People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Batna 2

Faculty of Mathematics and Computer Science

Department of Computer Science

T H E S I S

Submitted in fulfilment of the requirements

for the degree of Doctor of Science in Computer Science

Title

Automatic multi-documents text summarization using Binary Biology Migration Algorithm

Defended by

Mohamed BOUSSALEM

Publicly defended on: 22/05/2024

Committee members:

President:	Pr. Ali Behloul	University of Batna 2
Supervisor:	Pr. Hamouma Moumen	University of Batna 2
Co-supervisor:	Pr. Samia Aitouche	University of Batna 2
Examinators :	Pr. Kamal Amroun	University of Béjaïa
	Dr. Leila Boussaad	University of Batna 1
	Dr. Makhelouf Ledmi	University of Khenchela

Academic year: 2023/2024

Dedicated to my dear parents, brothers, sisters, wife, son Mouad, and daughters Hanin and Douha.

Acknowledgement

First and foremost, I thank the Almighty God for the health, the will, and for all the courage and patience he has given me during the period of this work.

I would like to express my gratitude to my supervisor Professor: Moumen Hemouma and my co-supervisor Professor: Samia Aitouche for their patience, continuous support, and guidance throughout this work.

I also thank all the members of the jury Pr. Ali Behloul, Pr. Kamal Amroun, Dr. Leila Boussaad, and Dr. Makhelouf Ledmi for accepting to review and evaluate this work.

Special thanks to Dr. Hichem Houassi, Dr. Hichem Rahab, and Dr. Abdelaali Bakhouche for their help, and knowledge.

I thank the ICOSI laboratory members and all those who helped me in some way for realizing this work.

Finally, my thanks go to my family and my wife for their support and encouragement.

Abstract

As the World Wide Web continues to expand, the process of identifying pertinent information within its vast volume of documents becomes increasingly challenging. This complexity necessitates the development of efficient solutions, one of which is automatic text summarization; an active research area dedicated to extracting key information from extensive text. The difficulties are further compounded when addressing multi-document text summarization, due to the diversity of topics and sheer volume of information. In response to this issue, this study introduces a novel approach based on swarm intelligence algorithm called biology migration algorithm (BMA). Our proposed approach is; Binary Biology Migration Algorithm for Multi-Document Summarization (BBMA-MDS). Viewing multi-document summarization as a combinatorial optimization problem, this approach leverages the biology migration algorithm to select an optimal combination of sentences. Evaluations of the proposed algorithm's performance are conducted using the ROUGE metrics, which facilitate a comparison between the automatically generated summary and the reference summary, commonly known as the 'gold standard summary'. For a comprehensive evaluation, the well-established DUC2002 and DUC2004 datasets are employed. The results demonstrate the superior performance of the BBMA-MDS approach when compared to alternative algorithms, including firefly and particle swarm optimization, as indicated by the selected metrics. This study thus contributes effectively according to the evaluation to the field by proposing BBMA-MDS as an effective solution for the multi-document text summarization problem.

Keywords

Text summarization, multi document text summarization, optimization, biology migration algorithm, swarm intelligence, ROUGE metrics.

Résumé

Au fur et à mesure que le World Wide Web se développe, il devient de plus en plus difficile d'identifier des informations pertinentes dans cette énorme quantité de documents. Cette difficulté exige l'élaboration de solutions efficaces, dont l'une est le résumé automatique de texte; un domaine de recherché actif dédié à l'extraction d'informations clés à partir d'un texte. Les difficultés s'aggravent lorsqu'il s'agit de résumer plusieurs documents (multi-documents), en raison de la diversité des sujets et le grand volume d'informations. En réponse à cette question, cette étude introduit une nouvelle approche basée sur un algorithme d'intelligence en essaim qui est; algorithme de migration biologique (BMA), notre approche est : algorithme binaire de migration biologique pour le résumé multi-document (BBMA-MDS). En considérant le résume multi-document comme un problème d'optimisation combinatoire, cette approche utilise l'algorithme de migration biologique pour sélectionner une combinaison optimale de phrases. Les évaluations de la performance de l'algorithme proposé ont effectuées à l'aide des mesures ROUGE, qui font une comparaison entre le résumé généré automatiquement et les résumés de référence, communément connu sous le nom de "résumés d'or". Pour une évaluation complète, les benchmarks DUC2002 et DUC2004 sont utilisés.

Les résultats des expérimentations montrent que l'approche BBMA-MDS a des performances supérieures par rapport aux algorithmes de la littérature, tels que Firefly et PSO Particle Swarm Optimization. Ainsi, cette étude apporte une contribution au domaine de résumé automatique de texte en suggérant BBMA-MDS comme une solution efficace au problème de résumé multi-documents.

Mots clés

Résumé de texte, résumé multi-documents de texte, optimisation, algorithme de migration de biologie, intelligence en essaim, métriques ROUGE.

ملخص

مع استمرار شبكة الويب العالمية في التوسع، تصبح عملية تحديد المعلومات ذات الأهمية ضمن الحجم الهائل من المستندات صعبة بشكل متزايد. يتطلب هذا التعقيد تطوير حلول فعالة، أحدها التلخيص التلقائي للنص؛ فهو مجال بحث نشط مخصص لاستخراج المعلومات الأساسية من نص واسع النطاق. وتتفاقم الصعوبات أكثر عند معالجة تلخيص نص متعدد المستندات، وذلك بسبب تنوع المواضيع وحجم المعلومات الهائل. وكحل لهذه المشكلة، تقدم هذه الدراسة منهجًا جديدًا يعتمد على خوارزمية هجرة الأحياءBMA Biology migration algorithm) (و هو واحد من خوارزميات الأسراب الذكية (swarm intelligence)،الحل المقترح هو الخوارزمية الثنائية لهجرة الأحياء لتلخيص المستندات المتعددة, binary biology migration algorithm for multi-document text summarization (BBMA-MDS).

باعتبار تلخيص المستندات المتعددة مشكلة تحسين اندماجي (combinatorial optimization)، فإن الحل المقترح يعتمد خوارزمية هجرة الأحياء لتحديد مجموعة من الجمل تشكل التلخيص المثالي. يتم إجراء تقييمات لأداء الخوارزمية المقترحة باستخدام مقاييسROUGE ، التي تسهل المقارنة بين الملخص الذي تم إنشاؤه تلقائيًا والملخص المرجعي، المعروف باسم "الملخص القياسي الذهبي". لإجراء تقييم شامل، يتم استخدام مجموعات البيانات DUC2002 و DUC2004 الواسعة الاستعمال في مجال التلخيص التلقائي للنص. توضح النتائج الأداء المتفوق لنهج BBMA-MDS عند مقارنته بالخوارزميات المقترحة في الدراسات السابقة، بما في ذلك Firefly و PSO Particle Swarm Optimization .وبالتالي تساهم هذه الدراسة في هذا المجال من خلال اقتراح BBMA-MDS كحل فعال لمشكلة تلخيص النص متعدد المستندات.

الكلمات المفتاحية

تلخيص النص، تلخيص نص متعدد المستندات، التحسين، خوارزمية هجرة الأحياء، الأسراب الذكية، مقاييس ROUGE

Contents

List of Figures

List of tables

General introduction

1. Context

The process of finding relevant information within the large volume of documents on the World Wide Web becomes increasingly difficult as it expands. This complication necessitates the development of effective solutions, one of which is automatic text summarization, an important research area dedicated to extracting key information from large amounts of text.

The aim of Automatic Text Summarization (ATS) systems is to extract or generate the most important and relevant information from the source text (Babar and Patil 2015), allowing readers to immediately understand the major points without having to read the entire document. ATS is the process of generating succinct and coherent summaries from lengthy pieces of text, such as articles, documents, or web pages, utilising computer algorithms and natural language processing (NLP) techniques. Borko and Bernier (Bernier. 1975) state that a summary has numerous advantages, including time-saving, facilitating selection and search, and enhancing indexing efficiency. Individual interpretations of a given document can yield varying summaries. This can arise from varying areas of focus or from each person's interpretation.

A lot of research have been conducted to advance the field of text summarization since Luhn's work (H. P. Luhn 1958). Early work in field dealt on single document summarization, in which methods created a summary of a single document, such as a news report, scientific article, TV show, or lecture. As research advanced, a new sort of summary problem emerged: multi document summarization (Nenkova and McKeown 2011).

The transition from automatic single-document summarization to automatic multi-document summarization signifies a notable advancement in the field of natural language processing (NLP) research. While the former focuses on condensing key information from a single document, the latter extends its scope to synthesize information from multiple documents.

In the realm of automatic single-document summarization, the primary goal is to distill essential points from a singular text, enabling users to quickly grasp the content without delving into the entire document. Techniques employed in this approach range from key phrase extraction to abstract generation based on linguistic models. Conversely, automatic

multi-document summarization faces a more intricate challenge, as it necessitates aggregating pertinent information from several sources.

2. Challenges of Multi-document summarization

Despite progress, multi-document summarization confronts various obstacles that add to the task's complexity (Goldstein et al. 2000). The variety of information sources is a considerable challenge. The documents in a multi-document context frequently differ widely in terms of content, style, and perspective. Combining this extensive mix of facts into a short summary presents a significant difficulty.

Another problem stems from the requirement to appropriately resolve redundancy. When dealing with many documents on the same subject, redundancy is inevitable. Redundant information might dilute the summary's quality and decrease its informativeness. Developing strategies to discover and minimise redundancy while preserving key content is an ongoing challenge.

Achieving clarity and cohesion in the summary presents an additional challenge. Advanced approaches are required to preserve a logical flow and coherence in the summary due to the variability of source documents. Achieving a balance between inclusion of diverse perspectives and maintaining a unified theme in the summary is a delicated task.

Scalability poses a significant obstacle in multi-document summarization, especially when confronted with a substantial number of documents. The computational complexity escalates proportionally with the amount of source documents, imposing limitations on processing time and resource demands. Effective algorithms and scalable approaches are crucial when handling large document collections.

Furthermore, evaluation metrics for multi-document summarization pose a challenge. Developing robust and comprehensive evaluation criteria that capture the nuances of summarizing information from multiple documents remains an open research question. Ensuring that automated evaluation methods align with human judgment in assessing the quality of multi-document summaries is crucial for advancing the field (Republic 2009).

In summary, addressing the challenges of diversity, redundancy, coherence, scalability, and evaluation metrics is paramount for the continued progress of multi-document summarization systems.

3. Motivation and aim of the thesis

Multi-document summarization systems are created to satisfy various needs in information management and document processing. These systems provide time-saving advantages by extracting crucial information from many documents at the same time, assisting in the efficient management of large amounts of data (Bernier. 1975). Multi-document summaries offer a brief and comprehensive summary, allowing efficient retrieval of pertinent information, which is particularly beneficial in decision-making contexts (Mani et al. 2002). They contribute in reducing cognitive stress, especially for users who are managing a substantial amount of documents. Moreover, these systems are used in research to accelerate the review of pertinent academic or scientific documents.

In addition to the previously outlined motivations, our work aim to further significance in diverse practical applications.

Firstly, we aim to offer a valuable tool for scientific reviews, streamlining the process and minimizing the efforts required by scientists in preparing the state of the art of a certain domain of research.

Moreover, the improved summaries generated by our algorithm have real-world implications, enhancing the efficiency and precision of business intelligence in enterprises for decision support. They help in inhibiting the tremendous wasted times in repeated contents in web pages or offline documents to take advantages from emergent knowledge and technologies efficiently for the benefit of business and research institutions .

Beyond business applications, our work extends to content analysis on social media platforms, providing insights into social trends, if needed, to offer more efficiency to commercial plans of businesses according to the preferences of customers and provide summaries to recommender systems of customers in all levels of purchases.

Additionally, the algorithm's capabilities are poised to assist journalists in efficiently preparing brief yet comprehensive news summaries without biases and in time.

4. Contributions

In this study, we introduce a new extractive approach for multi-document text summarization called: Binary Biology Migration Algorithm for Multi-document text summarization, BBMA-

MDS, based on the swarm intelligence algorithm, BMA. This approach considers MDS as an optimization problem. Our key contributions are highlighted as follows:

1. Innovative application of biology migration algorithm (BMA):

In our knowledge, for the first time, we utilize the Biology Migration Algorithm (BMA) to address MDS problems. This pioneering use of BMA expands the repertoire of swarm intelligence approaches for tackling challenges in multi-document summarization.

2. Enhanced binarization through Sigmoid Transformation:

Unlike the existing simplistic binarization methods in the literature, our approach introduces a novel enhancement. For the first time, we employ a sigmoid function to improve the discretization of the continuous aspect of classical BMA. This function converts real numbers ranging between 0 and 1 into discrete values, mapping them to either 0 or 1.

3. Introduction of a Comprehensive Objective Function:

We propose a novel objective function specifically crafted to evaluate the quality of extractive summaries. This function incorporates three critical features; coverage, cohesion, and readability. Significantly, the corresponding parameters or weights are not arbitrarily chosen; rather, they are meticulously determined to be the most appropriate ones for yielding relevant and meaningful summarization results. This meticulous parameter attribution contributes to the precision and effectiveness of our proposed algorithm.

5. Thesis structure

The remaining parts of this thesis are organised as follows:

The first chapter, which is both introductory and bibliographic in nature, concentrates on the field of automatic text summarization. We defined text summarization, classified summarization systems, investigated the various approaches used to automatically generate a summary, and concluded with a section on methods for evaluating automatic summarization systems and the associated evaluation campaigns.

Chapter 2 provides a state of the art of the various approaches commonly used to address the multi-document text summarization problem. We delve into the major steps of the multi-document automatic summarization process, starting with the pre-processing phase and moving through the intermediate representation or criteria extraction phase. We emphasize

the crucial phase of utilizing these criteria to generate the summary, detailing traditional approaches (statistical and graph-based), methods employing machine learning, and finally, optimization methods. We extensively analyze a significant number of works in this last category and conclude this chapter with an in-depth discussion on all these methods.

Chapter three provides an overview of metaheuristics for solving challenging optimisation problems. We will discuss the fundamental concepts of metaheuristics, including their definition; type of problems that necessitate the use of metaheuristics, intensification and diversification. Additionally, we will explore the two types of metaheuristics: approaches based on iterative development of a unique solution and approaches based on a population of solutions. We have highlighted techniques that employ swarm intelligence for the optimization problem.

The fourth chapter will be divided into 2 parts; the first one will present our proposed approach to tackle the problem of multi-document text summarization. This section starts with the modelling of MDS as a combinatorial optimisation problem, followed by a presentation of the original BMA algorithm. Then comes the step of adapting the algorithm BMA to our problem in order to produce the summary. The second part presented the experimental results of our approach on the two datasets Duc2002 and Duc2004, as well as a comparison with other approaches. The chapter concludes with a discussion of the results achieved by our approach and the interpretation of the results of comparisons with state-of-the-art approaches.

The final part of this thesis is a general conclusion that summarises some potential challenges, limitations of current work, and future research.

