, 40, 41]:

- **Coverage**

  Coverage means that summary should contain all important aspects appear in the documents, and we formulated the coverage as follows:

$$f_{\text{coverage}}(X) = \sum_{s_{\text{i}} \in Summary} sim(si, cj) x_{\text{i}} \tag{3.10}$$

  where

  sim represents the cosine similarity.

  s_{i}= The ith sentence

  c_{j}= The medoid of the jth cluster

  x_{i}= x_{i} equals 1 if sentence s_{i} is included in the summary, else it will be 0.

  The coverage is computed by finding the similarity between the selected sentence to be included in the summary and its k-medoids, which is generated by k-medoids clustering algorithm with silhouette method to determine the optimal number of clusters.

- **Diversity**

  Diversity aims to reduce sentences that are similar in the output summary. We calculated the redundancy as:

$$f_{\text{redundancy}}(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} sim(si, sj) x_{\text{i}} \tag{3.11}$$

  This equation measures the similarity between sentences in the output summary, therefore the system tries to minimize this objective to generate a good summary with less redundancy.

- **Score**

  Score objective tries to include important sentences in the output summary. We added the score of sentences as a new objective to be maximized, since all objective functions based on the similarity between the sentences included in the summary, while the score described in equation 3.6 indicates the importance of each sentence separately. Therefore, this objective will promote the sentences with a high score to be included in the summary.

$$f_{\text{Score}} = \sum_{s_{\text{i}} \in Summary} Score(s_{\text{i}}) \tag{3.12}$$

  where

  s_{i} represents the $sentence_{\text{i}}$ which is included in the summary.

  In summary, the proposed system can be considered as a maximization problem described in the following equation:

$$f(X) = maximize \quad (f_{\text{coverage}}, f_{\text{Score}}, \frac{1}{f_{\text{redundancy}}}) \tag{3.13}$$

  s.t.:

$$lenght(S) \leq l \tag{3.14}$$

  Where S is the summary, and l is computed based on the compression ratio. The length of the summary is a constraint, and the evolutionary optimization algorithms treat it as an objective. Therefore, the length is considered as objective to be maximized.

3.4.4 Optimization Process

Considering the text summarization as a multi-objective optimization problem is the only way to deal with conflicting objectives, as described in equation 3.13 we have 3 objectives to be maximized, while the length of the summary is considered as a constraint. Text summarization problem has a set of optimal solutions, therefore evolutionary algorithms (EA) will be used. As mentioned in chapter 2, different evolutionary algorithms are proposed to solve such systems

efficiently, and these algorithms are formulated as the following:

1. **Encoding of the individuals and population initialization**

   The initial population of individuals is generated randomly from the search space, At each iteration t, the pth individual of the population has n components:

$$U_{\mathrm{p}}(t) = u_{\mathrm{p,1}}(t), u_{\mathrm{p,2}}(t), ..., u_{\mathrm{p,n}}(t) \tag{3.15}$$

   where $u_{\mathrm{p,i}}$ represents the ith decision variable of the pth individual; p=1,...P where P represents the population size. The P solutions are initialized randomly. At the start run of the evolutionary algorithms, independent variables are initialized to their numerical range. As a result, the ith decision variable of the pth individual could be initialized as follows:

$$u_{\mathrm{p,i}}(t) = {u_{\mathrm{i}}}^{min} + ({u_{\mathrm{i}}}^{max} - {u_{\mathrm{i}}}^{min}) \cdot rand_{\mathrm{p,i}} \tag{3.16}$$

   where ${u_{\mathrm{i}}}^{max}, {u_{\mathrm{i}}}^{min}$ are the lower and upper bound of ith decision variable, and $rand_{\mathrm{p,i}}$ is a uniform random number between 0 and 1 [47, 50, 52].

   In text summarization, individuals represent the candidate set of sentences to form the summary. It is initialized randomly from the decision space, each individual is represented as a binary coded vector e.g. the individual $x(i) = (0, 1, 1, 0, ...)$ means that sentence 1, and 4 are not included in the summary, while sentence 2, and 3 are included. In this stage, a discretization [50] is needed to transform the real-coded values into binary-coded based on this formula:

$$u_{\mathrm{p,i}}(t+1) = \left\{ \begin{array}{ll} 1, & if \;\; rand_{\mathrm{p,i}} \leq sigm(u_{\mathrm{p,i}}(t+1)) \\ 0, & otherwise \end{array} \right\}$$

   where $sigm(z)$ represents the sigmoid function and is calculated as follows:

$$sigm(z) = \frac{1}{exp(-z) + 1} \tag{3.17}$$

2. **Mutation**

Mutation process is based on a mutation operator, this operator is used to maintain the diversity from one generation to the next one. It adds an amount obtained by the difference between randomly chosen individuals. A set of individuals are chosen (target vectors) to carry the mutation operation to create new individuals (mutant vectors). For each target vector U_p, three vectors U_{p1},U_{p2}, and U_{p3} are selected to generate the mutant vector V_p by adding the weighted difference of two vectors to the third:

$$V_{\mathrm{p}}(t) = U_{\mathrm{p1}}(t) + F \cdot [U_{\mathrm{p2}}(t) - U_{\mathrm{p3}}(t)] \tag{3.18}$$

where F is called the mutation factor, and it is usually a number between 0.4 and 1.0. The base vector (U_{p1}) is chosen in different ways, however choosing it randomly or to pick the best individual are the two most common techniques [47, 50, 52, 67].

In text summarization, mutation randomly removes or adds sentences to the summary while not violating the length constraint.

3. **Crossover**

Crossover is used to increase diversity in the population [50, 67]. Despite that many crossover techniques available, binomial crossover and exponential crossover are the two most common methods used differential evolution algorithms. With exponential crossover, the child inherits consecutively the parameters from mutant for a set of Bernoulli experiments. While, the inheritance is non-consecutive with binomial crossover, which aims to keep the child different from its parents [67].

The target vector $U_{\mathrm{p1}}(t)$ is mixed with the mutant $V_{\mathrm{p}}(t)$ to produce a trial vector $Z_{\mathrm{p}}(t) = z_{\mathrm{p,1}}(t), z_{\mathrm{p,2}}(t), ..., z_{\mathrm{p,n}}(t)$ where $z_{\mathrm{p,i}}(t)$ is obtained as follows:

$$z_{\mathrm{p,i}}(t) = \left\{ \begin{array}{ll} v_{\mathrm{p,i}}(t), & if \;\; rand_{\mathrm{p,i}} \leq CR \;\; or \;\; i = k \\ u_{\mathrm{p,i}}(t), & otherwise \end{array} \right\}$$

Where CR $\in [0, 1]$ is the crossover constant that controls the generation of trial vectors from the target and mutant vector, $\mathrm{rand}_{\mathrm{p,i}}$ is a random number between 0 and 1, and k $\in [1, ..., n]$ which is the randomly chosen to ensure at least one element from mutant

vector is obtained by the trial vector.

4. **Selection**

   The selection process is used to keep the population size constant over subsequent generations, where individuals that have objective functions better or equal to individuals in current population will survive for next generation. The target vector $U_{\mathrm{p}}(t)$ is compared with trial vector $Z_{\mathrm{p1}}(t)$ based on the objective function as follows:

$$U_{\mathrm{p}}(t+1) = \left\{ \begin{array}{ll} Z_{\mathrm{p}}(t), & if \;\; f(Z_{\mathrm{p}}(t)) \geq f(U_{\mathrm{p}}(t)) \\ U_{\mathrm{p}}(t), & otherwise \end{array} \right\}$$

   Where $U_{\mathrm{p}}(t+1)$ is the target vector of next generation, and $f(Z_{\mathrm{p}}(t))$ is the objective function value of the trial vector $Z_{\mathrm{p}}(t)$ [47, 50, 52].

5. **Stopping criterion**

   Mutation, crossover and selection processes will continue until stopping criteria is met such as (a) maximum number of iterations, (b) CPU time limits, (c) the best objective functions are not changed, and (d) achieving a predefined objective function values [47, 50, 52].

Based on the above steps, we can summarize the framework of the used evolutionary algorithm with the following steps:

Step 1: Initializing. The population is initialized randomly, where each individual is represented by a binary-coded vector as shown in Figure 3.9.

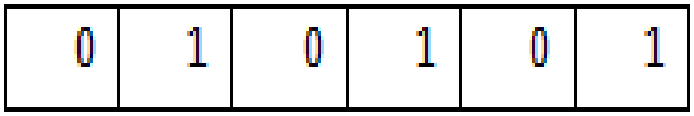

Figure 3.9: Binary-coded vector.

Step 2: Discretization. This step is needed to transform the real-coded values into binary-coded.

Step 3: Evaluation. Each solution is evaluated based on the objective functions.

Step 4: Mutation. Mutant vector is generated for the target vector using the mutant operator. In our problem, a bit flip mutation is used to produce the mutant vector as shown in Figure 3.10.

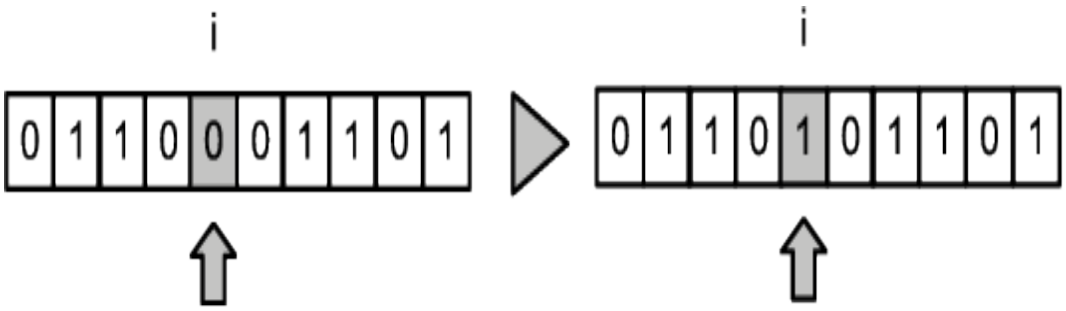

Figure 3.10: Bit flip mutation.