Chapter 1
Text summarization

1. Introduction

Automatic text summarization is considered one of the most important applications of natural language processing (NLP). Its primary function is to generate a brief, informative summary by extracting essential content from lengthy texts. Its aim is to assist humans in processing voluminous texts more efficiently and in short time. We typically read articles and documents and write our summaries manually. Given the time-consuming nature of this task, transitioning to automated summarization methods becomes imperative.

There are significant applications for text summarization in various NLP related tasks, such as text classification, question answering, legal document summarization, news summarization, and headline generation. Furthermore, summary generation can be integrated into these systems as an intermediate step to reduce the document's length.

This chapter will go over the essential concepts of automatic text summarization. This involves defining text summarising, categorising automatic text summarization systems based on various criteria, investigating several ways for evaluating these systems, and discussing the companies involved in their evaluation.

2. Definition

Automatic Text Summarization (ATS) systems aim to extract or generate the most important and relevant information from the source text (Babar and Patil 2015), allowing users to quickly grasp the key points without reading the entire document. ATS refers to the process of using computer algorithms and natural language processing (NLP) techniques to generate concise and coherent summaries from longer pieces of text, such as articles, documents, or web pages.

3. Classification of text summarization systems

The classification of automatic summaries can vary based on the criteria used. This is why automatic summaries do not exclusively belong to a single category, which means that a summary can belong to multiple categories. We can classify the summary into at least three main categories (E. Hovy and Lin 1998). These categories are based on three main factors: factors related to the source document, factors related to the purpose, and factors related to the final summary (Karen 1999)(E. Hovy and Lin 1998). Figure 1 illustrates these different categories of classification factors.

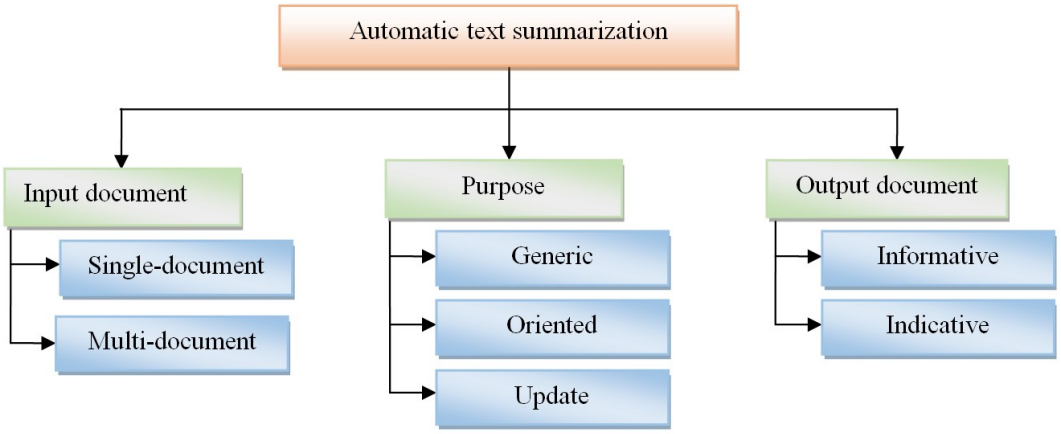

Figure 1: Classification of text summarization systems

3.1. Input based classification

Automatic Text Summarization Systems (ATS) can be divided into several classes based on the number of documents that need to be summarised. Thus, ATS systems can be classified as single-document summarization systems or multi-document summarization systems.

3.1.1. Single-Document Summaries

In the field of single-document text summarization, the focus is on processing a single document as input and generating a concise summary as an output (Gambhir and Gupta 2017),(Alzuhair and Al-Dhelaan 2019). A notable approach in this context was introduced by Thomas and al.(Thomas, Bharti, and Babu 2016), the proposed system aims to automatically generate keywords for summarizing single e-newspaper articles. This method facilitates the

efficient extraction of critical information from a single document. Marcus et al. (Marcus, Santorini, and Marcinkiewicz 1993) developed a discourse-dependent summarization method that predicts the adequacy of summary texts based on discourse in a single e-newspaper articles. It is generally more convenient to process a single document, since its sentences are more coherent and have less redundancy.

3.1.2. Multi document text summarization

When dealing with the task of summarizing multiple documents, the input consists of a collection of documents, and the goal is to produce a single summary document as an output (Shi et al. 2010)(Alguliev et al. 2011). Note that McKeown (McKeown and Radev 1995) was the first to use this kind of summary when they created the SUMMONS system. The work of Boudin (Boudin and Torres-Moreno 2009) is regarded as the first study to incorporate an evaluation of a multi-document automatic summarising approach, and it serves as a noteworthy example in the field of multi-document summarization. This type of summarization has the advantage of expanding the number of documents to be summarized (sources of information). However, multi-document text summarization is even more challenging than single-document summarization, due to the vast quantity of information and variety of topics.

3.2. Purpose based classification

Automatic Text Summarization (ATS) systems can be categorized into three distinct types based on the purposes of the summaries they generate.

3.2.1. Generic summary

By focusing just on the content of the original text and ignoring its context, the generic summary is created. In this type of summary, the ATS tries to concentrate on the subject matter supplied by the author and present the content of the input document. This kind of summary has several examples, among which we identify (Kazantseva 2006). The goal in this work is to provide the reader with a brief summary about a given story. The user in this work is supposed to search on the general theme of the story, to judge whether he wants to read it or not. This type of summary is advantageous if readers are more interested in the main idea of a text. Nevertheless, readers may have a preference for certain topics over others.

3.2.2. Query-oriented systems

Query-oriented systems produce documents' summaries according to user preferences. In (Bhaskar and Bandyopadhyay 2010) the authors propose a based query technique for summarizing multiple documents. They calculate the measure of correlation between sentences in each document to create clusters of similar sentences. Subsequently, these phrases are given scores based on their relevance to the query. These scores are then aggregated with the cluster scores to extract the highest-scoring sentences. Recent techniques, including deep learning, have been used to generate query-oriented summaries (Zhong et al. 2015). The advantage of query-oriented methods: They provide users with summaries that match their preferences. However, the most significant challenge faced by these methods is identifying and extracting query-relevant phrases from documents while maintaining coverage of the main topics.

3.2.3. Update summary

Update summarization is a variant of automatic text summarization including the additional dimension of time. While in the automatic summarization problem the input data is static, dynamic summarization introduces an additional difficulty by varying the input data on the time axis. Work on this type of summary can be classified into two categories. Sequential dynamic summary systems produce a summary reflecting the information contained in documents covering a given period by taking as a reference point the information known just before this period, embodied by a summary (HLTCOE, n.d.),(Z. Yang et al. 2013). Incremental dynamic summary systems produce updates to an initial summary each time new information appears regarding the subject of the initial summary (McCreadie, Macdonald, and Ounis 2014).

3.3. Output based classification

The summarizer system can generate two sorts of summaries: informative summaries and indicative summaries. Each type has a particular purpose and various levels of detail.

3.3.1. Informative summary

Informative summaries include crucial information extracted from the source document(s), helping readers to pick out the primary concepts, topics and ideas. This type of summary helps readers quickly understand the essential content of a document (s); it can be used in

newspapers, academic publications such as, theses, research articles,…etc. In conclusion, informative summary serves as concise reproductions of the source document(s). The main challenge in this type of summary is to cover most ideas presented in the original document (s) with a minimum of redundancy.

3.3.2. Indicative summary

Indicative summaries do not provide informative information about the content but offer an overall description of the input document(s), covering its purpose, field, and research approach. This assists the user in deciding if consulting the original document is helpful or not. In their study, Kan and al.(Kan, McKeown, and Klavans 2001) put out a proposed system for multi-document summarization known as CENTRIFUSER. This system is built upon the concept of content planning. To summarize, indicative summaries serve as a description to assist users in selecting documents. It can be more difficult to describe a document than to figure out what it is about. For example, offering information on places, personalities,…etc in documents.

4. Automatic Text Summarization (ATS) approaches

In the field of text summarization, we distinguish two main approaches: extractive summarization, which relies mainly on statistical methods to determine how to extract portions of the text to produce a summary. Abstractive summarization, on the other hand, necessitates a nuanced comprehension of language in order to create new, contextually appropriate summaries. There is also a hybrid approach that incorporates components of both extractive and abstractive techniques. This section delves into these approaches, offering light on the differences between them.

4.1. Extractive approaches

The extractive summary is made up of text segments extracted from the source text. These segments can be sentences, clauses, or any other unit of text. The initial studies in automatic summarization (H. P. Luhn 1958) rely on this method by utilising word frequency. The selection criteria were then enriched by taking into account the content and structure of the text (Edmundson 1969). These methods were initially the most popular since they avoided the difficulty of text generation, which was traditionally regarded as a challenging task. The advantage of extractive summarization is that it does not require text generation. This allows

concentrating on selecting relevant content while still obtaining a readable and linguistically correct summary. However, coherence is not assured. For example, if the summary method selects sentences having references (acronyms, personal pronouns, etc.) but not sentences containing their antecedents, the resulting summary may be incoherent. In order to fix this problem, several works use the paragraph as the extraction unit rather than the sentence (Salton et al. 1996). This preserves the coherence of the summary but cannot be used in the case of short summaries. Furthermore, it is evident that this strategy reduces summary precision by adding unnecessary sentences only to improve coherence. Other researchers apply pre/post-text processing steps to improve the overall coherence of the summary, such as the resolution of anaphoric references in the source text (Trandabat 2011). The basic procedure in extractive summarising is the selection of relevant and non-redundant text segments (typically sentences) without exceeding a summary size restriction. This principle limits the coverage of the information provided by the source text.

4.2. Abstractive approaches

Abstractive summarization techniques imitate the writing style of humans when producing summaries. These systems are characterised by their capacity to generate summaries that closely resemble those that are produced manually. The production of the summary consists of two main stages: understanding the original text and generate the summary (Khan and Salim 2014). Understanding the text requires a deep semantic analysis as the initial phase. This involves identifying the essential content of the text and determining the key elements to be included in the summary. This step sometimes take the form of a task of extracting information specific to the domain covered in the text (Genest and Lapalme 2011), or of grouping important sentences from the source text (Filippova 2009). The second step is text generation, which is a challenging task. A simplified approach in this step is to use text-to-text transformation techniques. For example, the use of paraphrases (Madnani and Dorr 2010), or sentence merging and compression (Filippova 2009) are common techniques.

Abstractive systems are challenging to build since they rely heavily on linguistic techniques. The domain of a system can be specified to make the design of this type of system easier (Mitkov and Unit 1993). A semantic representation of the text can be used to construct the abstractive summary (Barros et al. 2019),(W. Li and Zhuge 2019). Many recent efforts (Nallapati et al. 2016),(Ling 2017)(Gao et al. 2019) have attempted to construct abstractive summaries using deep learning. Abstractive summaries have importance because they are

close to human abstracts. However, producing summaries that conform to grammar rules is still an immense challenge.

4.3. Hybrid approaches

Both extractive and abstractive automatic text summarization techniques are helpful and have advantages and disadvantages. Extractive summarization is relatively simple to apply than abstractive summarization. Hybrid approaches for text summarization are created by combining various strategies (extractive and abstractive) by enhancing their advantages and minimising their disadvantages. It involves extracting specific sentences from the text to summarize and generating new sentences that do not exist in the original text (Binwahlan, Salim, and Suanmali 2010).

Extractive and abstractive approaches are combined with statistical and semantic features. Emotions in the text were employed by the authors (Bhat, Mohd, and Hashmy 2018) as a semantic characteristic. Since the user's emotional affinity is mostly determined by their emotions, the writer believes that lines with implicit emotional content are important and should be included in the summary. The extracted summary is then entered into the Novel Language Generator, a hybrid summarizer that turns an extractive summary into an abstractive summary by combining WordNet, the Lesk algorithm, and POS.

5. Evaluation of ATS

There is not one single method to evaluate the effectiveness of an automatic text summarization system; in this section we will discuss a number of approaches to evaluate the ATS, these approaches are divided into two categories, intrinsic and extrinsic evaluation approaches. In the first category we examine various intrinsic measures that are frequently used to assess information retrieval in general as well as automatic text summarization systems in particular. The Pyramid technique, BE, and ROUGE, which are all related to this type of evaluation, will be covered in detail. The second category, called extrinsic evaluation, examines the summaries based on a variety of criteria, such as reading comprehension and relevance assessment. Finally, we will discuss the challenges that make evaluating automatic summarization systems difficult.

5.1. Classification of ATS evaluation methods

The evaluation of text summaries can be classified into two different categories (Jones and Galliers 1995). The first category refers to intrinsic evaluation, which focuses on the internal assessment of the summary system. The main focus lies on the coherence and informative substance of the generated summaries. The second category, referred known as extrinsic evaluation, focuses on assessing the impact of summaries on several tasks, including relevance assessment and reading comprehension. Figure 2 shows the several classes of evaluation methods for an automatic summary.

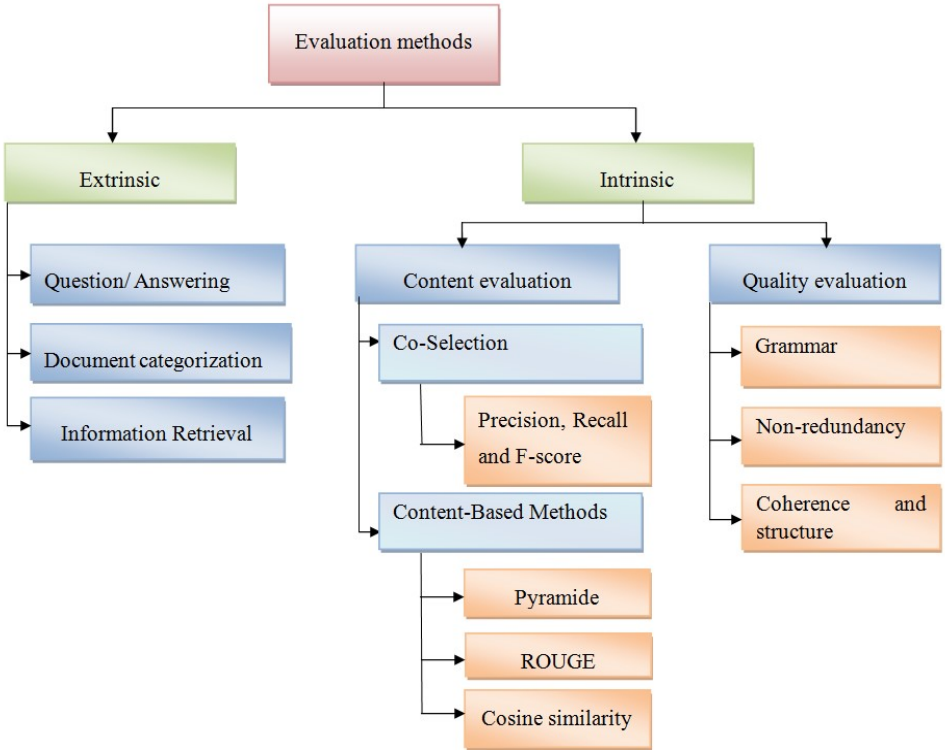

Figure 2: Classification of summary evaluation methods

5.1.1. Intrinsic evaluation

The process of intrinsic evaluation commonly involves the comparison of automatically generated summaries with a reference summary generally generated by an expert human or other summarization systems. This type of evaluation can be also categorised into two categories, content evaluation and text quality evaluation (Steinberger and Jezek 2009).

a. Quality evaluation

Methods for evaluating the quality of text aim to assess various linguistic aspects of the produced summary, including grammatical accuracy, clarity, coherence, non-redundancy and readability.