Step 5: Crossover. A trial vector is generated from the target vector using the crossover operator. In our problem, single point crossover is applied to the target vector with mutant vector to produce the trial vector as shown in Figure 3.10.

Figure 3.11: Single point crossover.

Step 6: Selection. The non-dominant solutions from the trial vectors are chosen to be the next generation.

Step 7: Stopping. The steps from 2 to 6 are repeated until a stopping criterion is met.

3.4.5 Evolutionary Multi-objectives Optimization Algorithms

Different evolutionary algorithms exist such as Non-dominated Sorting Genetic AlgorithmII (NSGAII), Indicator-Based Evolutionary Algorithm (IBEA), Strength Pareto Evolutionary Algorithm2 (SPEA2), and Multi-Objective Cellular algorithm (MOCell).

IBEA: This algorithm takes the user preferences into consideration, and it is often used as a baseline to evaluate the performance of other multi-objective algorithms [68].

MOCell: This algorithm uses an external set (archive) with the original population. Each individual finds its neighbour list, then two elements are chosen to form the parents. These parents will be used to generate the offspring through the crossover and mutation. The offspring is added to the archive if it is non-dominated, and it replaces the current element in the population if it isn't dominated by it. A set of archive elements are used to enhance the solutions by moving a number of non-dominated solutions from the archive to the population [69].

NSGAII: This algorithm is an improvement on the earlier multi-objective evolutionary algorithm NSGA. The population is sorted into fronts, where each front contains a set of solutions with the same fitness value. The solutions with the highest fitness values will be in the better fronts. The crowding distance metric which measures the distance of two neighbouring solutions on either side in each objective axis is used to distinguish solutions on the same front. solutions with different non-domination levels and better fitness values will be taken. Otherwise, the one with a higher crowded distance will be chosen to form the optimal Pareto-front [70, 71].

SPEA2: It is an improvement on earlier multi-objective evolutionary SPEA. It uses a regular population and an external set (archive). At the start time, the external set is empty, then iteratively the non-dominating solutions are added to the set. The archive has a fixed size, so if the non-dominating solutions are not sufficient; the dominated solutions will be added. The fitness value assigned to each member by counting the solutions dominated by this member. Also, SPEA2 has another metric which is the density estimation to avoid fitness sharing [72].

3.4.6 Summary Generation

The output of the optimization process is a set of solutions that are non-dominated by others in terms of the objective functions. In this context, the retrieved solution is a binary sequence

where ones indicate that the sentence must be included in the output summary, whereas a zero means to exclude it. However, we have a set of solutions not a single one, so the question then arises, ”Which sentences to include in the output summary?”.

Majority voting approach is adopted to combine the solutions, which is simple and performs very well with real data [73]. Two main approaches are tested: the first one is the majority voting approach over all the non-dominated solutions, where the summary is formed by choosing the set of sentences that appeared in most of the solutions (population). The number of the sentences is determined by the average number of the sentences in the corresponding gold summaries. For example, if the gold summaries of one topic have an average of 8 sentences, the top 8 sentences that appeared frequently in the solutions will be the summary. Figure 3.12 simulates the majority voting approach.

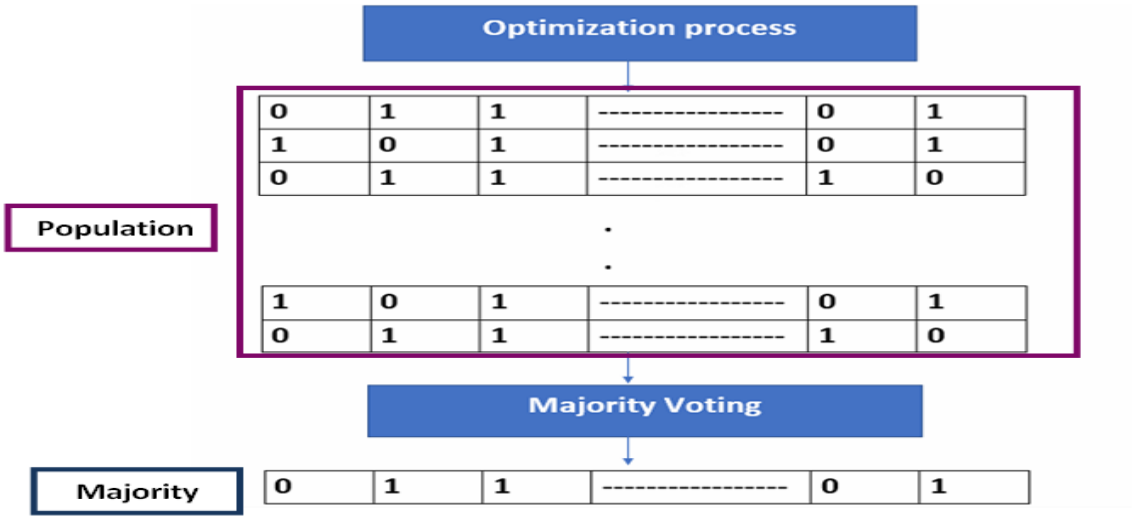

Figure 3.12: Majority voting approach.

The second one is a majority voting with a rule-based approach. The majority voting is applied over 3 solutions, which are the solutions that have the best fitness values. Also, a rule-based approach is applied when the majority voting output is longer than the desired summary, so sentences with the highest score are chosen. Figure 3.13 describes the majority voting with rule-based approach.

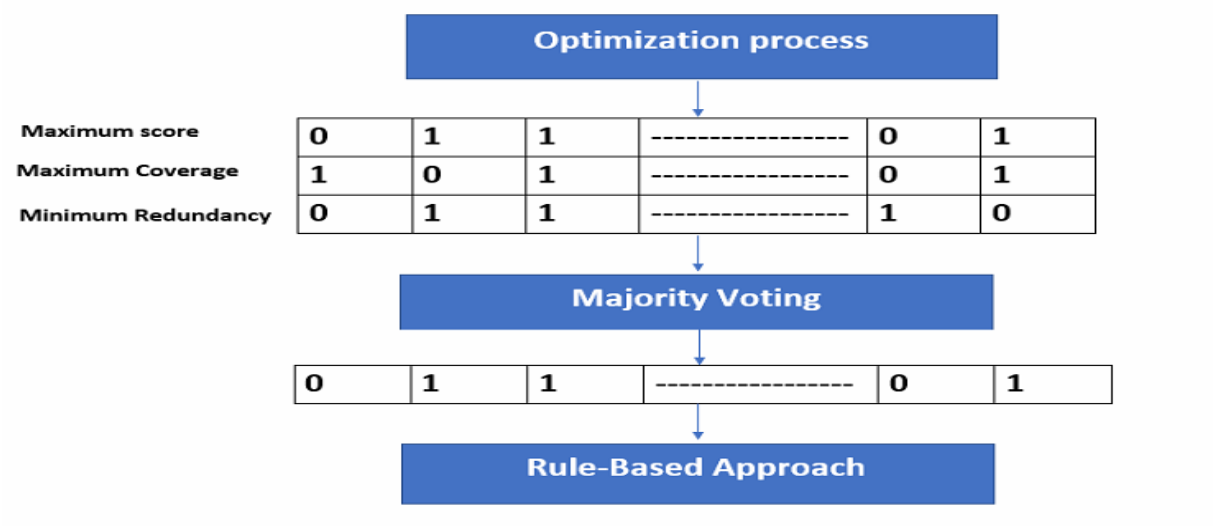

Figure 3.13: Majority voting with rule-based approach.

The following example shows the output summary of 10 articles talking about Tsunami, and summarized into six sentences:

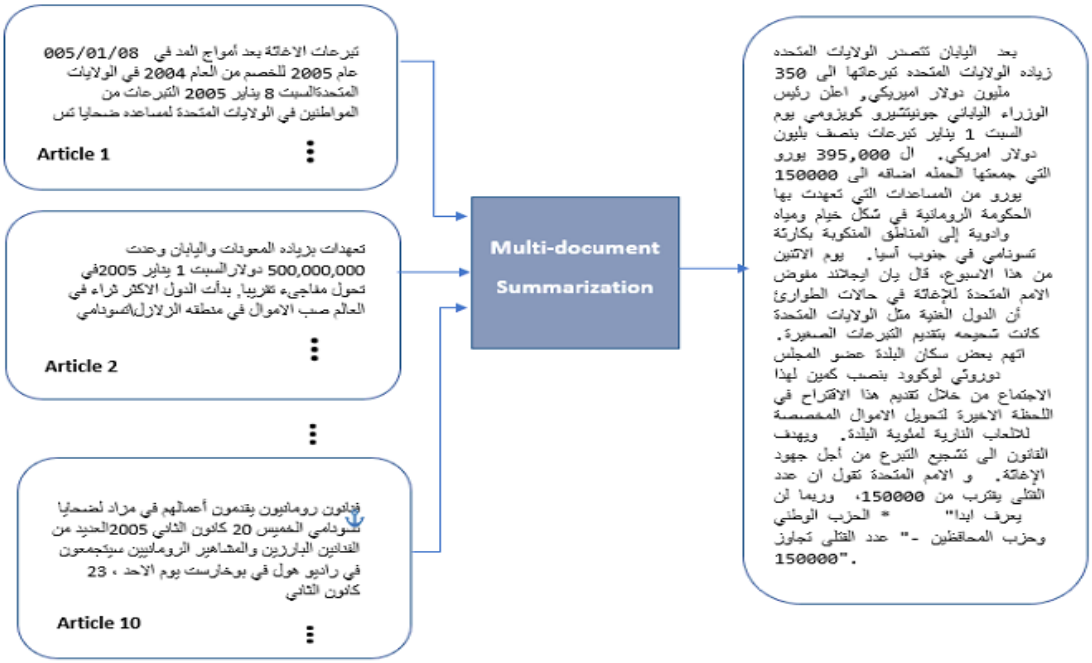

Figure 3.14: Multi-document summarization example.

Chapter 4 Experimental Results

This chapter starts by describing the available datasets that are used in Arabic text summarization. Also, it shows the evaluation measures to compare the output summaries with the gold standard summaries. Another section is dedicated to illustrate the tools which are used in each stage of the proposed system. The rest of the chapter presents experiments, results, and discussion.

4.1 Data sets

Any text summarization system needs a collection of documents and their gold-standard summaries to test the proposed system and compare its performance and quality with others. Dataset may be generated manually by human such as crowdsourcing, or automatically using Machine Translation (MT). The dataset generated manually is adequate and accurate; however, it takes a long time and effort, and sometime it could suffer from subjectivity [9].

For Arabic language, there are little corpses available for single and multi-document summarization. Mahmoud al-Haj [9] has created a set of resources for Arabic language summarization systems.

Creating resources for multi-document summarization is not an easy task as single document. It costs time and money. Therefore, Machine Translation (MT) is used to translate the English dataset (DUC 2002) from English to Arabic. Despite that translation saves time and money, it may affect the cohesion of the text. The java version of Google Translate API is used to translate the dataset sentence by sentence. Table 4.1 shows the corpus statistics of the DUC2002 [9]. Also, other versions of DUC dataset are available (DUC 2003-DUC 2007). However, we chose

to work with DUC 2002 since some Arabic systems already participated with this dataset, so we can compare our results to them.

Table 4.1: DUC-2002 Arabic Corpus Statistics [9].

Corpus Name	**DUC2002 (Arabic)**
Number of Documents	567
Number of Sentences	17,340
Number of Words	199,423
Number of Distinct Words	19,307
Number of Reference Sets	59
Documents per Reference Set	10 on average
Number of Gold-standard summaries	118 (two for each reference set)

Also, a multilingual dataset (TAC 2011 MultiLing Pilot dataset) is created by translating an English data set into 6 languages (Arabic, Czech, French, Greek, Hebrew, Hindi). The original dataset is extracted from WikiNews website. It contains 100 document classified into 10 clusters that cover a variety of topic. For each document set, 3 summaries are created by three different participants. Table 4.2 shows the TAC-2011 Arabic Dataset Statistics [9].

Table 4.2: TAC-2011 Arabic Dataset Statistics [9].

Corpus Name	**TAC 2011 MultiLing (Arabic)**
Number of Documents	100
Number of Sentences	1,573
Number of Words	30,908
Number of Distinct Words	9,632
Number of Reference Sets	10
Documents Per Reference Set	10 on average
Number of Gold-standard summaries	3 (three for each reference set)

Moreover, El-Haj and Koulali [5] have built a multi-purpose Arabic corpus called KALIMAT. It consists of 20,291 Arabic articles collected from an Omani newspaper along with automatically generated extractive summaries (20,291 single and 2057 multi-document summaries).
In this study, we use TAC-2011 Multi-Ling as the main dataset where the Arabic version is ready and available for research. However, Arabic version of DUC2002 is not available and thus we

have obtained the raw data set and translated it using Google translator. Thus, DUC2002 was used to test the effectiveness of the main approach only. For both datasets, we tokenized the gold summaries to determine the number of sentences that will be chosen to form the output summary using the majority voting approach mentioned in Chapter 3.

4.2 Evaluation Measures

Evaluating the summarization system is not an easy process since no ideal summary exists. Therefore, two main evaluation metrics have been used, which are: form metric and content metric. Form metric focuses on grammar, organization, and content structure, while content metric tries to compare the generated summary unit by unit with the human summaries.

Also, the evaluation could be classified as human-based or automated based. Human evaluation provides better quality and accuracy; however, it costs time and effort since the human has to evaluate the generated summary manually. Mahmoud El-Haj et al. [74] proposed two systems Arabic Query-Based Text Summarization System (AQBTSS) and Arabic Concept-Based Text Summarization System (ACBTSS). They have invited 1,500 participants to evaluate the readability of the generated summary from the two systems. They have asked them to read and evaluate the two systems based on the five-point Likert scale. The evaluation scale used to evaluate the two systems is shown in Table 4.3.

Table 4.3: The evaluation scale used to evaluate the two systems [74].

Evaluation	**Score**	**Interpretation**
V. Poor	0	The summary is very poor and is not related to the document at all.
Poor	1	The summary is poor as the core meaning of the document is missing.
Fair	2	The user is somehow satisfied with the result but expected more.
Good	3	The summary is readable and it carries the main idea of the document.
V. Good	4	The summary is very readable and focuses more on the core meaning of the document. The user is happy with the results

Human evaluation suffers from inconsistency. Therefore, automatic evaluation metrics are used

to match the human evaluation for content matching and machine translation such as Recall-Oriented Understudy for Gisting Evaluation (ROUGE) and BLUE [9].

ROUGE is used to determine the quality of the generated summary compared to the ideal summaries generated by a human. Different measurement metrics exist, including Rouge-N, Rouge-W, Rouge-L, Rouge-S, and Rouge-SU [6]. In general, ROUGE-N with uni-gram and bi-grams are used in most researches.

1. ROUGE-N: It counts the number of n-gram matches between the system summary and the gold-standard summary, ROUGE-N is calculated using Equation 4.1.

$$ROUGE - N = \frac{\sum_{S \in ReferenceSummaries} \sum_{gram_{\mathrm{n}} \in S} Count_{\mathrm{match}}(gram_{\mathrm{n}})}{\sum_{S \in ReferenceSummaries} \sum_{gram_{\mathrm{n}} \in S} Count(gram_{\mathrm{n}})} \quad (4.1)$$

   Where $\mathrm{Count}_{\mathrm{match}}$ represents the maximum number of n-grams occurred in both the candidate summary and the gold summaries. While the denominator of the equation represents the sum of n-grams appeared in the set of reference summaries, since the ROUGE is a recall-based measure. Also, n represents the length of the n-gram e.g. n equals 1 with ROUGE-1.

2. ROUGE-L: It measures the longest matching of words between the two summaries using Longest Common Sequence (LSA).

3. ROUGE-S: It measures the overlap of skip bi-grams between the two summaries.

4. ROUGE-SU: The same as ROUGE-S, while it considers the unigram-based co-occurrence statistics.

Three measures are used to evaluate the generated summary: precision, recall, and F-measure. Precision (P) represents the percentage of information presented in the system's summary. Recall (R) measures the coverage of the system. The precision checks the correctness of the summary, while the recall measures the completeness of the summary. Therefore, the F-measure can be viewed as a compromised version between recall and precision. These measures are

computes using the following equations:

$$\text{Precision} = \frac{\text{The number of retrieve and relevant sentences extracted by the system}}{\text{Total number of sentences extracted by the system}} \quad (4.2)$$

$$\text{Recall} = \frac{\text{The number of retrieve and relevant sentences extracted by the system}}{\text{Total number of sentences extracted manually}} \quad (4.3)$$

$$\text{F-measure} = \frac{2PR}{P+R} \quad (4.4)$$

4.3 Tools

In this section, all tools used in the implementation of the proposed system are summarized in Table 4.4.

Table 4.4: Used tools to implement the different stages of the proposed system.
stop words removal, and stemming

Stage	**Tool**
Preprocessing	• Tokenization- Hard coded (Java language), and Stanford CoreNLP • Normalization- AraNLP • Stop Words Removal- Hard coded (Java language) • Stemming- MADAMIRA v2.1, and Java API
Document Representation	Hard coded- Java language.
Features Extraction	Hard coded- Java language.
K-medoid clustering	R.
Silhouette	R.
Multi-objective Optimization	jMetal-5.1.
ROUGE	ROUGE-1.5.5-Python version.

Different tools have been used in the preprocessing stage. At first, Stanford CoreNLP [59] has been used for text tokenization. This tool is useful when the text is written as long as the text without punctuation marks. As a comparison, we have tokenized the text using a java code based on the punctuation marks too. On the other hand, stemming stage was implemented us-

ing two different stemmers, lemma stemmer from the MADAMIRA tool [75] which is one of the most important tools in Arabic NLP that used for tokenization, Named Entity Recognition (NER), morphological analysis (stemming), part-of-speech tagging (PSO), and discretization. Also, the Java version of Khoja stemmer has been used as root stemmer. In addition, AraNLP normalizer is used to transform words that are differently written into a common form. This tool is a Java-based library that gathers a set of Arabic text preprocessing tools into one library [76]. Moreover, text clustering was implemented using R which is a free software under the GNU General Public License. It provides a set of statistical techniques such as classical statistical tests, time-series analysis, classification, and clustering. Moreover, R can run on a wide variety of UNIX platforms, Windows, and MacOS [77].

Furthermore, jMetal is a java based framework for multi-objective optimization with meta-heuristic techniques. It has a set of classes, which are used for building block for multi-objective algorithms. Therefore, algorithms share a set of components such as genetic operators and density estimators. Also, it has set of quality indicators to measure and compare the performance of different algorithms [78]. Different algorithms implemented in jMetal framework such as: Non-dominated Sorting Genetic AlgorithmII (NSGAII), Indicator-Based Evolutionary Algorithm (IBEA), Strength Pareto Evolutionary Algorithm2 (SPEA2), and Multi-Objective Cellular algorithm (MOCell). Based on a set of experiments, the time required for both MOCell and IBEA is too large compared to SPEA2 and NSGAII. Also, it is well-known that NSGAII can find solutions with better spread and convergence with most problems [79]. Therefore, we chose to work with SPEA2 and NSGAII.

4.4 Experiments, Results, and Discussion

A list of experiments is conducted to show the effectiveness of the proposed approach. This section presents the experiments and the evaluation results, then it discusses them.