- **Grammar:**

There shouldn't be any formatting mistakes, grammar mistakes, or non-textual features like tags in the summary. Sentences ought to be free of improper terminology and grammar errors. Important aspects include gender and number agreement, tense agreement, and using idiomatic language correctly.

- **Non-redundancy:**

The generated summary should be concise and free of redundancies. To provide clarity and prevent reader confusion, it is important to present each piece of information only once, hence eliminating the inclusion of excessive or redundant data.

- **Coherence and structure:**

A good summary should be well organized, following a logical structure, and the sentences should flow coherently to make the summary clear and understandable to the reader.

b. Content evaluation

The content evaluation methods can be divided into two sub-classes: co-selection methods and content based methods.

- **Co-Selection Methods**

These approaches assess a system's ability to select appropriate sentences from a given text. This method is especially beneficial for tasks like summarising, in which the goal is to extract the most relevant or crucial sentences from a larger text or document. Co-selection methods often include comparing the sentences selected by the summarization system to a set of sentences deemed appropriate by human evaluators. Metrics such as precision, recall, and F-score may be used in the evaluation to determine how well the system's selections overlap with human selections.

We calculate these metrics as follows:

$\boldsymbol{R_{sum}}$**:** set of sentences from the reference summary.

$\boldsymbol{C_{sum}}$**:** set of sentences from the summary generated by the system (candidate).

$\boldsymbol{C_{sum} \cap R_{sum}}$**:** the intersection between the reference summary and the candidate summary.

- **Precision**

Precision refers to the correctness of the system, in other words, the number of correct sentences given by the system (the automatically generated summary) compared to the total number of sentences in the generated summary.

$$Precision = \frac{\|C_{sum} \cap R_{sum}\|}{\|C_{sum}\|} \qquad (1)$$

- **Recall**

Recall measures the number of correct sentences in the generated summary compared to the total number of sentences in the reference summary.

$$Recall = \frac{\|C_{sum} \cap R_{sum}\|}{\|R_{sum}\|} \qquad (2)$$

- **F-score**

The F-score is the harmonic mean of precision and recall, providing a balance between the two.

$$F - score = \frac{2 * Precision * Recall}{Precision + Recall} \qquad (3)$$

- **Content-Based Methods**:

Content-Based Methods, ROUGE (Lin and Hovy 2003), BE (E. H. Hovy et al. 2006), and Pyramid (Nenkova, Passonneau, and McKeown 2007) aim to explore more extensively by directing attention towards more granular components of text, such as individual words, sentences, or n-grams (which refer to consecutive sequences of n items extracted from a specific text sample). Content-based evaluation offers a more detailed and nuanced assessment, enabling a deeper understanding of the system's proficiency in capturing essential concepts, vocabulary, and stylistic components within the text. The concepts guiding each of these methods will be detailed and explained in the parts that follow.

- **ROUGE**

Lin and Hovy (Lin and Hovy 2003) introduced; Recall-Oriented Understudy for Gisting Evaluation (ROUGE), an approach inspired by another used for automatic translation evaluation called BiLingual Evaluation Understudy (BLEU) (Papineni et al., 2002). The goal is to automatically compare the quality of the generated summary to a reference one. The idea is to count the number of units (N-grams) in both the summary generated by the system to be evaluated and the reference summary and then compute the recall. This approach allows the use of several reference summaries because a text may have multiple reference summaries. A higher ROUGE score indicates that the automatically generated summary is more similar to the reference one. ROUGE has many variants(Lin 2004): ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S, and ROUGE-SU.

ROUGE-N: represents the overlap of N-grams between a system generated summary and a reference summary; it is calculated in (4) as follows:

$$ROUGE\_N = \frac{\sum_{S \in summ-ref} \sum_{N-g \quad \in S} Count_{Match}(N\_gram)}{\sum_{S \in summ-ref} \sum_{N-gram \in S} Count(N\_gram)} \qquad (4)$$

Where $Count_{Match}$ (N_gram) is the number of N_grams that are found both in the system summary and in a reference summary. For Count(N_gram), it corresponds to the number of N-grams in the reference summary.

ROUGE-L: The longest common subsequence (LCS) between the system-generated text and the reference text is precisely measured by ROUGE-L. The LCS is the longest word sequence that appears without gaps or changes in word order in both system-generated and reference summaries. ROUGE-L uses the LCS to compute precision, recall, and F-score.

ROUGE-S: The goal of ROUGE-S is to calculate the co-occurrence of skip-bigrams between a computer-generated summary and a reference summary (often written by humans). A skip-bigram is any pair of words in a sentence that can have any number of gaps. This enables the evaluation to capture more flexible and distant word associations than a basic word sequence.

ROUGE-SU: The weakness of ROUGE-S is that it only considers bigrams. If a sentence contains no overlapping of bigrams, it will not give any weight to these sentences. To overcome this problem of ROUGE-S, ROUGE-SU is an extension that also considers the

unigram with bigrams.

- **Pyramid**

Pyramid is a semi-automatic approach. It allows a candidate summary to be compared to a group of reference summaries (Nenkova and Passonneau 2004). Because there is no perfect summary and writing styles vary from person to person, utilising a single reference summary does not satisfy the criteria of equity among candidate summaries. To overcome this limitation, the assessment campaigns give at least four reference summaries. The PYRAMID approach works by annotating reference summaries in order to identify units known as SCUs (Summary Content Units). A SCU is a collection of textual units of reference summaries that express the same information. It obtains a weight according to the number of reference summaries that instantiate it. These SCUs can be organised in a pyramid, with each layer grouping SCUs of the same weight together. A summary is annotated to identify the candidate SCUs it contains in order to evaluate it. As a result, each candidate SCU obtains the weight of the pyramid's most similar SCU. The summary's PYRAMID score is finally the ratio of the total of the weights of its entire candidate SCUs to the weights of a reference summary with the same number of SCUs. The negative point of this approach is that the annotation step is still challenging to automate.

$$\text{Pyramid score} = \frac{\sum poids(SCU_{candidate})}{\sum poids(SCU_{reference})} \qquad (5)$$

- **Cosine similarity**

The cosine similarity evaluates the similarity between the summary generated by the system Sum_{system} and the reference summaries Sum_{system}. We calculate the cosine similarity between these two summaries by the formula (6);

$$Cosine\ similarity(Sum_{system}, Sum_{reference}) = \frac{\sum_{i=1}^{m} x_i * y_j}{\sqrt{\sum_{i=1}^{m} {x_i}^2} * \sqrt{\sum_{i=1}^{m} {y_j}^2}} \qquad (6)$$

Where, $(x_1,\ldots..,x_m)$ and $(y_1,\ldots\ldots,y_m)$ are the vectors representation of Sum_{system} and $Sum_{reference}$, respectively.

5.1.2. Extrinsic evaluation

The summary is evaluated in the extrinsic evaluation based on its usefulness for a target task. Task-based evaluation methods do not analyse lexical units in the summary, but rather attempt to quantify the probability that these summaries will be employed in a particular task. In the scientific literature, there are various techniques to task-based summary evaluation. Among these approaches, we highlight the three most significant tasks: document categorization, information retrieval, and question answering.

a. Document categorization

The evaluation of automatic summaries revolves around assessing their effectiveness in substituting entire documents for categorization purposes. This involves determining if the generic summary adequately captures the information needed for accurate document categorization. The evaluation requires a corpus of documents and their corresponding topics. Results are compared against categorizing full documents (upper bound) or random sentence extracts (lower bound). Categorization can be done manually (Mani et al. 1999) or through a machine classifier (Hynek and Jezek 2003).

The main evaluation metrics for classification are precision and recall. Precision, in this particular context, refers to the ratio of accurately assigned topics to a document, divided by the total number of topics allocated to that document. Recall is the ratio of the number of accurately assigned topics to a document, to the total number of topics that are supposed to be allocated to the document.

b. Information Retrieval

Information retrieval (IR) is a suitable task for evaluating the quality of a summary using a task-based approach. Relevance correlation (Radev et al. 2003) is an information retrieval (IR) metric used to evaluate the extent to which retrieval performance decreases when transitioning from complete texts to summaries. If a summary effectively contains the key elements of a document, then an information retrieval machine that is based on a collection of such summaries (rather than a collection of the complete documents) should yield a result that is nearly as satisfactory. Furthermore, the disparity in performance between the summaries and the complete documents could potentially be used as an indicator of the quality of the summaries.

c. Question Answering

An extrinsic evaluation measures the influence of summarization inside a particular task, such as question answering. In (Morris, Kasper, and Adams 1992), the authors conducted an experiment to investigate the impact of automated summarization on comprehension of documents using question and answer tasks. This evaluation centres on the extent to which summaries enhance the efficacy of responding questions. This entails analysing if summaries improve the extraction of pertinent information required to address user queries. The evaluation seeks to assess enhancements in task performance, user happiness, and the overall efficacy of summaries in enabling precise and speedy responses to questions by including summarization into the question-answering job.

6. Evaluation campaigns

6.1. TIPSTER SUMMAC

In 1998 (Mani et al. 1999), the TIPSTER SUMMAC (Summarization Conference) (organized by the National Institute of Standards and Technology (NIST)) Automatic Text Summary Assessments took place in Maryland. (USA). SUMMAC was the first large-scale, developer-independent evaluation of automatic text summarization systems. Based on the activities, which were generally conducted by U.S. government data analysts, three major evaluation tasks were specified:

• The ad hoc task (intrinsic).

• The categorization task (intrinsic).

• Question-answer (QA) task (extrinsic).

SUMMAC conducted an evaluation of the automatic text summary which proved to be very effective in measuring relevance. The summary algorithms eliminated the text (83% and 90% for the guided and generic summary, respectively), while achieving the same level of relevance as the source documents and reducing approximately half the analysis time. The Quality Assurance task introduced new automated methods to measure the informative nature of a specific subject-based summary. Automatic content-based rating is in positive correlation with the grades produced by human judges. The evaluation methods used in the SUMMAC campaign had two interests: to evaluate summaries and other results related to natural language processing (NLP) technologies.

6.2. DUC/TAC

Since 2001, the NIST has organized the "Document Understanding Conference Campaigns" (DUC) to evaluate the performance of NLP algorithms. As of 2008, these campaigns have been renamed Text Analysis Conference (TAC). TAC campaigns are much more ambitious than DUC campaigns. Since 2008, TAC has organized workshops around four themes: summary, question-answer, recognition of textual involvement and knowledge base population. The objective of the DUC/TAC campaigns is twofold: on the one hand, to promote progress in the field of automatic text summaries and, on the other hand, to enable researchers to participate in large-scale experiments, enabling them both to develop and evaluate their systems.

The DUC/TAC campaigns successively introduced the following tasks:

• DUC 2001 - DUC 2002: Mono and multi-document generic summaries.

• DUC 2003: Short Summary (headline) and multi-document.

• DUC 2004: Short Summary, multi-document generic summaries, Multilingual Document Summary and Biographical Summary.

• DUC 2005: Guided summary and biographical summary guided by questions on the subject

• DUC 2006: Guided summary of several documents.

• DUC 2007: Multi-document guided summary for lengths of or up to 250 words, from groups of approximately 25 documents, update summary.

• TAC 2008-TAC 2009: Update task summarization - Multi-document guided summary.

• TAC 2010: Guided summarization, Automatic Summary Evaluation (Automatically Evaluating Summaries Of Peers)

• TAC 2011: Guided summarization, automatic summary evaluation and the new MultiLing pilot task to promote the use of multilingual algorithms for the summary.

• TAC 2014: Summary of biomedical texts (Biomedical Summarization)

6.3. NTCIR

The third NTCIR workshop (2001-2002) focused on the evaluation of information extraction systems (RI task), question-and-answer systems (QA task) and automatic text summary systems (auto-text summary task). Organized by NTCIR in Japan, the "Text Summarization Challenge" (TSC) aimed to automatically summarize texts published from 1998 to 1999 in the Japanese newspaper Mainichi [5]. There were three objectives:

• Promote research in IR, QA and text summary to provide reusable test corpus.

• Create an exchange forum for research groups interested in comparing results and ideas in an informal atmosphere.

• Improve the quality of feedback-based test corpus.

6.4. MultiLing

The Multiling workshop, initiated in 2011, evaluates language-independent summarization systems on various languages(Giannakopoulos et al. 2011). Participants systems process at least two languages out of seven (English, French, Greek, Czech, Hebrew, Hindi and Arabic.), the size of the generated summaries must be between 240 and 250 word. For the performance assessment, the workshop uses ROUGE (ROUGE-1, ROUGE-2, ROUGE-SU4), MeMoG and AutoSummENG as an automatic evaluation methods; and uses a manual evaluation method, when the human experts assign for each summary a score between one and five, based on the quality of the language and the content.

In 2013, MultiLing has been expanded to three tasks: Multilingual Multi-document Summarization (MMS) (Giannakopoulos 2013), Multilingual Single document Summarization (MSS) (Kubina, Conroy, and Schlesinger 2013), and Multilingual summary evaluation. Three more languages Romanian, Chinese, and Spanish were added to the seven previous languages used in MMS. The test corpus includes 10 topics for French, Chinese, and Hindi, and 15 topics for the remaining languages. The evaluation methodology remains the same, with addition of automatic metrics like ROUGE-3 and NPowER.

Multiling 2015 introduced two tasks: Call Center Conversation Summarization (CCCS) and Online Forum Summarization (OnForumS). CCCS (Favre et al. 2015) generates abstractive summaries, while OnForumS (Kabadjov et al. 2015) combines automatic summarization,

argumentation mining, and sentiment analysis. Four research groups submitted their systems, evaluated using crowd-sourcing and human judges based on relation, agreement, and sentiment between sentences.

7. Conclusion

This chapter provided an overview of automatic text summarization. To assist the reader in comprehending this area, we have begun by providing definitions of the terms used in the field of text summarization. Because there are various classification factors, an automatic summary may belong to several classes or kinds. So far, we have found that these factors can be classified based on the source (the input document), the purpose, and the output document. We have seen that the summary is divided into two parts: the abstract summary and the extraction summary. The methods for automatic text summary evaluation were discussed. We discovered two ways to automatic summary evaluation: intrinsic and extrinsic evaluation. Measurements are employed to evaluate the summary; among those mentioned, the most commonly utilised in the intrinsic evaluation are ROUGE and Pyramid.