4.4.1 Experiments Setup

As mentioned in chapter 3, the input of the summarization system is a set of related documents D $= d_1, d_2, ..., d_{\text{i}}$, where i represents the number of documents. Tokenization step segments these documents into paragraphs, and then these paragraphs are tokenized further into sentences S $= s_1, s_2, ..., s_{\text{n}}$. Each sentence is represented as a bag of words (vector of words) using the vector space model. After that, the objective functions are calculated as shown in chapter 3. Then, the multi-objective optimization process is started to find the optimal set of solutions. All experiments use ROUGE-1.5.5 with the following parameters: -a -2 4 -u -c 95 -r 1000 -n 2 -f A -p 0.5 -t 0. The parameters used with rouge 1.5.5 are described in Table 4.5.

Table 4.5: The parameters used with ROUGE 1.5.5.

Parameter	**Description**
-a	Evaluate all systems
-n	Max n-gram
-2	Max-gap-length
-u	Include unigram in skip-bigram
-c	The confidence interval
-r	Number-of-samples
-f	Scoring formula
-p	Alpha, which is between 0 and 1
-t	Count by token instead of sentence
-d	Print per evaluation scores

We start by studying the effect of preprocessing in text summarization, then the effect of the used algorithm. Also, the effect of using the score as an additional objective function is investigated. Moreover, the effect of summary generation approach is studied. Table 4.6 summarizes all variable parameters in our system:

Table 4.6: The variable parameters in our system.

Parameter	**Values**
Tokenization	Stanford, and punctuation marks (Code)
Stemming	Lemma, and Khoja
Algorithm	NSGAII, and SPEA2
Population size	50, 100, and 150
Crossover	Single Point Crossover
Mutation	Bit Flip Mutation
Objective functions	Score, Coverage, Diversity
Summary Generation Approach	Majority voting, Majority voting with rule-based method

4.4.2 The Effect of Preprocessing in Arabic Text Summrization

As mentioned before, each stage of the preprocessing has alternatives which may affect the results of the proposed system. So, we studied the effect of tokenization and stemming in text summarization.

All the experiments are performed with Binary Tournament Selection, population size = 50, crossover Probability = 0.9, Max Iterations = 25000, mutation probability = $\frac{1.0}{Total\#ofBits}$, and NSGAII algorithm. Also, all experiments were performed on a machine that runs windows 7 ultimate with Intel(R) Core(TM) i7-3612QM CPU 2.1 GHz processor and 8 GB RAM.

The TAC 2011 MultiLing Pilot dataset is tokenized using Stanford CoreNLP and punctuation marks. Also, the dataset is stemmed using two stemmers: lemma and Khoja. Therefore, four versions of the data set are generated which are: Stanford tokenizer with lemma, Stanford tokenizer with Khoja, punctuation marks (Code) tokenizer with lemma, and punctuation marks (Code) tokenizer with Khoja.

The dataset has 10 different topics (reference sets); however, the results are reported for 9 only since the gold summaries of reference set # 8 have problems with rouge 1.5.5. As mentioned in chapter 3, the initial population of the evolutionary is randomly generated. Moreover, random variables are used to control the different steps such as population's update, crossover, and mutation. Therefore, we can not rely on the results of a single run, so that we use the average F-measure of 5-Independent runs to compare the results of our approach to the systems

participating with this dataset. Table 4.7 and 4.8 report the results of these experiments:

Table 4.7: The F-measure of 5-Independent runs of Stanford tokenizer with different stemmers.

	Stanford + Lemma				**Stanford + Khoja**			
Reference set	**R-1**	**R-2**	**R-L**	**R-SU4**	**R-1**	**R-2**	**R-L**	**R-SU4**
Reference set # 1	0.272	0.036	0.256	0.070	0.336	0.057	0.315	0.112
Reference set # 2	0.342	0.108	0.340	0.122	0.363	**0.171**	0.355	0.145
Reference set # 3	0.406	0.083	**0.394**	0.150	0.422	0.129	0.347	0.169
Reference set # 4	**0.411**	0.136	0.391	**0.180**	**0.513**	0.156	**0.476**	**0.269**
Reference set # 5	0.214	0.032	0.214	0.027	0.230	0.101	0.230	0.035
Reference set # 6	0.339	**0.216**	0.325	0.108	0.373	0.162	0.345	0.127
Reference set # 7	0.256	0.112	0.218	0.083	0.289	0.103	0.226	0.083
Reference set # 8	0.263	0.063	0.202	0.091	0.301	0.059	0.227	0.089
Reference set # 9	0.353	0.126	0.339	0.106	0.371	0.119	0.334	0.123
Average	0.317	0.101	0.297	0.104	**0.355**	**0.117**	**0.317**	**0.128**

Table 4.8: The F-measure of 5-Independent runs of punctuation marks tokenizer and lemma stemmer.

	Punctuation marks + Lemma				**Punctuation marks + Khoja**			
Reference set	**R-1**	**R-2**	**R-L**	**R-SU4**	**R-1**	**R-2**	**R-L**	**R-SU4**
Reference set # 1	0.296	0.046	0.285	0.089	0.292	0.029	0.286	0.091
Reference set # 2	0.423	0.201	0.388	**0.182**	0.405	0.199	0.402	0.158
Reference set # 3	**0.441**	0.160	0.363	0.162	0.510	0.279	0.390	0.179
Reference set # 4	0.403	0.126	**0.391**	0.173	0.458	0.153	0.435	0.231
Reference set # 5	0.196	0.087	0.196	0.0350	0.234	0.116	0.234	0.052
Reference set # 6	0.373	**0.176**	0.328	0.155	**0.517**	**0.320**	**0.482**	**0.281**
Reference set # 7	0.317	0.096	0.241	0.105	0.428	0.227	0.361	0.204
Reference set # 8	0.320	0.112	0.280	0.116	0.320	0.112	0.280	0.116
Reference set # 9	0.348	0.154	0.327	0.112	0.331	0.158	0.316	0.113
Average	0.346	0.129	0.311	0.125	**0.389**	**0.177**	**0.354**	**0.158**

It is clear from tables 4.7 and 4.8 that the best results achieved when punctuation marks tokenizer and Khoja stemmer are used. As shown in chapter 3, the punctuation marks tokenizer is more stable when the data is written with correct usage of punctuation marks, while Stanford tokenizer can be more effective if the data written as long line without punctuation marks. In our case, TAC 2011 corpus is written with full usage of punctuation marks, so this explains why

punctuation marks tokenizer outperforms Stanford tokenizer. On the other hand, Khoja stemmer beats lemma stemmer. This can be explained as follows: Khoja stemmer is a root-based stemmer which retrieves the root of a word and this increases the semantic similarity between sentences. Based on this result, all next experiments are conducted with punctuation marks tokenizer and Khoja stemmer. The boxplot of punctuation marks tokenizer and Khoja stemmer results is shown in Figure 4.1.

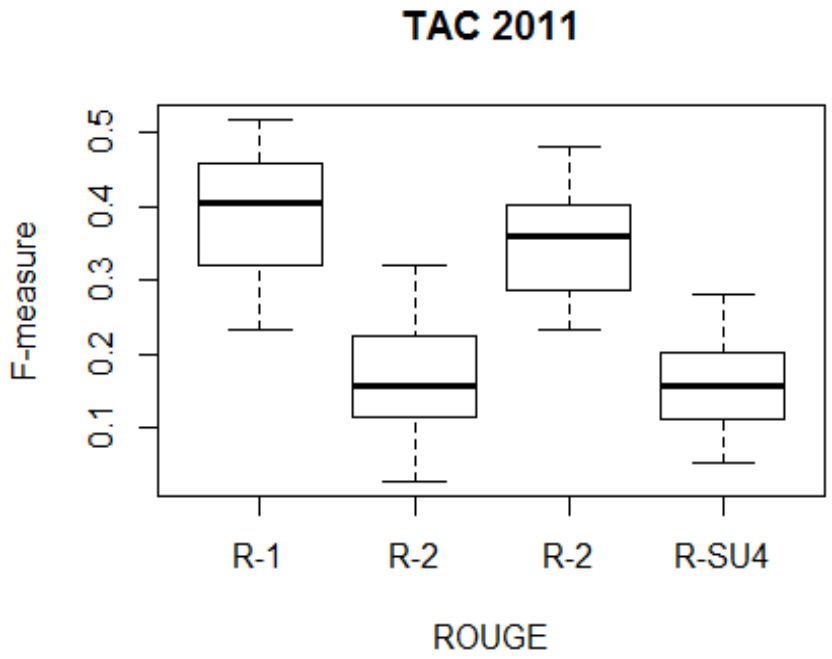

Figure 4.1: Boxplot of TAC 2011 ROUGE results.

On the other hand, the DUC 2002 data set was tokenized into sentences. Two versions of the data set generated which are: DUC 2002 with lemma and DUC 2002 with Khoja. The original data set has 58 reference sets (topics) of documents, but we faced some issues during the parsing of the data from the xml files, translation, and during the evaluation. Therefore, the results are reported for 28 clusters. The average F-measure of all 5-Independent runs are summarized Table 4.9.

Table 4.9: The F-measure of all 5-Independent runs with DUC 2002 dataset.

System	R-1	R-2	R-L	R-SU4
DUC 2002 + Lemma	0.37158	0.16214	0.37158	0.14400
DUC 2002 + Khoja	**0.47055**	**0.23733**	**0.47055**	**0.20355**

Also, with DUC 2002 data set, the results achieved with Khoja stemmer is higher than lemma stemmer. As we explained before, Khoja stemmer is a root-based stemmer which retrieves the root of a word, and this increases the semantic similarity between sentences. The boxplot of DUC results with Khoja stemmer is shown in Figure 4.2.