The last section in this chapter is dedicated to the evaluation companies in the field of text summarization due to the importance of assessment in validating the proposed approaches. The campaigns, including SUMMAC, NTCIR, and DUC/TAC, have a vital function in evaluating and setting a standard for the performance of different summarization systems. Their offerings include standardised datasets, evaluation measures, and specialised activities, all of which enable a fair and comprehensive comparison of various methodologies.

The next chapter will present a state-of-the-art review of multi-document text summarization, which is the central focus of our thesis.

.

Chapter 2

State of the art of multi-document text summarization approaches

1. Introduction

Our research fits within the field of multi-document summarization, a type of automatic text summarization that deals with multiple documents rather than just one. The purpose of a multi-document summarization aims to summarise information from many documents on a specific topic in a simple and useful manner. Rather than requiring individuals to read all documents in their entirety, the multi-document summarization provides a brief overview of essential points, concepts, and pertinent information. This method improves overall understanding of complex issues by decreasing the amount of text to read. On a practical level, our thesis focuses mostly on the extractive approach to solving the problem of multi-document summarization.

This chapter provides a state-of-the-art overview of extraction approaches utilized in automatic multi-document summarization. We begin by outlining the steps involved in automatic multi-document summarising, including pre-processing and the criteria utilised in extractive techniques. Following that, we go over several multi-document summary extraction approaches, followed by a detailed discussion of these methods.

2. Multi-document text summarization steps

Automatic text summarization is a complex process that takes place in several distinct steps, whether in single-document or multi-document approaches. These steps are crucial to condensing information in an intelligible and informative way, either by directly selecting fragments of the original text (extractive approach) or by generating new sentences that capture the main ideas of the content (abstractive approach). These steps include the pre-processing of the original text, the determination of selection criteria, and the application of

these criteria to produce the final summary. In this section, we will explain these different steps in detail.

2.1. Pre-processing

In automatic text summarization, pre-processing is crucial. We discuss the several phases of pre-processing that involve the application of Natural Language Processing (NLP) techniques (Brants 2003). These techniques are designed to make the input text more available for further processing and to standardise it so that it can be used as input for the next module of the process.

2.1.1. Normalization

Within the text preparation process, the normalisation stage is critical for making the text more uniform and facilitating future processing. It consists of numerous procedures aiming at standardising the text, eliminating extraneous variances, and optimally preparing it for the future processing phases. This standardisation could include the following tasks:

- ✓ Lowercase all text.
- ✓ Correction of spelling errors.
- ✓ The elimination of special characters and accents, and other adjustments to ensure consistency and uniformity in textual content.

2.1.2. Text segmentation into Sentences

Text segmentation into sentences is an important natural language processing task that requires dividing a continuous block of text into separate grammatical units known as sentences. This procedure is essential for a variety of applications such as text analysis, machine translation, text summarization, sentiment analysis, and others. Here's an explanation of how text is segmented into sentences:

a. Identification of sentence boundaries

The process of sentence segmentation usually involves the identification of sentence-ending punctuation symbols, such as periods (.), exclamation signs (!), and question marks (?) (Nunberg 1990).These punctuation marks frequently signify the end of a sentence.

b. Handling Abbreviations

When performing the segmentation process, it is important to take into account the existence of abbreviations, as some punctuation marks may not always indicate the end of a sentence

(Reynar and Ratnaparkhi 1997). Take, for instance, the phrase "Mr. Smith," which should not be interpreted as two distinct sentences.

c. Dealing with points in Acronyms

Certain acronyms like "U.S.A." or "e.g.", include periods. The process of sentence segmentation should ensure that these terms are not separated into distinct sentences (Riley 1989).

d. Contextual Analysis

Contextual analysis is employed in more advanced methods to determine segmentation by considering grammar and syntax. It helps in identifying cases when final punctuation marks may lack clarity.

2.1.3. Tokenization

Word segmentation is an essential step in the majority of natural language processing tasks(Sun, Luo, and Chen 2017). The procedure involves inputting a document in the form of text and then segmenting it into words and punctuation marks (Appel et al. 2016). The purpose of this process is to analyse the words within sentences. The significance of word segmentation resides in the identification of pertinent keywords (Kumar and Chandrasekhar 2012). In Latin languages such as English, word segmentation is typically accomplished by utilising spaces and punctuation marks.

2.1.4. Stemming

Stemming, a crucial pre-processing step in natural language processing (NLP), entails a series of tasks. To begin, the text is tokenized to identify individual words, and then a stemming algorithm such as Porter (Porter 1980) or Snowball Stemmer (Porter 2001) is applied. Each word is then stemmed to its root form, with any prefixes or suffixes removed. The results are simplified root forms of the original terms. Lemmatization may be used as an option for a more context-aware reduction. The stemmed words are refined in post-processing steps, and the result is fed into the larger NLP pipeline for tasks such as classification or sentiment analysis. By focusing on fundamental word meanings, stemming can reduce dimensionality, promote consistency, and improve model generalisation. It is especially useful in situations where different words have similar semantics and a streamlined word representation is desirable for subsequent tasks.

2.2. Selection criteria and intermediate representation

The selection criteria in an automatic text summarization system have an important role in determining which textual units should be included in the final summary. These criteria vary depending on the language model used and can include factors such as semantic significance, term frequency, grammatical structure, or other content-specific properties. Sentences, N-grams, or other textual units can be chosen as textual units. It is important to highlight that these requirements are not specific to any approach and apply to all forms of extractive summaries, whether single-document or multi-document summaries. These criteria can be divided into two distinct categories: the first category includes the criteria related to the text's content, and the second category includes the criteria related to the text's form and structure. Each category will be covered in detail in this section.

2.2.1. Criteria related to text content

This category of criteria focuses on the content of the text and the information it provides. Either surface methods, like calculating word frequency of occurrence, or semantic methods, which take advantage of word meaning and semantic linkages, like semantic role annotation, are used to analyse the content. The most widely applied criteria are shown below.

a. Term Frequency TF

Luhn originally proposed this criterion in 1958 (H. P. Luhn 1958). The idea behind it is that the most frequent words are closely related to the topic of the text. The term frequency TF is widely exploited, even in recent systems where it is combined with other criteria. Even approaches based on semantic analysis of words often start with TF to determine the main themes of the text. The notable advantage of this criterion is its complete independence from language.

TF_{ij} represents term frequency of term i in document dj, it's given in formula (7).

$$TF_{ij} = \frac{freq_{ij}}{\|d_j\|} \qquad (7)$$

Where, $freq_{ij}$ is the number of occurrences of term i in document dj.

$\|d_j\|$is the sum of frequencies of all terms in document dj.

When it comes to domain-relative terms, TF has an issue. To overcome this issue, Luhn(H. P. Luhn 1958) employs two thresholds to ensure that the sentence is important but not particular to the document's domain. A more advanced technique is to employ TF-IDF, which was first defined by Salton et al (Salton and Yang 1973). Given a corpus D including N document, the measure Inverse Document Frequency (IDF) is calculated as illustrated in the Equation (8).

$$IDF_i = \log_{10}(N/n_i) \qquad (8)$$

n_i represents the number of document containing the term i.

The weights of terms of each document in TF-IDF representation are given in formula (9)

$$\text{TF} - \text{IDF}_{ij} = TF_{ij} * IDF_i \qquad (9)$$

b. Cue words

The identification of certain words, such as "significant" or "impossible," can serve as robust indicators of a sentence's relevance to the central theme, as outlined by Edmundson in (Nobata and Sekine 2004). To systematically capture and categorize these cues, a dictionary is constructed from a corpus, cataloging three types of cue words: Bonus words, indicating positive relevance; Stigma words, denoting negative relevance; and Null words, signifying irrelevance. The scoring mechanism for a sentence, denoted as 's_i,' is derived from this feature and calculated as the summation of weights assigned to each word 'w' based on its classification in the dictionary, as expressed in Equation (10).

$$Score_{cue}(s_i) = \sum_{w \epsilon s_i} cue(w) \ \ where\ cue(w) = \begin{cases} b > 0 \ \ if(w \epsilon Bonus) \\ \delta < 0 \ \ if(w \epsilon Stigma) \\ 0 \qquad\quad otherwise \end{cases} \qquad (10)$$

This approach leverages a nuanced understanding of the semantic implications of specific words, enabling a more sophisticated assessment of sentence importance within the context of the overarching topic.

c. Similarity between text segments

The measurement of textual similarity plays a crucial role in various related areas such as text categorization, information retrieval, clustering, topic retrieval, subject tracking, and text summarization (Abo-Elghit, Al-Zoghby, and Hamza 2020). In the literature, several methods

are available to measure the similarity between textual segments. Among these, we can cite the most commonly used measures, such as cosine similarity (Deza et al. 2009) and the Jaccard index (Jaccard 1912).

The cosine similarity evaluates the similarity between two documents (sentences) vectors. We calculate the cosine similarity between two documents d_i and d_j using the following formula (11);

$$Cosine\ similarity(d_i, d_j) = \frac{\sum_{k=1}^{m} w_{ik} * w_{jk}}{\sqrt{\sum_{k=1}^{m} {w_{ik}}^2} * \sqrt{\sum_{k=1}^{m} {w_{jk}}^2}} \qquad (11)$$

W_{ik} and w_{jk} are the weights of the word k in the documents d_i and dj, respectively.

Jaccard similarity between two documents is the ration between the intersection and the union of the sets of words that represent these documents. We calculate the Jaccard similarity between two documents; d_i represented by the set of words A and d_j represented by the set of words B using the following formula (12);

$$Jaccard\ similarity(d_i, d_j) = \frac{\|A \cap B\|}{\|A \cup B\|} \qquad (12)$$

The similarity between textual segments is initially used in the field of automatic text summarization to eliminate redundancy, but it also plays an indirect role in the selection of relevant sentences. Furthermore, it is used during evaluations to compare results with model summaries. Certain summarization approaches, such the mono-document TextRank algorithm created by Mihalcea (Rada Mihalcea 2004), depend entirely on this particular criterion.

d. Named entities identification

The identification of named entities in text enhances the process of selecting pertinent information(Hassel 2003). Additionally, it enables us to respond to factual questions such as WHERE, WHO, WHEN,…etc. within the guided summary (Tan 2011). Certain researchers surpass this stage and determine the semantic roles of the identified entities (Trandabat 2011). The entity that occurs most frequently is identified and regarded as the most important entity. Following that, sentences that include this entity are selected. In the end, only sentences in which the main entity plays a crucial (non-auxiliary) semantic role are retained for the summary.

Semantic roles can also be utilised to simplify complex sentences, i.e., sentences which includes two or more predicates. Typically, the predicate is a verb. In this situation, the predicates for which the main entity has an auxiliary role are deleted.

2.2.2. Criteria related to the form and structure of the text

The structure of the text is crucial in determining the importance of a sentence. The sequence of sentences is not arbitrary while producing a manuscript. Furthermore, writing styles change from one field to the next. In the journalistic sector, for example, the most relevant information is frequently provided within the beginning of the paragraph. This is not necessarily true in a research paper or a book. NLP researchers have used this factor to determine the relevance of textual segments. In the following, we will go over the most significant criteria.

a. Position in the text

This criterion is dependent on the distinct characteristics and types of the document. Typically, sentences positioned at the beginning of a document are more informational, frequently providing a description of the primary topic. Moreover, the introductory sentences of each paragraph typically offer additional relevant information (Lin and Hovy 1997). Edmundson (Edmundson 1969) considers that initial and final sentences generally include more informative content than other sentences.

The study conducted by Baxendale (Baxendale 1958) confirms that the initial and concluding sentences inside paragraphs hold greater significance. The author utilised a corpus consisting of 200 paragraphs to evaluate the significance of sentences based on where they were situated inside a paragraph. He discovered that in 85% of these paragraphs, the initial sentence holds the greatest significance, while in 7% of them, the final sentence does.

The position of sentences in the text can be utilised in several ways to provide scores to sentences. In (Nobata and Sekine 2004) Nobata and Sekine provide three methods for evaluating the score of a sentence s_i, where i represents its location inside a text containing n sentences.

The first method, as described in formula (13) considers that only the initial sentences up to a specified position N have significance.

$$Score_{pos}(s_i) = \begin{cases} 1 & if \quad (i < N) \\ 0 & otherwise \end{cases} \qquad (13)$$

In formula (14), the second method proposes that the relevance of a sentence is inversely proportional to its position.

$$Score_{pos}(s_i) = \frac{1}{i} \qquad (14)$$

According to formula (15), the third method proposes that the initial and final sentences have more significance.

$$Score_{pos}(s_i) = \max(\frac{1}{i}, \frac{1}{n-i+1}) \qquad (15)$$

Fattah and Ren(Nobata and Sekine 2004)employ sentence's position within paragraphs as an alternative to the entire text. It is assumed that the beginning five sentences constitute the most important portion of a paragraph, formula (16).

$$Score_{pos}(s_i) = \begin{cases} 6-i & if \quad (i \leq 5) \\ 0 & otherwise \end{cases} \qquad (16)$$

The position of the sentence within the paragraph is denoted by i.

b. Title and subtitle words

The document's title reflects its subject matter, which was initially proposed by Edmundson in (Edmundson 1969). When we partition a document into sections and subsections, we select indicative headings for each. Any sentence that includes words from a title is regarded as significant. This feature could be regarded as if the title represented a request(Salton and Buckley 1988).

Ishikawa et al (Ishikawa 2001)combine this characteristic with term frequency, giving more importance to the frequencies of the words in the title compared to normal words. In Equation (17), a sentence s_i is evaluated based on the frequencies of its terms w. If a word belongs to the title, its frequency is multiplied by a number A that is greater than 1 (the authors specifically specified A = 3).

$$Score_{title}(s_i) = \sum_{\{w\} \epsilon s_i} \alpha(w) * tf(w) \ \ where \ \alpha(w) = \begin{cases} A > 1 & if (w \ \epsilon \ title) \\ 1 & otherwise \end{cases} \qquad (17)$$

In order to evaluate sentences based on the terms in the title, Nobata (Nobata and Sekine 2004) suggests two approaches. The initial approach involves calculating the term frequency

multiplied by the inverse document frequency (tf * idf) of the terms in the title T, as specified in Equation (18).

$$Score_{title}(s_i) = \frac{\sum_{w \in T \cap s_i} \frac{tf(w)}{tf(w)+1} * idf(w)}{\sum_{w \in T} \frac{tf(w)}{tf(w)+1} * idf(w)} \qquad (18)$$

The second approach utilises named entities, denoted as "e," and the frequencies of words, denoted as "Tf," as demonstrated in the equation (19).

$$Score_{title}(s_i) = \frac{\sum_{e \in T \cap s_i} \frac{tf(e)}{tf(e)+1}}{\sum_{w \in T} \frac{tf(e)}{tf(e)+1}} \qquad (19)$$

c. Sentence length

The average length of a sentence in a text varies by genre. In general, very short sentences are typically thought to be uninformative, whereas too long sentences are expected to explain information that has already been presented more clearly. This feature is used by defining a length range, which is typically between 15 and 30 words. Sentences that surpass this limit will be penalized (Schiffman, Nenkova and McKeown 2002).