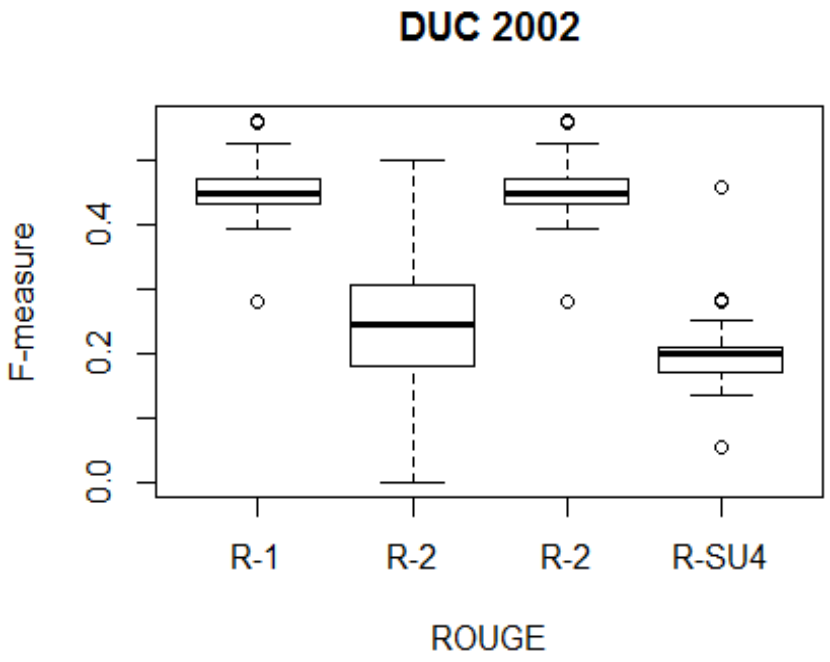

Figure 4.2: Boxplot of TAC DUC 2002 ROUGE results.

4.4.3 The Effect of Summary Generation Approach

Two main experiments were conducted to show the effectiveness of the proposed approach that used to form the output summary. The first one, it uses the majority voting over all population size, and the second one majority voting over the top 3 solutions with rule-based.

These experiments are conducted with punctuation marks and Khoja stemmer, NSGAII, Binary Tournament Selection, crossover Probability = 0.9, Max Iterations = 25000, mutation probability = $\frac{1.0}{Total\#ofBits}$, and 50 population size. The results of the two approaches are shown in Table 4.10.

Table 4.10: The F-measure of 5-Independent runs of different summary generation techniques.

	Majority voting + Rule based				**Majority voting**			
Reference set	**R-1**	**R-2**	**R-L**	**R-SU4**	**R-1**	**R-2**	**R-L**	**R-SU4**
Reference set # 1	0.221	0.033	0.207	0.057	0.292	0.029	0.285	0.091
Reference set # 2	0.248	0.088	0.221	0.058	0.405	0.199	0.402	0.158
Reference set # 3	0.337	0.071	0.29	0.081	0.510	0.279	0.390	0.179
Reference set # 4	0.358	0.115	0.342	0.175	0.458	0.153	0.435	0.231
Reference set # 5	0.216	0.101	0.216	0.048	0.234	0.116	0.234	0.052
Reference set # 6	0.375	0.208	0.345	0.188	0.518	0.320	0.482	0.281
Reference set # 7	0.278	0.109	0.226	0.108	0.428	0.227	0.361	0.204
Reference set # 8	0.267	0.085	0.234	0.094	0.320	0.112	0.280	0.116
Reference set # 9	0.254	0.102	0.243	0.082	0.331	0.158	0.316	0.113
Average	0.284	0.101	0.258	0.099	**0.389**	**0.177**	**0.354**	**0.158**

It is clear that the majority voting approach with all solutions is better than relying on 3 solutions where each one is the best in one objective, and applying a rule-based to cut sentences when the summary size is greater than the model summaries.

4.4.4 The Effect of Population Size

In this section, the population size effect on the results is studied. All experiments are conducted with punctuation marks and Khoja stemmer, NSGAII, Binary Tournament Selection, crossover Probability = 0.9, Max Iterations = 25000, and mutation probability = $\frac{1.0}{Total\#ofBits}$. Table 4.11 shows the results with 100 and 150 population sizes.

Table 4.11: The F-measure 5-Independent runs Using NSGAII and population size 100.

	NSGAII + 100 population size				**NSGAII + 150 population size**			
Reference set	**R-1**	**R-2**	**R-L**	**R-SU4**	**R-1**	**R-2**	**R-L**	**R-SU4**
Reference set # 1	0.204	0.015	0.191	0.043	0.170	0.019	0.164	0.035
Reference set # 2	0.357	0.173	0.350	0.121	0.309	0.118	0.302	0.107
Reference set # 3	0.458	0.224	0.338	0.167	0.423	0.191	0.328	0.138
Reference set # 4	0.357	0.109	0.341	0.135	0.315	0.079	0.296	0.099
Reference set # 5	0.179	0.079	0.172	0.031	0.206	0.086	0.206	0.039
Reference set # 6	0.454	0.271	0.411	0.212	0.321	0.151	0.294	0.108
Reference set # 7	0.397	0.222	0.380	0.187	0.332	0.172	0.120	0.141
Reference set # 8	0.274	0.116	0.248	0.087	0.246	0.059	0.207	0.069
Reference set # 9	0.226	0.084	0.220	0.064	0.203	0.0304	0.182	0.062
Average	**0.322**	**0.144**	**0.295**	**0.116**	0.281	0.101	0.233	0.089

The results obtained for the different population sizes show that 50 is the best size of the population, where the ROUGE results decrease when the population size increases.

4.4.5 The Performance of Two Well-known Multi-objective Optimization Algorithms

In this section, we study the effect of different algorithms in text summarization. All experiments are conducted with punctuation marks and Khoja stemmer, Binary Tournament Selection, crossover Probability = 0.9, Max Iterations = 25000, and mutation probability = $\frac{1.0}{Total\#ofBits}$. The behaviour of SPEA2 and NSGAII is studied in text summarization. The results are compared based on the ROUGE values and the running time of 5 independent runs. Table 4.12 presents the results of NSGAII and SPEA2:

Table 4.12: The F-measure and running time of 5-Independent runs Using NSGAII and SPEA2.

	NSGAII					SPEA2				
Reference set	**R-1**	**R-2**	**R-L**	**R-SU4**	**Time(s)**	**R-1**	**R-2**	**R-L**	**R-SU4**	**Time(m)**
Reference set # 1	0.292	0.029	0.286	0.091	2.5	0.206	0.043	0.206	0.043	6
Reference set # 2	0.405	0.199	0.402	0.158	2.8	0.388	0.172	0.388	0.150	4
Reference set # 3	0.510	0.278	0.390	0.179	3.8	0.376	0.153	0.328	0.105	5
Reference set # 4	0.458	0.153	0.435	0.231	3.4	0.270	0.075	0.265	0.069	4
Reference set # 5	0.234	0.116	0.234	0.052	3.2	0.193	0.075	0.194	0.022	4
Reference set # 6	0.517	0.320	0.482	0.281	4.4	0.321	0.150	0.294	0.108	6
Reference set # 7	0.428	0.227	0.361	0.204	4.4	0.332	0.172	0.120	0.141	6
Reference set # 8	0.320	0.112	0.280	0.116	2.9	0.244	0.064	0.205	0.070	5
Reference set # 9	0.331	0.158	0.316	0.113	3.5	0.308	0.106	0.079	0.110	5
Average	**0.389**	**0.177**	**0.354**	**0.158**	**3.4**	0.293	0.112	0.231	0.0909	5

The time of NSGAII is in seconds, whereas it is in minutes for SPEA2. Based on the results appeared in Table 4.12, NSGAII surpasses SPEA2 in both time and ROUGE results.

4.4.6 The effect of Score as an Objective Function

In order to study the effect of adding the score as an objective function, an experiment with coverage and diversity only is conducted, the results are shown in Table 4.13.

Table 4.13: The F-measure of 5-Independent runs with and without the score.

	Coverage and Diversity				Coverage, Diversity, and Sore			
Reference set	**R-1**	**R-2**	**R-L**	**R-SU4**	**R-1**	**R-2**	**R-L**	**R-SU4**
Reference set # 1	0.170	0.019	0.164	0.035	0.292	0.029	0.285	0.091
Reference set # 2	0.309	0.118	0.302	0.107	0.405	0.199	0.402	0.158
Reference set # 3	0.423	0.191	0.328	0.138	0.510	0.279	0.390	0.179
Reference set # 4	0.315	0.079	0.296	0.099	0.458	0.153	0.435	0.231
Reference set # 5	0.206	0.086	0.206	0.039	0.234	0.116	0.234	0.052
Reference set # 6	0.321	0.151	0.294	0.108	0.517	0.320	0.482	0.281
Reference set # 7	0.332	0.172	0.120	0.141	0.428	0.227	0.361	0.204
Reference set # 8	0.246	0.059	0.207	0.069	0.320	0.112	0.280	0.116
Reference set # 9	0.203	0.030	0.182	0.062	0.331	0.158	0.316	0.113
Average	0.278	0.078	0.263	0.086	**0.389**	**0.177**	**0.354**	**0.158**

Also, the relative improvements of adding the score of a sentence as objective function are

reported in Table 4.14.

Table 4.14: The relative improvements of adding the score as an objective function.

System	**R-1**	**R-2**	**R-L**	**R-SU4**
Coverage and diversity only	0.27845	0.07816	0.26262	0.08639
Coverage, diversity, and score	0.38853	0.17119	0.354041	0.15842
Improvement of using the score as objective function	+39.5%	+119.0%	+34.8%	+83.4%

The relative improvement is calculated using the following equation:

$$Relative - Improvements = \frac{score(new - system) - score(old - system)}{score(old - system)} * 100 \quad (4.5)$$

It is clear that adding the score as an objective function affects highly the results, where we achieve a relative improvement equals to 39.5%, 119.0%, 34.8%, and 83.4% with Rouge-1, Rouge-2, Rouge-L, Rouge-SU4 respectively.

4.4.7 The Effect of Cutting the Search Space Based on the Score Function

Also, a set of experiments is performed to study the effect of cutting the search space based on the score function, so different cutting ratios were tested and the average value of each cutting ratio is reported in Table 4.15.

Table 4.15: The F-measure of 5-Independent runs with different cutting ratios.