Two methods, presented by Nobata et al (Nobata and Sekine 2004), propose a formula for assigning a score to a sentence.

The first method assesses the score of a sentence by dividing its length by a predefined maximum value, L_{max}, if its length is less than L_{max}. Otherwise, it assigns a score of 1. The sentence's score is calculated in the equation (20).

$$Score_{len} \quad (s_i) = \begin{cases} \frac{|s_i|}{L_{max}} & if \quad (|s_i| \leq L_{max}) \\ 1 & otherwise \end{cases} \qquad (20)$$

The second method, which produces better outcomes, allows for a negative score to penalize sentences that are shorter than the specified minimum value L_{min}. The sentence's score is calculated in the equation (21).

$$Score_{leng} \ (s_i) = \begin{cases} 0 & if \quad (|s_i| \geq L_{min}) \\ \frac{|s_i| - L_{min}}{L_{min}} & otherwise \end{cases} \qquad (21)$$

Fattah and Ren (Fattah and Ren 2009) introduce an alternative formula that calculates the score of a sentence s_i in a document d based on its length. The score is calculated by multiplying the length of the sentence by the total number of sentences in document d, and then dividing the product by the overall length of document d. The score is given in the formula (22).

$$Score_{lengh}(s_i) = \frac{|s_i| * |\{s: s \in d\}|}{|d|} \quad (22)$$

d. Salient expressions

According to Paice (Paice 1980), salient expressions are structures that commonly appear and plainly indicate that the sentences in which they are found have significant information to convey about the subject matter or the message of the document. Statements like "the principal goal of this article is to investigate...", might be used as example of salient expressions. Recognising such expressions can be difficult, and Paice provides a detailed explanation of the processes involved in this procedure:

• Due to the diverse range of indicators, it is not feasible to provide an exhaustive list. These statements, such as "This article is concerned with...", "Our paper deals with..." and "The following discussion is about...", all have the same structure. Therefore, the resolution entails employing certain templates.

• Templates can include additional word sequences that are not considered indications themselves. To address this issue, it is recommended to employ skip limits to separate the paradigms within a specific template.

Optional words exist, although they can carry additional significance when utilised, as exemplified by the word "here" in the phrase "The purpose here is to ...". This issue can be resolved by establishing numerous routes in the template.

• In order to account for different word forms, the templates can utilise the stems of the words instead.

Figure 3 shows a template in which the words or stems serve as paradigms. The skip limitations are indicated as follows:

[3]. The weight increments are displayed in the following manner: The user's text is "+2." A query symbol (?) represents an optional pattern or model.

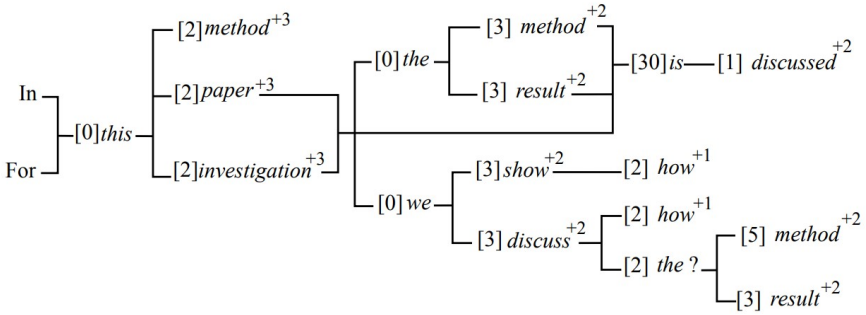

Figure 3: A slightly simplified template [Paice, 1981]

2.3. Exploitation and integration of criteria

Automatic multi-document text summarization systems typically do not rely on a single factor for sentence selection from the source text. Instead, they commonly combine numerous criteria. The integration strategies utilised for this objective are manifold and distinct. In this section, we thoroughly examine the many techniques used to combine these criteria and utilise the merged set to efficiently choose sentences that contribute to the ultimate summary.

2.3.1. Statistical methods

Statistical approaches for automatic multi-document summarization utilise statistical analysis of documents to extract crucial information and generate a cohesive summary. The criterion discussed in the previous section, such as term frequency, similarity between segments of text and other statistical criteria are employed to determine the most important sentences of the source documents. These techniques include term frequency (TF) extraction, which prioritises the most frequently occurring phrases, the document position method, which assigns significance to sentences located at the start of the text, and cosine similarity between sentences that identifies sentences sharing similar information.

The scoring of a sentence often involves the use of previous statistical features, as depicted in Equation 17. These features play a crucial role in evaluating the significance of a text unit in relation to the main theme or user queries, contributing to the generation of meaningful and contextually relevant summaries.

$$Score(s_i) = \sum_{f \in F} \alpha_f * Score_f(s_i) \ldots . (17)$$

The variable α_f represents the weight assigned to a specific feature f inside the set of features F.

2.3.2. Graph-based methods

Graph techniques, motivated by the PageRank algorithm (Rada Mihalcea and Tarau 2004), depict documents as a connected graph. The graph's vertices are formed by sentences, and the edges between the sentences represent their degree of similarity. A frequently used method to link two vertices involves assessing the similarity between two sentences and establishing a connection if the similarity exceeds a certain threshold. Cosine similarity with TF-IDF weights is the most commonly used method for measuring similarity. This graph representation yields two outputs. The divisions (sub-graphs) inside the graph represent distinct themes addressed in the documents. The second result concerns the identification of the crucial sentences within the document. Sentences that have several connections to other sentences in the partition have a higher probability to be the focal point of the graph and more probable to be included in the summary.

Graph-based techniques are applicable for both single and multi document text summarization (Erkan and Radev 2004). Due to their lack of requirement for language-specific linguistic processing, apart from sentence and word boundary recognition, they can be utilised for several languages as well (Rada Mihalcea and Tarau 2005). However, the utilisation of the TFIDF weighting system for similarity measurement is constrained as it solely retains the word frequency and does not consider the syntactic and semantic information. Therefore, incorporating similarity metrics that rely on both syntactic and semantic information improves the overall performance of the summarization system (Chali and Joty 2008).

In (Shen and Li 2010) a graph based approach is proposed to multi document text summarization, this approach typically represents documents to be summarized by a graph whose vertices are sentences. Two sentences are connected by an edge if their similarity is greater than a given threshold. The problem then comes down to finding the dominant set D of the graph which is a subset of vertices such that any top that does not belong to D has at least one edge in common with one of the vertices of D.

This paper (Bhaskar and Bandyopadhyay 2010), introduces a method for query-oriented multi-document summarization. The first stage is to cluster sentences by arranging them according to the similarity graph between sentences. Following that, each node is assigned a query closeness score. The best sentences from each cluster are then selected. The selected sentences are compressed using a syntactic analyzer to improve their conciseness.

2.3.3. Machine learning methods

In terms of learning, the process of automatic summarization of text by extraction has been tackled as different issues; each solved using different machine learning techniques. Most studies on artificial summarization view this process as a combined classification and regression challenge. The learning approaches strive to create a model for sentence selection in summaries by utilising a collection of source texts and their corresponding summaries. Source texts are classified based on different selection criteria.

In terms of classification, the selected model differentiates between sentences that should be included in the summary and those that should be excluded. Naive Bayesian models are commonly used and yield outstanding outcomes (Neto, Freitas and Kaestner 2002). In the following we present some works in multi-document summarization based on classification methods.

Zhang et al. (Y. Zhang et al. 2016) propose using an approach that relies on multi-view convolutional neural networks, which involve considering many perspectives. Initially employed in the field of 3D photography, these networks have since been utilised for the purpose of multi-document summarization. By utilising word embeddings of the words in the texts, these multi-view neural networks enable the generation of vector representations of phrases from several perspectives. These representations are ultimately merged. The final layer of the neural network employs the sentence representations to give each of them a score.

In the context of regression, the model is able to make predictions about the scores of sentences, as demonstrated by Conroy et al (Conroy et al. 2011). This allows for a quantitative assessment of the decision. Here, we present some studies in multi-document summarization that utilise regression as a basis.

In (Ren et al. 2016), Ren et al. present a novel regression-based approach, it is different from standard regression-based automatic summarization systems in that it addresses importance and redundancy independently. Rather, it addresses the assessment of importance and redundancy at the same time by comparing the relative importance of a sentence to a collection of sentences selected for the summary. This method incorporates extra criteria generated from the links between sentences, in addition to the criteria specific to each sentence. Experiments on the DUC 2001, 2002, and 2004 multi-document summary datasets show that this method surpasses other methods in term of ROUGE metrics.

Hong et al. (Hong, Marcus, and Nenkova 2015) presented an approach for merging summarization systems that is based on scores rather than rankings. After producing all candidate sentence combinations for the final summary, a regression model is trained to estimate the candidate summary's ROUGE-1 score. To do this, the model is based on various factors such as similarity to source texts, sentence location, redundancy, word frequency in huge corpora, and so on.

2.3.4. Methods based on Integer Linear Programming

McDonald (McDonald 2007) suggested that the problem of automatically summarising text could be formulated as an integer linear programming problem whose goal is to give the selected sentences the most weight. This weight is penalized because the words that are already in the summary are repeated. The model also includes the constraint of the maximum summary size. Gillick and Favre (Gillick and Favre 2009) changed the problem by adding an objective function that focuses on maximizing the weight of selected word bigrams, always under the constraint of the maximum length of the summary. Not having redundancies is generally favoured. Because each bigram is only counted once in the objective function, no matter how many times it appears in the final summary, this function tends to be higher as more bigrams are chosen. This also limits the number of times each bigram appears, which gets rid of redundancy.

In the paper (C. Li, Qian and Liu 2013), Li et al. recommend using the ILP model presented by Gillick(Gillick and Favre 2009) in a supervised context. Instead of utilising the frequency as a weight for each bigram, a regression model calculates the frequency of this bigram in the reference summary. The regression model employs a set of criteria (frequency, position, and length of the sentence containing the bigram, for example) to minimise the difference between the estimated and real frequencies. The ILP model is then in charge of selecting sentences from the summary.

2.3.5. Optimization approaches

As an optimisation problem, multi-document text summarization is effectively addressed by metaheuristic algorithms, particularly those based on swarm intelligence principles. The goal is to rapidly extract crucial information from various sources by picking a selection of sentences or paragraphs that constitute a succinct and meaningful summary when combined.

In this procedure, optimisation criteria like as relevance, coverage, and coherence are critical. Particle swarm optimisation (PSO) (Khan, Salim, and Jaya Kumar 2015) and ant colony optimisation (ACO) (Donis-Díaz, Bello, and Kacprzyk 2015) are two examples of metaheuristic algorithms that excel at navigating the complex solution space inherent in multi-document summarization. These algorithms iteratively seek for the ideal combination of phrases, providing high-quality summaries that overcome the constraints provided by large information in various publications by replicating collective behaviours observed in nature. The efficacy of swarm intelligence-based methods in handling the multi-document summarization problem is highlighted by this optimization-driven methodology. Here, we present different works that utilise swarm intelligence to solve the Multi-Document Summarization (MDS) problem.

In (Verma and Om 2019), the MDS problem is solved in two stages, in the first stage, the authors create a single document from the initial collection of documents, where similar sentences are removed to minimize the redundancy. Maximum coverage of the topic is considered to select the relevant sentences. The second stage is the summary generation, where the process is modelled as a shark smell optimization (SSO) problem. The experiments are performed on MultiLing13, TAC08, TAC11, DUC04, DUC06 and DUC07 datasets. The results show that the proposed system has better performance than the compared works in term of ROUGE-1 and ROUGE-2. The PSO algorithm is also used in step two in the experimentation phase.

In (Song et al. 2011), a fuzzy evolutionary optimization model (FEOM) is introduced for MDS. FEOM performs document clustering then selects the most pertinent sentences for each cluster to generate a summary. The model utilizes genetic algorithms to generate solution vectors for the groups and incorporates three control parameters to regulate the probability of crossover and mutation for each solution.

A Cat Swarm Optimization algorithm (CSO) is used to create a Multi-document summary (Rautray and Balabantaray 2017). Documents are represented by TF-IDF representation to calculate sentences' informativeness and then the inter-sentences similarity is calculate using the cosine similarity. The quality of the generated summary is measured using; the contents coverage, readability and cohesion. The datasets used in the experimentations are DUC2006 and DUC 2007. In the experimental study, the authors compare the performance of CSO

against harmony search (HS) and PSO algorithms. The comparison shows that the CSO algorithm gives better results than the other two algorithms.

A new summarizer based on cuckoo search algorithm is proposed for resolving muti-document summarization problem is proposed in (Rautray and Balabantaray 2018). In their objective function conception, the authors try to cover a set of objectives when building the summaries. The considered objectives include; readability, cohesion and non-redundancy. In their experiment, DUC 2006 and DUC 2007 dataset have been used. The proposed approach in compared with state of art summarization algorithms i.e., CSO and PSO using ROUGE-N as ROUGE-1 and ROUGE-2, readability and sentence similarity metrics. From three observations, the proposed cuckoo search algorithm based approach performs better, in most cases, when compared to CSO and PSO.

In (Selvan and Arutchelvan 2021b), the improved Cuckoo Search Optimization Algorithm (ICSA) is proposed to resolve MDS issue. The intensification strategy is enhanced by the inclusion of the simulated annealing and orthogonal concepts. Cohesion, readability and coverage are adopted to evaluate the obtained summary quality. Two benchmarked corpora where included in the experiments; DUC2007 and DUC2006. The proposed approach, when compared with other similar works, achieves better results with 0.09582 of F-measure, 0.10010 of recall and 0.09387 of precision.

A Multi-document Summarization approach based on the algorithm Social Spider Optimization (SSO) is proposed in (Selvan and Arutchelvan 2021a). The experimental study is conducted on two datasets namely, DUC2006 and DUC2007. The adopted evaluation metrics include ROUGE-1 and ROUGE-2 in addition to accuracy and F-measure. The performance of the proposed approach is compared with similar summarization works which use PSO and harmony search algorithms. The evaluation results indicate that the SSO-based summarization approach outperforms both PSO and harmony search algorithms in terms of ROUGE-1, ROUGE-2, accuracy and F-score metrics. From the used methods, PSO-based summarization performs the worst in terms of the different used metrics.

In (Tomer and Kumar 2022), a new MDS approach based on the firefly algorithm is proposed; to measure the quality of the resultant summary, the authors used a new fitness function composed of three factors; topic relation factor, cohesion factor and readability factor. The benchmark datasets used in the experiments are DUC2002, DUC2003 and DUC2004. The performance of the proposed summarization approach is compared with other

summarization works which use nature-inspired algorithms such as genetic algorithm (GA) and PSO. The evaluation criteria used in the experiments are the ROUGE score. The results show that this approach is more efficient than other approaches.