Cutting Ratio	**R-1**	**R-2**	**R-L**	**R-SU4**
No-Cutting	**0.38853**	**0.17687**	**0.35404**	**0.15842**
10%	0.31313	0.12058	0.26078	0.09200
20%	0.28745	0.11376	0.26752	0.09033
30%	0.26822	0.09544	0.24574	0.07540
40%	0.27574	0.09529	0.24633	0.07520
50%	0.26462	0.08862	0.24385	0.06696

It is clear that when the cutting ratio increases the rouge results decrease. The reason behind the above results can be explained as follows: depending on the score only can retrieve the important sentences but it may be redundant and don't cover the different aspects appeared

in the original documents. This proves the effectiveness of the multi-objective optimization approach with score, coverage, and diversity objective functions simultaneously.

4.4.8 Discussion

In this section, we discuss the results presented in the previous section and compare them with the results of the systems that appear in the literature which are shown in Table 4.16.

Table 4.16: Systems participated with DUC-2002 and TAC 2011 datasets.

ID	System	Approach	Features	Dataset
ID1	Baseline [55]	Clustering-based approach	-	TAC 2011
ID2	Global and Local Models for Multi-Document Summarization [80]	Unsupervised models for latent structure discovery	TF-ISF weighting	TAC 2011
ID3	The CIST Summarization System at TAC 2011 [81]	hierarchical Latent Dirichlet Allocation (hLDA)	Title similarity, keywords, name entity, sentence coverage, and word abstractive level	TAC 2011
ID4	LIF at TAC Multiling: Towards a Truly Language Independent Summarizer Arabic/English multi-document summarization with CLASSYthe past and the future [82]	Maximal Marginal Relevance (MMR)	TF-ISF weighting	TAC 2011
ID5	University of Essex at the TAC 2011 Multilingual Summarisation Pilot [83]	Clustering-based approach	TF-ISF weighting	TAC 2011
ID6	Arabic/English multi-document summarization with CLASSYthe past and the future [19]	Clustering, Linguistics, and Statistics for Summarization Yield (CLASSY)	The signature terms, and the probability of a term occurs in the sentences	TAC 2011
ID7	Guided and Multilingual Summarization Tasks [84]	LSA-based summarizer	TF weighting	TAC 2011
ID8	Ant Colony System for Multi-Document Summarization [55]	Ant Colony optimization algorithm to maximize the summary coverage	TF-ISF weighting with PageRank and HITS ranking	TAC 2011
ID9	Topline [55]	Genetic algorithm	-	TAC 2011
ID1	Multi-document Arabic Text Summarisation [9]	Cluster-based summrization	Similarity using three models: VSM, LSA, and Dice	DUC 2002
ID2	Automatic Multi-Document Arabic Text Summarization Using Clustering and Keyphrase Extraction [17]	Combined clustering method to group the documents into clusters	Sentence count, TF, First/last occurrence in text, and C-value	DUC 2002

The systems that participated with TAC 2011 MultiLing Pilot dataset and their results are shown

in Table 4.17. Moreover, the relative improvement (%) is computed using equation 4.5 and appended to the table of results.

Table 4.17: The F-measure values of ROUGE results and the relative improvement of the participating systems with TAC 2011 MultiLing Pilot dataset.

System	R-1	R-2	R-L	R-SU4	R-1	R-2	R-L	R-SU4
ID1	0.231	0.095	0.212	0.097	+68.4%	+80%	+66.98%	+62.89%
ID2	0.224	0.086	0.214	0.099	+73.66%	+98.84%	+65.42%	+59.6%
ID3	0.232	0.089	0.220	0.099	+67.67%	+92.13%	+60.91%	+59.6%
ID4	0.263	0.086	0.239	0.107	+47.91%	+98.84%	+48.12%	+47.66%
ID5	0.268	0.097	0.248	0.115	+45.15%	+76.29%	+42.74%	+37.39%
ID6	0.292	0.103	0.273	0.133	+33.22%	+66.02%	+29.67%	+18.8%
ID7	0.300	0.128	0.272	0.151	+29.67%	+33.59%	+30.15%	+4.64%
ID8	0.308	0.149	0.269	0.155	+26.3%	+14.77%	+31.6%	+1.94%
ID9	0.312	0.120	0.284	0.130	+24.68%	+42.5%	+24.65%	+21.54%
Stanford + Lemma	0.317	0.101	0.297	0.104	+22.71%	+69.31%	+19.19%	+51.92%
Code + Lemma	0.346	0.129	0.311	0.125	+12.43%	+32.56%	+13.83%	+26.4%
Stanford + Khoja	0.355	0.117	0.317	0.128	+9.58%	+46.15%	+11.67%	+23.44%
Code + Khoja	**0.389**	**0.171**	**0.354**	**0.158**	-	-	-	-

ROUGE-1 results of the systems participating with DUC-2002 dataset are reported and compared to our results in Table 4.18.

Table 4.18: ROUGE-1 results of the systems participating with DUC-2002 dataset

System	Recall	Precision	F-measure	Recall	Precision	F-measure
ID1	0.395	0.384	0.390	+15.75%	+26.02%	+20.64%
ID2	0.452	0.418	0.434	+1.15%	+15.77%	+8.41%
DUC 2002 + Lemma	0.352	0.391	0.372	+29.89%	+23.76%	+26.48%
DUC 2002 + Khoja	**0.4572**	**0.4839**	**0.4705**	-	-	-

It is clear from the results in the previous section that preprocessing affects highly the summarization process where Khoja stemmer with punctuation marks outperforms lemma stemmer and Stanford tokenizer. Punctuation marks tokenizer is more stable when the data is written with correct usage of punctuation marks, while Stanford tokenizer is more effective if the data is written as long line without punctuation marks. In our case, TAC 2011 corpus is written with full usage of punctuation marks, so this explains why punctuation marks tokenizer outperforms

Stanford tokenizer. On the other hand, Khoja stemmer beats lemma stemmer. This can be explained as follows: Khoja stemmer is a root-based stemmer which retrieves the root of a word, and this increases the semantic similarity between sentences.

Moreover, NSGAII outperforms SPEA2 algorithm in terms of rouge results and time, where the average running time of NSGAII is very small approximately seconds compared to SPEA2 which takes 5 minutes on average. Furthermore, the ROUGE results of NSAGII higher than SPEA2. In addition, Table 4.11 illustrates that ROUGE results decrease when the population size increases. In summary, NSGAII with population size equals 50, Binary Tournament Selection, crossover Probability = 0.9, Max Iterations = 25000, and mutation probability = $\frac{1.0}{Total\#ofBits}$ are the best controlling parameters of the optimization process.

Also, one of the main findings based on the results shown in Table 4.13, adding the score as an objective function shifted the rouge results significantly. Moreover, cutting the search space is not effective enough where the rouge results decrease when the cutting ratio increases. The reason of this can be explained as follows: depending on the score only can retrieve the important sentences but it may be redundant and don't cover the different aspects appeared in the original documents. This proves the effectiveness of the multi-objective optimization approach with score, coverage, and diversity objective functions simultaneously.

Tables 4.17 and 4.18 show that our approach results outperform all systems participating with TAC 2011 and DUC-2002. Results of punctuation marks (Code) and Khoja stemmer outperform all participating systems in terms of all ROUGE metrics. In addition, our results outperform all systems in terms of ROUGE-1 and ROUGE-L with other tokenizers and stemmers. It is worth mentioning that our approach performance is better than other peer systems, which it is clear from ROUGE-2 results which is bi-gram matching. Also, ROUGE-L which is calculated by counting the main in-sequence words. These two measures are better than other peer systems and indicate that our summaries are closer to the gold-standard summaries. Our system showed improvements of +24.68%, +42.5%, +24.65%, and +21.54% over the top-ranked system in terms of Rouge-1, Rouge-2, Rouge-L, and Rouge-SU4 respectively.

Also, with DUC 2002 dataset, the results of Khoja stemmer outperform Lemma stemmer. Moreover, our system beats all systems participating with this dataset in terms of all ROUGE metrics

which are shown in Tables 4.18. Our system showed improvements of +1.15%, +15.77%, +8.41% over the top-ranked system in terms of recall, precision, and F-measure of Rouge-1 respectively. Figure 4.3 and 4.4 show a comparison of our system to the most recent related work with both datasets.

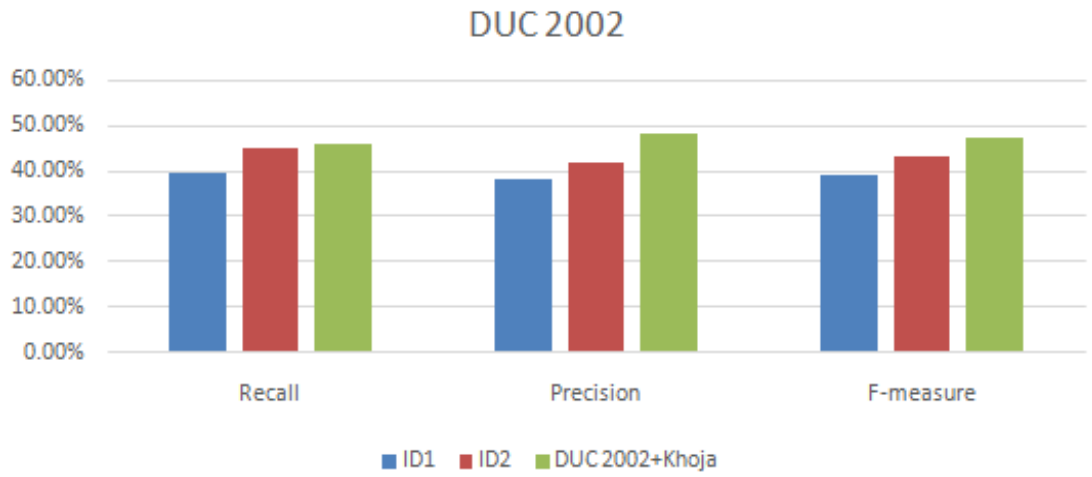

Figure 4.3: Comparison of DUC 2002 results.