3. Discussion

Statistical-based techniques have the following advantages: they demand fewer computer resources (memory and processing), they do not require linguistic pre-processing, and they are language independent. The quality of summaries in the statistical-based approaches, on the other hand, is weak since some similar statements may have high scores while other crucial ones receive low scores. Machine-learning-based techniques to multi-document summarization can provide summaries that are appropriate for human readers' styles and can be prepared based on user needs. However, machine-learning systems necessitate a large number of manually written summaries in order to enhance sentence selection. Graph-based approaches leverage the inherent structure of documents; utilize centrality measures for sentence importance, and offer scalability and language independence. However, graph-based approaches are susceptible to inaccuracies in capturing semantic relationships and dependencies between sentences due to their high reliance on the quality of the graph structure. Certain algorithms' computational complexity and susceptibility to noisy data can lead to scalability challenges and potentially suboptimal summaries. Additionally, establishing meaningful cross-document relationships is a challenge, and fine-tuning parameters for optimal performance adds complexity to their implementation. Optimization-based techniques have recently acquired popularity due to their capacity to handle the multi-document Summarization problem and optimise many criteria, including redundancy reduction and content coverage. Various works, such as those using: Particle Swarm Optimization (PSO), Ant Colony Optimization (ACT) Shark Smell Optimization, Fuzzy Evolutionary Optimization Model, Cat Swarm Optimization, Cuckoo Search Optimization, Social Spider Optimization, and Firefly Algorithm to navigate the solution space, rapidly extracting pertinent information, demonstrate the effectiveness of swarm intelligence in solving the multi-document summarization problem.

4. Conclusion

In conclusion, this chapter has offered a comprehensive overview of the state of the art in multi-document summarization. The exploration commenced with an overview of the sequential steps integral to automatic multi-document summarization, encompassing pre-processing and the criteria employed in extractive techniques. Subsequently, a thorough examination of various approaches to extracting multi-document summaries was presented, followed by an in-depth discussion of these methodologies.

After thoroughly examining Multi-Document Summarization (MDS) and its methodologies, stages, and research from the past two decades, we found that extractive approaches, particularly those utilising optimisation algorithms, were the most prevalent. Approaches which consider the MDS as an optimisation problem and employ metaheuristic algorithms have proven their efficacy with encouraging outcomes. The upcoming chapter will focus on examining metaheuristic algorithms to offer an optimisation approach as a resolution to the MDS problem.

Chapter 03

Optimization metaheuristics

1. Introduction

The search of efficient and effective solutions to difficult problems has led to the development of novel approaches in the field of optimisation. This chapter dives into the use of metaheuristic approaches as an effective method for tackling optimisation challenges. With difficulties spanning domains as diverse as operations research, engineering, and computational biology, the demand for effective problem-solving tools has grown. Metaheuristics, which are distinguished by their capacity to traverse large solution spaces and solve complex combinatorial optimisation problems, emerge as attractive contenders to meet these tough challenges.

This chapter provides an overview of metaheuristics for solving challenging optimisation problems. We commonly identify two types of metaheuristics: approaches based on iterative development of a unique solution (classic examples are the Tabu method and Simulated Annealing) and those that simultaneously modify a full population of solutions (for example, genetic algorithms, the ant colony method, particle swarm optimisation, and so on). We have highlighted techniques that employ swarm intelligence in order to comprehend it and provide a solution for the problem of multi-document text summarization that is based on a new algorithm of this type.

2. Optimization problem

Within the context of an optimisation problem, the objective is to identify the optimal solution from among a number of feasible solutions (Tsai and Chiang 2023). The goal is to maximise or minimise a specific criterion, such as profit, efficiency, or cost, while remaining within set restrictions. Optimisation problems are common in a variety of domains, including operations research, engineering, finance, and AI.

A typical optimisation scenario includes a defined objective function that quantifies the quality of a solution, as well as a set of constraints that limit the number of potential

alternatives. The goal is to find the optimal solution that maximises or minimises the objective function while taking into account the limitations.

Formally, an optimization problem can be defined as follows(Tsai and Chiang 2023):

- ✓ **Minimization Problem:**

  Minimize f(x) subject to $x \in X$

- ✓ **Maximization Problem:**

  Maximize f(x) subject to $x \in X$

Where:

- ✓ f(x) is the objective function that needs to be either minimized or maximized.
- ✓ x is the vector of decision variables representing a potential solution.
- ✓ X is the feasible set, defining the constraints that the solution must satisfy.

Optimisation problems take many different forms, ranging from linear programming problems with linear constraints and objectives to nonlinear optimisation problems with complicated, nonlinear interactions. These problems are frequently addressed using mathematical programming approaches, heuristic methods, or metaheuristic algorithms. In the remainder of this chapter, we will investigate various metaheuristic algorithms.

3. Definition of metaheuristic algorithm

Various definitions of metaheuristic can be found in the literature. Here we present two definitions:

Definition 1: "A metaheuristic is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. In other words, a metaheuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem." ("" Http://Www.Metaheuristics.Net," n.d.).

Definition 2: "A metaheuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may

manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a constructive method." (S. Voß, S. Martello 1999).

After these definitions, we can summary that a metaheuristic is an algorithm that may effectively guide and lead the search process in a solution space, which is typically vast and contains many regions with optimal solutions. By abstracting and generalising this strategy, it becomes applicable to a wide range of subjects. Metaheuristics enable the discovery of solutions for certain applications, which may not always be optimal, but are extremely close to the optimum and can be found within an acceptable time. Metaheuristic algorithms use intensification and diversification as crucial strategies to augment their efficacy in solving optimisation problems.

4. Intensification and diversification

Metaheuristics adopt two principal strategies; exploration or diversification versus exploitation or intensification. In diversification, the algorithm seeks to generate a diversified solution in the goal of search space exploration. While in the intensification, the algorithm exploits the best obtained solutions to focus the search on promising regions (X. S. Yang and He 2013).

The effectiveness of an optimisation algorithm frequently depends on the efficient interaction between intensification and diversification. Intensification enables the algorithm to effectively utilise and improve favourable solutions, while diversification guarantees that the algorithm investigates a wide range of solutions, hence minimising the possibility of prematurely converging to poor solutions. The balance between these two strategies improves the algorithm's adaptability, flexibility, and capacity to discover better solutions across a wide range of optimisation problems.

5. Classification of metaheuristics

Combinatorial optimization problems are frequently extremely complicated, with exact solutions taking a long time or even impossible. The application of heuristic approaches allows for the generation of high-quality solutions in a reasonable time period. Heuristics are

also very useful for the development of exact methods based on evaluation and separation techniques, for example Branch and Bound algorithm (Narendra and Fukunaga 1977).

A heuristic is an algorithm that seeks a feasible solution without guaranteeing optimality, contrary to exact techniques, which guarantee exact solutions. Because exact solution methods to solve difficult problems have exponential complexity, it may be better to apply heuristics to generate an approximate solution or to accelerate the exact solution process. A heuristic is usually created for a specific problem, although approaches may include more general concepts; in this case, we refer to metaheuristics.

A common way to categorize metaheuristics is to differentiate between those that function with a population of solutions and those that only handle one solution at a time. Local search methods or trajectory methods are approaches that iteratively try to enhance a solution. The Tabu method, Simulated Annealing, and Variable Neighbourhood Search are common examples of trajectory methods. These strategies build a trajectory in the solution space by attempting to move towards optimal solutions. Genetic Algorithms, Particle Swarm Optimization, and Ant Colony Optimization are examples of approaches that utilize a population of solutions for optimization. Figure 4 shows the taxonomy of metaheuristics.

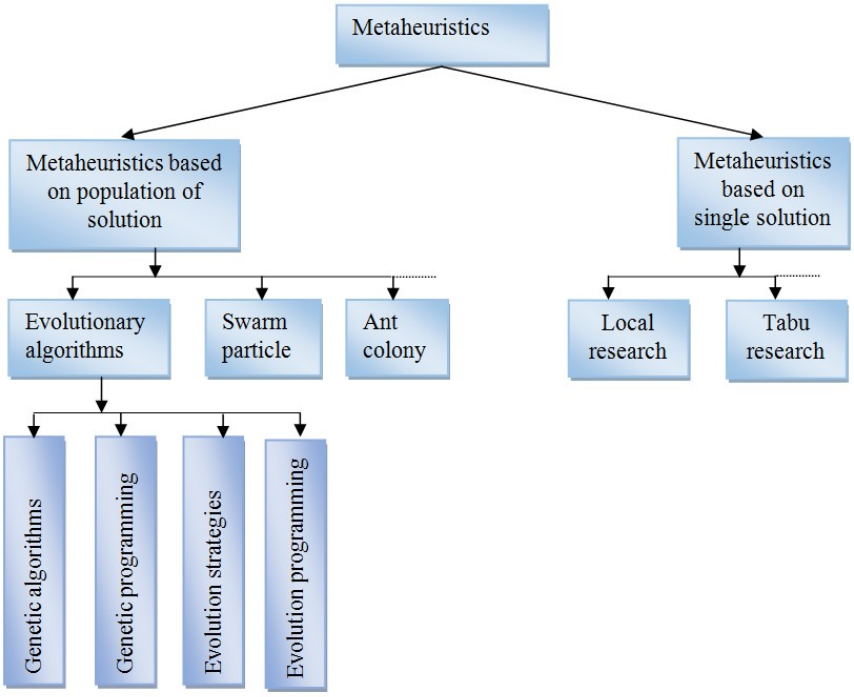

Figure 4: Taxonomy of metaheuristics.

5.1. Single solution based approaches

This section focuses on metaheuristics that are based on single solution, frequently referred to as trajectory approaches. In contrast to population-based metaheuristics, these methods start with a single initial solution and gradually move away from it, so creating a trajectory in the search space. The most common trajectory methods include descent method, simulated annealing, tabu search, GRASP method, variable neighbourhood search, iterated local search, and their variants. Each of these strategies aims to guide the search towards optimal or nearly optimal solutions by exploring the solution space in a sequential manner.

5.1.1. The descent method (DM)

The descent method (DM) is one of the simplest approaches in the literature. It is also known as hill climbing. The idea behind it is to select a point near the current solution (neighbour) that strictly improves the fitness function at each iteration, starting with an initial solution. There are several methods for selecting this neighbour: randomly selecting a neighbour from among those who improve the existing solution (first improvement), or selecting the best neighbour who improves the current solution (best improvement). When no neighbouring solution improves the current solution, the stop criterion is reached (Bonabeau, Dorigo, and Theraulaz 1999).

One of the fundamental drawbacks of the Descent Method is that it is susceptible to become stuck in a local optimum, so failing to explore the whole solution space. Variations of the Descent Method, such as the Multiple Start Random Hill Climbing algorithm, are used to overcome this problem. These methods entail beginning the optimisation process many times from diverse initial solutions in order to increase the possibility of obtaining the global optimum rather than settling for a local one.

5.1.2. Tabu Search algorithm

The Tabu Search algorithm is a search strategy that was developed by Fred Glover (Glover 1986) . It varies from ordinary local search methods in that it uses a history of previously visited solutions to make the search less random. It is thus conceivable to escape from a local minimum, but certain solutions are considered taboo in order to avoid falling back into it periodically. The novelty of the approach lies in the incorporation of a memory by using

taboos. This involves recording the events that occurred in previous steps and taking strategies to avoid them from occurring again. The obvious objective is to prevent settling into or going back to a local optimum too soon and to encourage a wide exploration of the space of solutions.

The objective of the taboo search method is to maintain a taboo list L, with a specific length l, including the solutions that have been recently visited. Each time, a new solution is selected; it is added to the taboo list. If the list of taboo is too large, we eliminate the oldest solution, so making it no longer taboo. The fact of creating a taboolist is to avoid being trapped and thus going in circles, and that is why we create the list L which memorizes the last solutions visited and which prohibits any movement towards a solution of this list. This list L is called the Taboo list; it is the memory of the search and the knowledge of exploring the space of solutions.

5.2. Metaheuristics with population of solutions

Compared to algorithms that start with a single solution, metaheuristics in this case, use a population of solutions to improve and progress a set of solutions through multiple iterations. There are two main groups within this category: evolutionary algorithms, which are based on Charles Darwin's theory of natural selection (Darwin 1876), and swarm intelligence (SI) algorithms. The SI, similar to evolutionary algorithms, takes inspiration from the similarities with natural biological phenomena.

5.2.1. Evolutionary algorithms

Evolutionary algorithms, often known as evolutionary computation (EC), are a collection of algorithms which take inspiration from the theory of evolution in order to address diverse problems. Based on Charles Darwin's theory (Darwin 1876), the evolution of species happens due to the combination of two phenomena: firstly, natural selection, which favors the survival and reproduction of individuals that are most adapted to their environment, thereby passing on their genes to future generations; and secondly, the existence of random genetic variations (mutations) within species.

In brief, evolutionary algorithms encompass fundamental tasks essential to their functioning. These tasks, common to most classic instances of evolutionary algorithms, involve the iterative evolution of a population. The process includes the initialization of a population

representing potential solutions, the evaluation of each individual's fitness based on the optimization objective, and the subsequent selection of individuals for reproduction. Genetic variation is introduced through crossover and mutation operators applied to selected parents, generating new offspring. The offspring are then incorporated into the population, and this cyclical process continues through multiple generations. When a stopping condition, such as a maximum number of generations or a maximum number of evaluations, is met, the reproduction process comes to an end. Figure 5, shows the basic tasks of the evolutionary algorithm (EA). The concept of evolutionary computation covers a wide range of metaheuristics, including genetic algorithms (Holland 1975), evolution strategies (Vent 1975), evolutionary programming (Fogel 1998), and genetic programming (Koza John 1992).

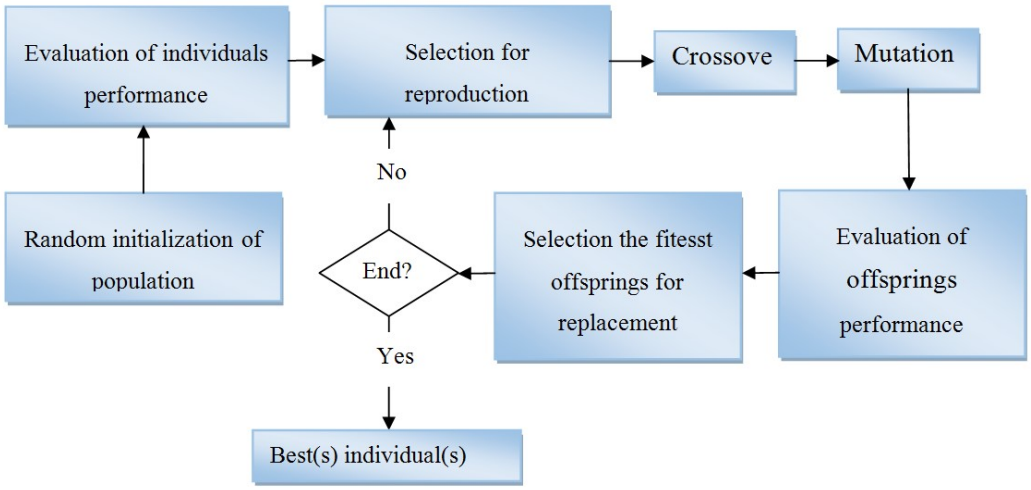

Figure 5: Basic tasks of the evolutionary algorithm (EA).

a. Genetic algorithms

Genetic algorithms, (Goldberg 1989)(Eiben and Smith 2015) are based on evolutionary theory and genetic principles that control how organisms adapt to their environments. The following mechanisms are included in these algorithms:

- ✓ Natural selection: Individuals that are most adapted to their environment are more likely to survive for a longer amount of time, improving their probability of reproducing.
- ✓ Reproduction via crossover includes the transmission of an individual's characteristics from its parents. Hence, the crossing of two individuals that are highly adapted to their environment tends to produce an offspring that is similarly well adapted to this particular environment.

- ✓ Mutation refers to the random appearance or disappearance of particular characteristics, which might provide new abilities for adapting to the environment. These abilities can then propagate through selection and crossover mechanisms.

Genetic algorithms utilise these mechanisms to construct a meta-heuristic. The process entails the progressive development of a population of combinations through the mechanisms of selection, crossover, and mutation. The evaluation of a combination's adaptation capacity in this context is determined by the fitness function that needs to be optimised.

b. Genetic Programming (GP)

The technique of genetic programming originated from the research conducted by Koza in 1992 (Koza John 1992). It is an algorithmic approach for generating computer programs based on a high-level problem description, specifying what is wanted. The distinctive characteristic of GP consists in its use of tree-based representations of programs and programming languages that have a simple syntax. These languages allow for easy syntactic manipulations that consistently result in valid programmes.

The starting population is formed using a random method of selecting trees that represent the programmes. A depth limit or maximum number of nodes is typically specified to restrict the size of randomly generated trees. The three primary operators utilised in GP are: mutation, crossover, and reproduction. Additional transformation operators can be incorporated, such as permutation, editing, or encapsulation (Koza John 1992).

- ✓ Permutation operators: involve changing the order of sub-trees or nodes within a tree. This can introduce variations in the structure of the programs.
- ✓ Editing operators: focus on modifying the structure of a tree by adding, deleting, or rearranging nodes. They contribute to the exploration of different program structures.
- ✓ Encapsulation operators: involve grouping certain nodes into subprograms or functions. This introduces modularity and can facilitate the reuse of code within the evolving programs.

By combining these operators, Genetic Programming explores the space of possible programs to find solutions to a given problem. The iterative application of these operators over generations allows GP to evolve increasingly fit programs based on a specified fitness criterion.

c. Evolutionary programming (EP)

Evolutionary Programming (EP) was first developed in the early 1960s by L. J. Fogel as an evolutionary method for artificial intelligence, specifically for solving learning problems using finite automata (L. J. Fogel, A. J. Owens 1966). Subsequently, in the 1990s, D. B. Fogel adopted this methodology to tackle more general problems (Fogel 1991). The distinguishing characteristic of these algorithms is their only use of mutation and replacement operators, without relying on crossover operators. This property distinguishes them apart from other evolutionary techniques, which typically use crossover operators. Evolutionary Programming techniques iteratively improve solutions by introducing random changes and replacing less effective individuals with newly created solutions. Although this method may seem uncomplicated, it can be extremely effective for specific types of problems, especially those in which the connections between solution elements are not adequately represented by conventional crossover operators.

The EP method is not widely utilised in comparison to other algorithms within the same category due to its similarity to an ES. However, it was developed independently and incorporates a replacement strategy that is more stochastic than deterministic, allowing even the worst individuals a small chance of survival (Back 1993).

d. The Evolution Strategy

The Evolution Strategy was initially proposed by Rechenberg in the 1960s (Rechenberg 1965) and further improved by Schwefel (Schwefel 1981). The most basic algorithm, known as the two-membered ES or (1 + 1)-ES, operates on a single individual. At each generation, the algorithm creates a new individual by mutation from the parent individual and chooses one or the other to keep in the population. In order to incorporate the concept of population, which has previously only been utilised in the simplified version of Evolution Strategies (ES), Rechenberg suggests the implementation of multi-membered ES (or $(\mu + 1)$-ES), where more than one parent ($\mu > 1$) can contribute to the creation of an offspring.

These algorithms incorporated recombination as a result of the inclusion of many parents. Schwefel (Schwefel 1981) presents two further variations of multi-membered evolutionary strategies, which are determined by the selection technique employed, namely $(\mu + \lambda)$-ES and (μ, λ)-ES. The process of creating a new population involves producing λ individuals from μ parents and selecting only the top μ individuals. This can be done either by considering only

the descendants (in which case λ must be bigger than μ, known as the (μ, λ)-ES scheme), or by considering both the descendants and their parents (known as the (λ + μ)-ES scheme). Two other recognised variations of ES are (μ/ρ + λ)-ES and (μ/ρ, λ)-ES. The parameter ρ represents the number of parents contributing to the reproduction of an individual offspring.

5.2.2. Swarm intelligence

The computational complexity of optimization problems leads researchers to seek efficient methods for search space exploration. Bio-inspired metaheuristics inspire their behaviour from nature to solve real-world problems (Parpinelli and Lopes 2011). The optimization power of swarm-intelligence (SI) methods makes it an essential branch in the artificial intelligence (IA) field (Chakraborty and Kar 2017). SI methods use the collective behaviour of birds, fishes, ants and different species to solve complex optimization problems. Self-learning, flexibility, adaptability and versatility among others are the most important reasons behind the immense success of bio-inspired metaheuristic methods (Valdez 2021). The utility of swarm-based methods comes from finding a reasonable solution for an NP-hard problem using a limited number of parameters and within acceptable computational time.

Swarm Intelligence (SI) evolved from mathematical and computational modeling of biological phenomena observed in ethology (Bonabeau, Dorigo, and Theraulaz 1999). It includes a set of algorithms based on a population of agents or entities able to executing specific tasks and interacting locally with each other and with their environment. Despite the fact that individual agents have limited capacities, their collective interactions enable the achievement of complicated tasks that are critical for existence. Although there is no centralized control structure that dictates how individual agents should behave, local interactions between agents often lead to the emergence of collective behavior and self-organization.In the following, we will examine the two crucial algorithms in this category: Ant Colony Optimisation (ACO) and Particle Swarm Optimisation (PSO).

a. Ant Colony Optimisation (ACO)

Marco Dorigo and other researchers originally proposed Ant Colony Optimization (ACO) (Dorigo 1992),(Dorigo, Maniezzo, and Colorni 1996). It is based on the behavior of real ants searching for food (Deneubourg et al. 1990). In spite of lacking highly developed individual cognitive capacities, they manage to discover the shortest path between their nest and a food source. The ants begin by randomly walking around their colony to inspect the environment.

They release a volatile chemical substance known as a pheromone on the ground along their path between the food source and the nest in order to identify particular favourable paths that should assist the other ones to the food source (Dorigo and Blum 2005). After a period, the shortest path from the nest to the food source contains more pheromone and thus attracts more ants. Artificial ant colonies use this trait to create solutions to optimization problems. They construct solutions by browsing a complete graph G_C (V, E), where the nodes V are solution components and the edges E are connections between the components (The concept of a solution component varies according to the problem at question. In the Travelling Salesman Problem (TSP), a city added to a path is a solution component (Lawler, Lenstra, and Kan 1985)). The ACO general algorithm is divided into three steps for each iteration:

Construct Ant Solutions. During this phase, a group of ***m*** artificial ants iteratively generate solutions from the collection of potential solution components C = $\{c_i^j\}$ (i = 1, . . . ,n ; j = 1, . . . , $|D_i|$). A solution is constructed by starting with an empty partial solution, denoted as $s_p = \emptyset$. During each step of the construction process, the current partial solution s_p grows by including a solution component c_i^j from the set of feasible neighbours $N(s_p) \subseteq C$. The selection of a component in $N(s_p)$ is determined probabilistically at each stage of the construction process. Each component c_i^j belonging to the set $N(s_p)$ has a probability $p\left(c_i^i \middle| s_p\right)$ to be selected. The decision is affected mainly by the quantity of pheromone τ_{ij} associated with each element of $N(s_p)$, as well as the heuristic information on the problem. The most frequently employed rule for this stochastic choice is that developed originally for the Ant System (AS) (Dorigo 1991) and given by equation (23).

$$p\left(c_i^j \middle| s_p\right) = \frac{\tau_{ij}^{\alpha}.[\eta(c_i^j)]^{\beta}}{\sum_{c_i^j \in N(s_p)} \tau_{il}^{\alpha}.[\eta(c_i^l)]^{\beta}} \text{ , } \forall\, c_i^j \in N\left(s_p\right) \qquad (23)$$

The heuristic information, denoted by the function η (.), assigns a heuristic value to each component of the feasible solution $c_i^j \in$ N (sp). The ants use this value to make probabilistic determinations concerning their movement within the construction graph. The parameters α and β define the impact of the pheromone values (respectively, heuristic values) on the ant's decision-making process (Dorigo and Stützle 2019). Their role is to counterbalance the algorithm in a phase of intensification or diversification.

Daemon actions. In the context of ant colony optimisation (ACO), daemon actions are problem-specific or centralised activities that cannot be performed independently by each ant. Typically, these operations involve a local search among constructed solutions. Only locally optimised solutions are used to update the pheromone traces in this procedure. Daemon actions play an important role in enhancing the ACO algorithm's exploration and exploitation capabilities by introducing problem-specific information or heuristics to guide the ants towards better solutions.

Update Pheromones. In the context of ant colony optimisation (ACO), the term "Update Pheromones" refers to the act of updating or modifying the pheromone levels on the solution graph's edges. This critical step in the ACO algorithm consists of two major stages. The first stage involves a reduction of all pheromone values by evaporation. The second is an increase in the pheromone values associated with a set of good solutions, which is commonly referred to as S_{upd}. This increase is accomplished by a mechanism known as pheromone deposition. The Update Pheromones mechanism is crucial in guiding artificial ants towards better solutions because higher-quality pathways accrue more pheromones, influencing the algorithm's overall exploration-exploitation balance. The specific implementation of Update Pheromones can differ based on the ACO algorithm employed and design problems. The implementation of this update typically follows the method described by Dorigo and Thomas in 2010 (Dorigo and Stützle 2019), the update is represented by equation (24):

$$\tau_{ij} = (1-\rho)\tau_{ij} + \sum_{s \epsilon S_{upd} | c_i^j \epsilon\, s} g(s) \qquad (24)$$

Where $0 < \rho \leq 1$ is the rate of evaporation of pheromones and $g: S \rightarrow R^+$ is a function such that: $f(s) < f(s') \Rightarrow g(s) \geq g(s')$. The function g is commonly called quality function.

b. Particle Swarm Optimization

Russell Eberhart and James Kennedy introduced Particle Swarm Optimisation (PSO) in 1995 (Kennedy and Eberhart 1995). PSO was inspired by the collective movements observed in social animals, particularly fish and bird migrations. These animals tend to imitate successful behaviours observed in their peers while adding their own variations. The idea behind PSO comes from the work of C. Reynolds (Reynolds 1987) and Heppner and Grenander (Reynolds 1987). They made mathematical models to show how birds fly together and fish school together. PSO uses these principles to develop a population-based optimisation method that

iteratively improves solutions by exploiting the collective dynamics of particles in a search space.

The particle swarm consists of a group of agents referred to as "particles". Every particle, represented as a potential solution to the optimisation problem, moves across the search space in search of the best possible solution. The movement of a particle is influenced by three components (refer to figure 6): (1) A physical component: The particle has a tendency to continue moving in its current direction. (2) A cognitive component: The particle has a tendency to move towards the best location it has previously visited. (3) A social component: The particle relies on the experiences of its peers and therefore moves towards the best location that its neighbours have already reached.

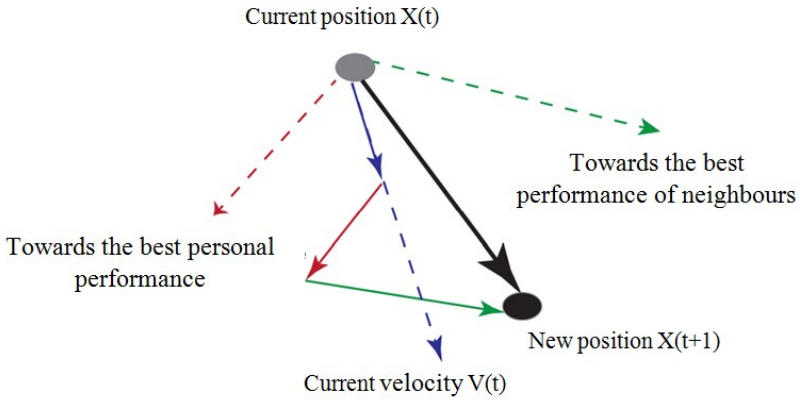

Figure 6: Movement of a particle

The notion of neighbourhood in particle swarm optimisation (PSO) can be delineated either in terms of physical proximity or sociometric connections. Spatial definition entails the examination of the Euclidean distance between the positions of two particles within the search space. Alternatively, the sociometric definition entails using the position of an individual particle within the total swarm as a way to quantify its interaction with its neighbouring particles (Kennedy and Mendes 2002). The definition of a neighbourhood is essential in determining the social aspect of a particle's mobility. It affects the particle's behaviour by considering the experiences and positions of its adjacent peers in the swarm. The selection between spatial and sociometric neighbourhood definitions allows for adaptability in applying PSO to various optimisation scenarios.

Each individual particle i in the swarm is defined by its position, denoted as $\overrightarrow{X_i}$, and its velocity or speed, represented by a position change vector, $\overrightarrow{V_i}$. Each particle possesses a

memory that enables it to keep information about its most optimal solution previously identified, referred to as $\vec{P_i}$ (personal best), as well as the best known position in its neighbourhood, denoted $\vec{P_g}$ (global best). At each iteration, every particle navigates through the search space by following a vector. This vector is determined by calculating a weighted sum of the vectors indicating the particle's current velocity $\vec{V_i}$, its personal best position $\vec{P_i}$, and the global best position $\vec{P_g}$. The new velocity $\vec{V_i}$(t+1) is determined in the equation (25)

$$\vec{V}_i(t+1) = \omega\vec{V}_i(t) + C_1\varphi_1\left(\vec{P}_i(t) - \vec{X}_i(t)\right) + C_2\varphi_2\left(\vec{P}_g(t) - \vec{X}_i(t)\right) \quad (25)$$

Where i = 1, 2. . . N, and N is the number of particles (swarm size); the inertia coefficient ω makes it possible to control the influence of the velocity obtained in the previous step.

A large inertia factor creates huge amplitude of movement, while a small inertia factor focuses the search on a limited space. φ1 and φ 2 are two random values selected uniformly from [0,1], and C1 and C2 are two constants representing a positive acceleration. They correspond to the cognitive and social component (respectively) of the movement. Each iteration updates the position of each particle, as shown in the equation (26).

$$\vec{X_i}(t+1) = \vec{X_i}(t) + \vec{V_i}(t+1) \quad (26)$$

To avoid particles from moving excessively fast in the search space and perhaps overlooking the optimal solution, it may be essential to establish a maximum speed (referred to as V_{max}). This ensures that each component of $\vec{V_i}$. maintains within the range of [-V_{max}, +V_{max}] (Eberhart, Simpson, and Dobbins 1996). Selecting the optimal V_{max} value is a complex task, as it significantly affects the balance between exploration and exploitation.