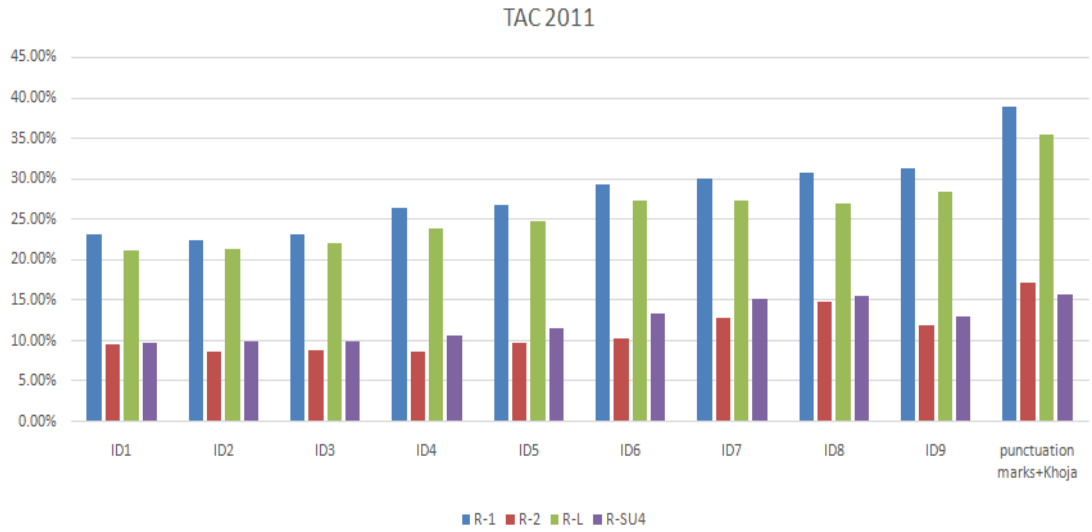

Figure 4.4: Comparison of TAC 2011 results.

Chapter 5 Conclusion, Future work and Limitations

5.1 Conclusion

In this thesis, we studied the effect of preprocessing in Arabic text summarization. where punctuation marks tokenizer beats Stanford tokenizer, and Khoja stemmer outperforms lemma stemmer. Also, we addressed the text summarization as a multi-objective optimization problem with three objective functions (coverage, score, diversity) to generate a relevant summary that covers all topics appear in the original set of documents.

Moreover, semantic features are considered by using the Arabic WordNet to find synonyms of terms. Subsequently, similarity measures and text matching are performed based on the existence of a term or its synonyms. Also, silhouette method is used to find the optimal number of clusters which will be used in the k-medoids clustering algorithm, this parameter is very important since it represents the number of different topics appear in the original set of documents. The ROUGE results showed the effectiveness of considering Arabic multi-document summarization as multi-objective optimization problem.

Furthermore, this work will be the first research that treated the Arabic multi-document summarization as multi-objective optimization. Indeed, our research targets the Arabic language, since most of summarization systems are available for English language and other European languages. Despite that our approach was tested with the Arabic language only, but it is language dependent. The approach will work perfectly by changing the used tools during the feature extraction to the desired language.

The performance of the proposed system is evaluated using TAC 2011 and DUC 2002. The experimental results using the ROUGE evaluation measure show the effectiveness of our system compared to the state of the art systems. With TAC 2011, our system outperforms other peer systems with all ROUGE metrics, we achieved an F-measure of 38.9%, 17.7%, 35.4%, and 15.8% for Rouge-1, Rouge-2, Rouge-L, and Rouge-SU4 respectively. Moreover, our system with DUC 2002 dataset achieves an F-measure of 47.1%, 23.7%, 47.1%, 20.4% for Rouge-1, Rouge-2, Rouge-L, and Rouge-SU4 respectively.

5.2 Future Work and Limitations

We faced many problems during data collection and preparation. The process of requesting the data was really long and time-consuming. Also, the Arabic version of DUC 2002 was not available, so we had to extract and translate the articles. During this process, many files and gold summaries were corrupted, and the translation was not adequate enough, so we just tested the main approach and the effect of stemming on the output summary and didn't rely on this data for further experimentations. Moreover, we have a variety of tools used in the pre-processing stages that harden the automation of the results generation.

As future work, we are planning to automate all the system stages with one click from the reading of input documents to the generation of the ROUGE results. Also, we can handle the problems occurred with DUC 2002 during the translation and the preprocessing, and upload the Arabic version online for research use. Also, we can work on the coherence objective function as post-processing stage to improve the quality of the generated summary to be more readable. Moreover, we will try other stopping criterion such as the time and compare the solutions from the different algorithms.

Bibliography

[1] Hu, Y. H., Chen, Y. L., & Chou, H. L. (2017). Opinion mining from online hotel reviews - A text summarization approach. Information Processing & Management, 53(2), 436-449.

[2] Lagrini, S., Redjimi, M., & Azizi, N. (2017). Automatic Arabic Text Summarization Approaches. International Journal of Computer Applications, 164(5).

[3] Ferreira, R., de Souza Cabral, L., Lins, R.D., e Silva, G.P., Freitas, F., Cavalcanti, G.D., Lima, R., Simske, S.J. and Favaro, L. (2013). Assessing sentence scoring techniques for extractive text summarization. Expert systems with applications, 40(14), 5755-5764.

[4] Meena, Y. K., & Gopalani, D. (2016). Efficient Voting-Based Extractive Automatic Text Summarization Using Prominent Feature Set. IETE Journal of Research, 62(5), 581-590.

[5] Al-Saleh, A. B., & Menai, M. E. B. (2016). Automatic Arabic text summarization: a survey. Artificial Intelligence Review, 45(2), 203-234.

[6] Al Harazin, K. S. (2015). Multi-document arabic text summarization. Multi-document Arabic Text Summarization.

[7] Kalyanmoy, D. (2001). Multi objective optimization using evolutionary algorithms (pp. 124-124). John Wiley and Sons.

[8] Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering & System Safety, 91(9), 992-1007.

[9] El-Haj, M., Kruschwitz, U., & Fox, C. (2011, July). Multi-document Arabic text summarisation. In Computer Science and Electronic Engineering Conference (CEEC), 2011 3rd (pp. 40-44). IEEE.

[10] Alguliev, R. M., & Aliguliyev, R. M. (2005, September). Effective summarization method of text documents. In Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on (pp. 264-271). IEEE.

[11] Conroy, J. M., & Schlesinger, J. D. (2008). CLASSY and TAC 2008 Metrics. In TAC.

[12] Morita, H., Sakai, T., & Okumura, M. (2012). Query snowball: a co-occurrence-based approach to multi-document summarization for question answering. Information and Media Technologies, 7(3), 1124-1129.

[13] Gupta, V., & Lehal, G. S. (2010). A survey of text summarization extractive techniques. Journal of emerging technologies in web intelligence, 2(3), 258-268.

[14] Kumar, Y. J., & Salim, N. (2012). Automatic multi document summarization approaches. In KS Gayathri, Received BE degree in CSE from Madras University in 2001 and ME degree from Anna University, Chennai. She is doing Ph. D. in the area of Reasoning in Smart.

[15] Radev, D. R., Hovy, E., & McKeown, K. (2002). Introduction to the special issue on summarization. Computational linguistics, 28(4), 399-408.

[16] Al-Taani, A. T., & Al-Omour, M. M. (2014). An extractive graph-based Arabic text summarization approach. In The International Arab Conference on Information Technology, Jordan.

[17] Fejer, H. N., & Omar, N. (2015). Automatic multi-document Arabic text summarization using clustering and keyphrase extraction. Journal of Artificial Intelligence, 8(1), 1-9.

[18] Haboush, A., Al-Zoubi, M., Momani, A., & Tarazi, M. (2012). Arabic text summarization model using clustering techniques. World of Computer Science and Information Technology Journal (WCSIT) ISSN, 2221-0741.

[19] Schlesinger, J. D., Oleary, D. P., & Conroy, J. M. (2008, February). Arabic/English multi-document summarization with CLASSYthe past and the future. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 568-581). Springer, Berlin, Heidelberg.

[20] Radev, D. R., Jing, H., Sty, M., & Tam, D. (2004). Centroid-based summarization of multiple documents. Information Processing & Management, 40(6), 919-938.

[21] Sarkar, K. (2009). Sentence clustering-based summarization of multiple text documents. TECHNIAInternational Journal of Computing Science and Communication Technologies, 2(1), 325-335.

[22] Jain, H. J., Bewoor, M. S., & Patil, S. H. (2012). Context Sensitive Text Summarization Using K Means Clustering Algorithm. International Journal of Soft Computing and Engineering, 2(2).

[23] Waheeb, S. A., & Husni, H. (2014). Multi-Document Arabic Summarization Using Text Clustering to Reduce Redundancy. International Journal of Advances in Science and Technology (IJAST), 2(1), 194-199.

[24] Keskes, I. (2015). Discourse analysis of arabic documents and application to automatic summarization (Doctoral dissertation, Universit de Toulouse, Universit Toulouse III-Paul Sabatier).

[25] AlSanie, W., Touir, A., & Mathkour, H. (2005). Towards an infrastructure for Arabic text summarization using rhetorical structure theory. Master's Thesis. King Saud University, Riyadh, Saudi Arabia.

[26] Azmi, A. M., & Al-Thanyyan, S. (2012). A text summarizer for Arabic. Computer Speech & Language, 26(4), 260-273.

[27] Mathkour, H. I. (2009). A Novel rhetorical structure approach for classifying Arabic security documents. International Journal of Computer Theory and Engineering, 1(3), 195.

[28] Ibrahim, A., Elghazaly, T., & Gheith, M. (2013). A novel Arabic text summarization model based on rhetorical structure theory and vector space model. International Journal of Computational Linguistics and Natural Language Processing, 2(8), 480-485.

[29] Fattah, M. A., & Ren, F. (2009). GA, MR, FFNN, PNN and GMM based models for automatic text summarization. Computer Speech & Language, 23(1), 126-144.

[30] Belkebir, R., & Guessoum, A. (2015). A supervised approach to arabic text summarization using adaboost. In New Contributions in Information Systems and Technologies (pp. 227-236). Springer, Cham.

[31] Al-Radaideh, Q. A., & Bataineh, D. Q. (2018). A Hybrid Approach for Arabic Text Summarization Using Domain Knowledge and Genetic Algorithms. Cognitive Computation, 1-19.

[32] Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In Mining text data (pp. 43-76). Springer, Boston, MA.

[33] Al-Khawaldeh, F., & Samawi, V. (2015). Lexical cohesion and entailment based segmentation for arabic text summarization (lceas). The World of Computer Science and Information Technology Journal (WSCIT), 5(3), 51-60.

[34] Tatar, D., Mihis, A., Lupsa, D., & Tamaianu-Morita, E. (2009). Entailment-based linear segmentation in summarization. International Journal of Software Engineering and Knowledge Engineering, 19(08), 1023-1038.

[35] Imam, I., Nounou, N., Hamouda, A., & Khalek, H. A. A. (2013). An ontology-based summarization system for arabic documents (ossad). Int. J. Comput. Appl, 74(17), 38-43.

[36] Al-Sabahi, K., Zhang, Z., Long, J., & Alwesabi, K. (2018). An Enhanced Latent Semantic Analysis Approach for Arabic Document Summarization. Arabian Journal for Science and Engineering, 1-16.

[37] Umam, K., Putro, F. W., Pratamasunu, G. Q. O., Arifin, A. Z., & Purwitasari, D. (2015). Coverage, diversity, and coherence optimization for multi-document summarization. Jurnal Ilmu Komputer dan Informasi, 8(1), 1-10.

[38] Chen, K. Y., Liu, S. H., Chen, B., & Wang, H. M. (2016, March). Improved spoken document summarization with coverage modeling techniques. In Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on (pp. 6010-6014). IEEE.

[39] Li, L., Zhou, K., Xue, G. R., Zha, H., & Yu, Y. (2009, April). Enhancing diversity, coverage and balance for summarization through structure learning. In Proceedings of the 18th international conference on World wide web (pp. 71-80). ACM.

[40] Cai, X., Li, W., & Zhang, R. (2014). Enhancing diversity and coverage of document summaries through subspace clustering and clustering-based optimization. Information sciences, 279, 764-775.

[41] Luo, W., Zhuang, F., He, Q., & Shi, Z. (2013). Exploiting relevance, coverage, and novelty for query-focused multi-document summarization. Knowledge-Based Systems, 46, 33-42.

[42] Nayeem, M. T., & Chali, Y. (2017). Extract with Order for Coherent Multi-Document Summarization. arXiv preprint arXiv:1706.06542.

[43] Huang, L., He, Y., Wei, F., & Li, W. (2010, April). Modeling document summarization as multi-objective optimization. In Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on (pp. 382-386). IEEE.

[44] Wu, Z., Lei, L., Li, G., Huang, H., Zheng, C., Chen, E., & Xu, G. (2017). A topic modeling based approach to novel document automatic summarization. Expert Systems with Applications, 84, 12-23.

[45] Oufaida, H., Nouali, O., & Blache, P. (2014). Minimum redundancy and maximum relevance for single and multi-document Arabic text summarization. Journal of King Saud University-Computer and Information Sciences, 26(4), 450-461.

[46] Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation, 1(1), 32-49.

[47] Peyrard, M., & Eckle-Kohler, J. (2016). A General Optimization Framework for Multi-Document Summarization Using Genetic Algorithms and Swarm Intelligence. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (pp. 247-257).

[48] Alguliev, R. M., Aliguliyev, R. M., Hajirahimova, M. S., & Mehdiyev, C. A. (2011). MCMR: Maximum coverage and minimum redundant text summarization model. Expert Systems with Applications, 38(12), 14514-14522.

[49] Al-Abdallah, R. Z., & Al-Taani, A. T. (2017). Arabic Single-Document Text Summarization Using Particle Swarm Optimization Algorithm. Procedia Computer Science, 117, 30-37.

[50] Alguliev, R. M., Aliguliyev, R. M., & Hajirahimova, M. S. (2012). GenDocSum+ MCLR: Generic document summarization based on maximum coverage and less redundancy. Expert Systems with Applications, 39(16), 12460-12473.

[51] Jung, C., Datta, R., & Segev, A. (2017, July). Multi-document summarization using evolutionary multi-objective optimization. In Proceedings of the Genetic and Evolutionary Computation Conference Companion (pp. 31-32). ACM.

[52] Saleh, H. H., & Kadhim, N. J. (2016). Extractive multi-document text summarization using multi-objective evolutionary algorithm based model. Iraqi Journal of Science, 57(1C), 728-741.

[53] Sanchez-Gomez, J. M., Vega-Rodrguez, M. A., & Prez, C. J. (2018). Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach. Knowledge-Based Systems, 159, 1-8.

[54] Rautray, R., & Balabantaray, R. C. (2018). An evolutionary framework for multi document summarization using Cuckoo search approach: MDSCSA. Applied computing and informatics, 14(2), 134-144.

[55] Al-Saleh, A., & Menai, M. E. B. (2018). Ant Colony System for Multi-Document Summarization. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 734-744).

[56] Shardan, R., & Kulkarni, U. (2010). Implementation and evaluation of evolutionary connectionist approaches to automated text summarization.

[57] Meena, Y. K., Dewaliya, P., & Gopalani, D. (2015, February). Optimal features set for extractive automatic text summarization. In Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on (pp. 35-40). IEEE.

[58] Attia, M. A. (2007, June). Arabic tokenization system. In Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources (pp. 65-72). Association for Computational Linguistics.

[59] Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations (pp. 55-60).

[60] Ranks.nl. (2018). *Arabic*. [online] Available at: https://www.ranks.nl/stopwords/arabic [Accessed 6 Jan. 2018].

[61] Khoja, S., & Garside, R. (1999). Stemming arabic text. Lancaster, UK, Computing Department, Lancaster University.

[62] Mustafa, M., Eldeen, A. S., Bani-Ahmad, S., & Elfaki, A. O. (2017). A Comparative Survey on Arabic Stemming: Approaches and Challenges. Intelligent Information Management, 9(02), 39.

[63] Brahmi, A., Ech-Cherif, A., & Benyettou, A. (2013). An arabic lemma-based stemmer for latent topic modeling. Int. Arab J. Inf. Technol., 10(2), 160-168.

[64] Ayedh, A., Tan, G., Alwesabi, K., & Rajeh, H. (2016). The effect of preprocessing on arabic document categorization. Algorithms, 9(2), 27.

[65] Meena, Y. K., & Gopalani, D. (2014, November). Analysis of sentence scoring methods for extractive automatic text summarization. In Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies (p. 53). ACM.

[66] Reynolds, A. P., Richards, G., & Rayward-Smith, V. J. (2004, August). The application of k-medoids and pam to the clustering of rules. In International Conference on Intelligent Data Engineering and Automated Learning (pp. 173-178). Springer, Berlin, Heidelberg.

[67] Lin, C., Qing, A., & Feng, Q. (2011). A comparative study of crossover in differential evolution. Journal of Heuristics, 17(6), 675-703.

[68] Brockhoff, D. (2015, March). A bug in the multiobjective optimizer ibea: Salutary lessons for code release and a performance re-assessment. In International Conference on Evolutionary Multi-Criterion Optimization (pp. 187-201). Springer, Cham.

[69] Nebro, A. J., Durillo, J. J., Luna, F., Dorronsoro, B., & Alba, E. (2009). MOCell: A cellular genetic algorithm for multiobjective optimization. International Journal of Intelligent Systems, 24(7), 726-746.

[70] Fortin, F. A., & Parizeau, M. (2013, July). Revisiting the NSGA-II crowding-distance computation. In Proceedings of the 15th annual conference on Genetic and evolutionary computation (pp. 623-630). ACM.

[71] Zhang, Y., Harman, M., & Mansouri, S. A. (2007, July). The multi-objective next release problem. In Proceedings of the 9th annual conference on Genetic and evolutionary computation (pp. 1129-1137). ACM.

[72] Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. TIK-report, 103.

[73] Narasimhamurthy, A. M. (2003, June). A framework for the analysis of majority voting. In Scandinavian Conference on Image Analysis (pp. 268-274). Springer, Berlin, Heidelberg.

[74] El-Haj, M., Kruschwitz, U., & Fox, C. (2009, November). Experimenting with automatic text summarisation for arabic. In Language and Technology Conference (pp. 490-499). Springer, Berlin, Heidelberg.

[75] Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., ... & Roth, R. (2014, May). MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In LREC (Vol. 14, pp. 1094-1101).

[76] Althobaiti, M., Kruschwitz, U., & Poesio, M. (2014). Aranlp: A java-based library for the processing of arabic text.

[77] R-project.org. (2018). *R: What is R?*. [online] Available at: https://www.r-project.org/about.html [Accessed 5 Jan. 2018].

[78] Nebro, A. J., & Durillo, J. J. (2011). jMetal: A Java Framework for Multi-Objective Optimization. Advances in Engineering Software, vol. 42, pp. 760-771.

[79] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 6(2), 182-197.

[80] Das, P., & Srihari, R. K. (2011). Global and Local Models for Multi-Document Summarization. In TAC.

[81] Liu, H., Ping'an Liu, W. H., Heng, W., & Li, L. (2011). The CIST Summarization System at TAC 2011. In TAC.

[82] Hmida, F., & Favre, B. (2011). LIF at TAC MultiLing: Towards a Truly Language Independent Summarizer. In TAC.

[83] El-Haj, M., Kruschwitz, U., & Fox, C. (2011). University of Essex at the TAC 2011 multilingual summarisation pilot.

[84] Steinberger, J., Kabadjov, M. A., Steinberger, R., Tanev, H., Turchi, M., & Zavarella, V. (2011). JRC's Participation at TAC 2011: Guided and MultiLingual Summarization Tasks. TAC, 11, 1-9.