The ant Colony Optimisation (ACO) and basic Particle Swarm Optimisation (PSO) play critical roles in building the foundation of swarm intelligence. Marco Dorigo introduced ACO in the early 1990s (Dorigo 1992), revolutionising the area by demonstrating how decentralised agents, inspired by ant foraging behaviour, can collectively solve complicated optimisation problems through indirect communication via pheromone trails. Similarly, Russell Eberhart and James Kennedy presented basic PSO in 1995 (Kennedy and Eberhart 1995), drawing inspiration from the social behaviour of birds and fish to demonstrate how a population of particles, led by personal experience and peer influence, efficiently explores and exploits

solution spaces. Together, these seminal models demonstrate the power of decentralised, swarm-based techniques to addressing difficult optimisation issues, inspiring the development of varied swarm intelligence algorithms and making major contributions to the broader landscape of artificial intelligence.

6. Conclusion:

To summarise, this chapter offers an examination of metaheuristics used to address complex optimisation problems. The scope of our discussion includes the essential concepts of metaheuristics, such as their definition, the types of problems that necessitate the use of metaheuristic approaches, and the principles of intensification and diversification. In addition, we explore the two main categories of metaheuristics: those based on the iterative improvement of a single solution and those focused on a population of solutions.

During this investigation, we have thoroughly examined the current state of optimisation metaheuristics. We have categorised the leading algorithms into two main types: those that produce a single solution and those that are fundamentally based on the concept of a population of solutions. This chapter establishes the foundation for comprehending the many methodologies in the field of metaheuristics, preparing for later discussions on particular algorithms and their utilisation in addressing complex problems.

Zhang et al. (Q. Zhang et al. 2019) have introduced a new swarm-based optimisation method known as the biology migration algorithm, which is inspired by the biological migration phenomena of many species such as insects, mammals, fish, and others. This algorithm has not previously been applied to the MDS problem. In the following chapter, we will provide a thorough explanation of this algorithm, outlining its operation and demonstrating its use in addressing the MDS problem

Chapter 4

BBMA-MDS: Binary biology migration algorithm for multi document text summarization

1. Introduction

In this chapter, we introduce our approach dedicated to multi-document text summarization, called BBMA-MDS (Binary Biology Migration Algorithm for Multi-Document Text Summarization) (Boussalem et al. 2023). The chapter is structured into two distinct sections.

The first section focuses on the presentation of our approach, based on a new swarm intelligence algorithm, BMA(Q. Zhang et al. 2019). We start by modelling the MDS problem as an optimization problem, propose an objective function that measures the quality of the generated summary, and outline the original BMA algorithm. The adaptation of BMA to our MDS problem is then detailed, highlighting the addition of a binarization step. We then describe the key steps of our approach, including pre-processing, the intermediate representation of the documents to be summarized, and finally the summary generation phase. We conclude this section by presenting the final algorithm of our approach.

The second section is dedicated to the experimental results of our approach. We detail the datasets used, the evaluation metrics employed, discuss the parameters of the algorithm and the objective function. Comparisons with state-of-the-art approaches are made, and at the conclusion of this part, we analyze the results obtained by our approach.

2. Problem formulation

The MDS is the process of automatically selecting the most important sentences from the original documents to create a compressed version, which provides useful information for the

reader. In this paper, we aim to consider the MDS as a combinatorial optimization problem and proposing a solution based on a metaheuristic algorithm.

The input of our approach is a collection of documents;

$CD = \{D_1, D_2,...,D_N\}$, where; D$i$ is i^{th} document and n denotes the number of documents in the collection. Each document Di is composed of a set of sentences $D_i = \{s_{i1}, s_{i2}, s_{i3},...,s_{im}\}$. The sets of sentences of documents are fusioned to obtain the final set $CD = \{s_1, s_2,s_3,...,s_D\}$, where; D denotes the number of sentences in CD. The output is a summary Sum from the input CD, where Sum is a subset of sentences selected from the original CD with a size under a fixed threshold T of number of words, as shown in Equation (27).

$$Sum = \{s_1, s_2, ... ,s_j\} / s_j \in CD \ , \ \text{Size}(Sum) <= T \quad (27)$$

There are a huge number of combinations of sentences (summary or solution) that satisfy Equation (27) called feasible solutions and presented in Equation (28).

$$\text{Feasible solutions } (Sum_{candidate}) = \{Sum_1, Sum_{2,}....,Sum_k\} \quad (28)$$

K is the number of all possible solutions, and Sum_i is given in Equation (27).

Our objective is to find the best solution among the feasible solutions. The best summary is the one that has the high quality expressed by a fitness function, explained as follows.

2.1. Quality of the summary

The quality of a summary depends on three criteria; coverage, cohesion and readability. Which are normalised between 0 and 1.These criteria are detailed below:

The coverage is presented in Equation (29)**,** it represents the content coverage of all documents in CD. In other words, it checks if the summary takes in account the continent of all documents. The summary Sum_i with the highest coverage is considered to be the best summary.

$$Coverage = \sum_{i=1}^{N} \sum_{j=1}^{n} \frac{sim(s_i, s_j)}{n-1} \quad (29)$$

N is the number of sentences in CD and n represents the number of sentences in the summary.

The cohesion represents the connection between sentences in the summary Sum_i, as presented in Equation (30); taking in account this factor; a good summary is expected to have a high cohesion value.

$$Cohesion = \frac{\sum_{i,j=1}^{n}(1 - sim(s_i, s_j))}{(n * (n-1)/2)} \qquad (30)$$

Where *n* gives the number of sentences in the summary, i, j =1....n, and j >= i.

In a readable summary, each sentence should be related to the sentence following it. Readability is given by the Equation (31):

$$Readability = \frac{\sum_{i=1,j=i+1}^{n} sim(s_i, s_j)}{n-1} \qquad (31)$$

Where *n* is the number of sentences in the summary; s_i, s_j are two consecutive sentences in the summary.

In what follows, we use these three criteria in a single formula to express the fitness function that measures the quality of the summary.

2.2. Fitness function

The fitness function expresses the quality of the summary by the formula (32). The used fitness function is a weighted sum of the aforementioned three criteria; coverage, cohesion and readability given in Equations (29), (30), and (31) respectively.

$$Fitness = \propto * Coverage + \beta * Cohesion + \gamma * Readability \qquad (32)$$

Where $\propto, \beta, \gamma \in [0,1]$, and $\propto + \beta + \gamma = 1$.

A good summary has a high combination value between coverage, connection between sentences, readability and cohesion. The approach searches for the optimal solution which has the maximum value of the fitness function.

Then, the MDS optimization problem can be formulated as follow:

Maximize (Fitness (Sum))

Where:

$$\begin{cases} \text{Sum is a summary of multi documents CD} \\ \text{CD is the input multi documents to be summarized} \\ \text{Size(Sum)} <= T \end{cases}$$

3. Original Biology Migration Algorithm (BMA)

Inspired by the biological migration phenomenon of different species such as insects, mammals, fish and others, Zhang et al. proposed recently a new swarm-based optimization algorithm, called biology migration algorithm (Q. Zhang et al. 2019)**.**

In BMA, the biological species (particles) represent the population that searches for solutions to the problem in their habitat (search space). As all swarm-based optimization algorithms, BMA starts the optimization process with a randomly initialised population as *N* D-dimensional real vectors in the range [0, 1], where *N* is the population size.

After the initialization, particles displace in the search space by generating a new generation at each iteration until a maximum number of iterations is met. The search process of BMA is processed in two main phases: migration phase and updating phase.

3.1. Migration phase

In this phase, species modify their position (they move from their current position toward a new position) according to two alternatives: On one hand, the best specie of the population as presented in Equations (33) and (34), and on another hand, its neighbourhood candidates as presented in Equation (35). The neighbourhood candidates of each particle are the two randomly selected particles from the population. Then, the migration phase can be mathematically given by Equations (33), (34) and (35).

$$X_i(t+1) = X_i(t) + \lambda * \sec(t) * L(t) * |X_{best} - X_i(t)| \qquad (33)$$

Where, $X_i(t)$ and $X_i(t+1)$ are the positions of the i^{th} particle at iterations *t* and t+1 respectively, λ is a random vector in the range [0, 1], *L(t)* is the step size defined in the Equation (8), X_{best} is the best position in the current iteration.

$$L(t) = 2 - 1.7\left(\frac{t-1}{T-1}\right) \quad (34)$$

Where, *t* is the current iteration and *T* is the maximum number of iterations.

$$X_i(t+1) = X_i(t) + ç * \left(X_j(t) - X_k(t)\right), i \neq j \neq k \qquad (35)$$

Where, $X_i(t)$ and $X_i(t+1)$ the positions of the i^{th} particle at iterations t and $t+1$ respectively, X_j (t) and $X_k(t)$ are two positions selected randomly from the population, ç is a random vector in the range [0, 1]

3.2. Updating phase

In the updating phase, the current position $x_i(t)$ and the new obtained position by applying the migration phase $x_i(t+1)$ are compared. If the new position $x_i(t+1)$ cannot improve the quality of the current solution within a predetermined number of cycles *C*, it will be abandoned and replaced by a new randomly generated position to explore other search zones.

4. Proposed MDS Approach

The MDS problem is known as an NP-hard problem because of the big size of texts and a large number of sentence combinations of sentences to be selected in the summary. Therefore, we need a long time to discover an optimal summary. It is possible to obtain a near-optimal summary in a reasonable time for such problems by using metaheuristic methods.

One of the advantages of metaheuristic algorithms is that they are problem-independent and have a good approach to solve problems in different domains.

In this section, we present in detail our proposed Approach. The proposed approach is named BBMA-MDS. The main steps of BBMA-MDS are showed in the flowchart of Figure (7)

4.1. Pre-processing

Pre-processing is an essential step in every text summarising process. In chapter 2, we have provided an in-depth discussion of the peculiarities of this specific phase. In the subsequent section, we provide a concise explanation of the employed techniques in this research, along with the corresponding notations.

4.1.1. Sentence segmentation

For each document *D* in the collection of documents to be summarized, the text is segmented into sentences. The result of this treatment is: $D=\{s_1, s_2, s_3,.....s_n\}$, where s_i represents the i^{th} sentence in the document *D*; n is the number of sentences in document *D*.

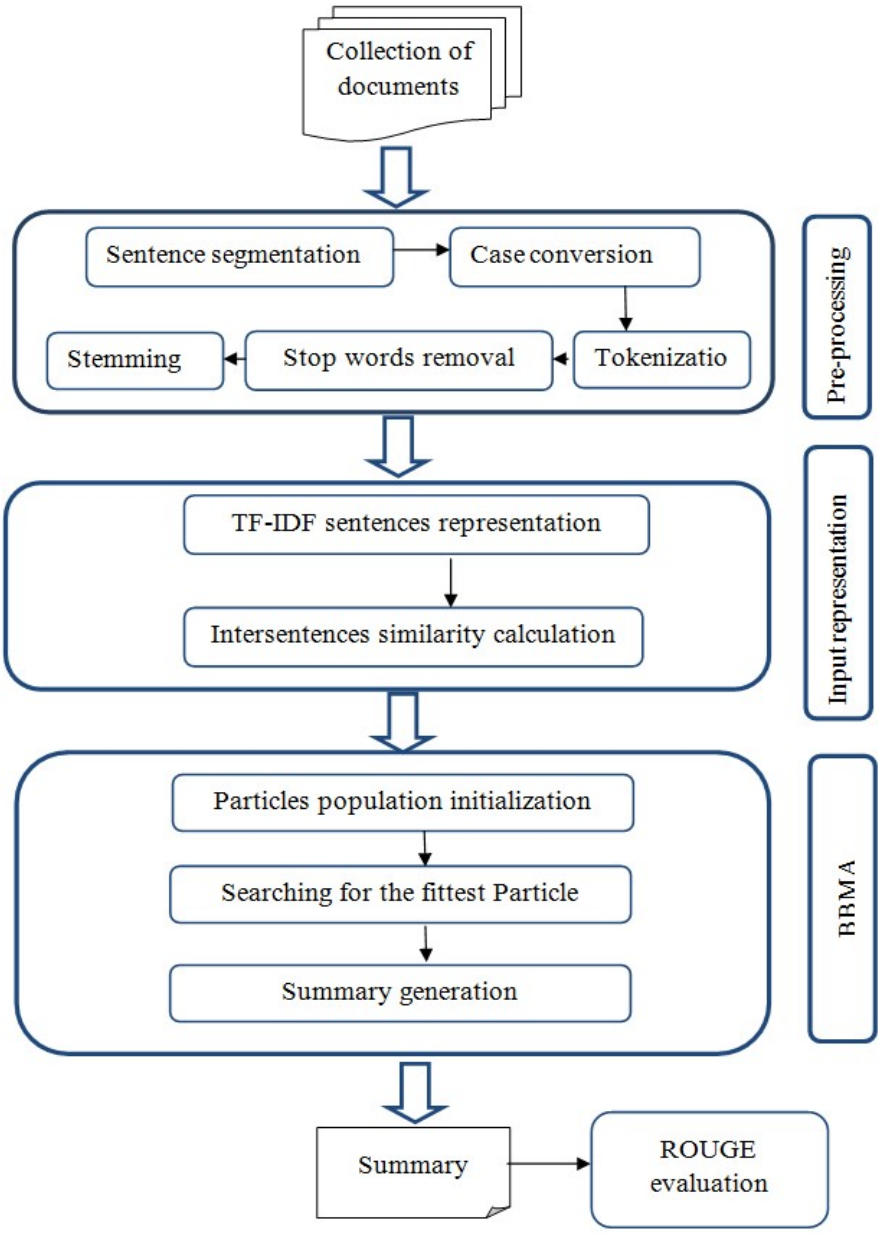

Figure 7: Main steps of BBMA-MDS

4.1.2. Case conversion

The text in documents should be in the same form, lower-case or upper-case. This unification helps in reducing the vector space, which allows the optimization of computing power and time. In our study the lower-case is considered.

4.1.3. Tokenization

Tokenization is an important step for preparing documents for the step of removing stop words and stemming, this phase consists of dividing the sentences into tokens. The result of this step is; s= $\{t_1, t_2, t_3, ..., t_k\}$, where t_j represents the j^{th} token of the sentence *s,* and *k* is the number of tokens in the sentence *s*.

4.1.4. Stop words removal

The objective of this step is to exclude stop words such as "a," "the," "then," "before," and others, as they are commonly used but do not hold significant importance for the summarization task.

4.1.5. Stemming

In this process, words that come from the same root, are identified and grouped into a single group and are replaced with their root.

4.2. Input representation

4.2.1. Sentences representation

TF-IDF stands for Term Frequency-Inverse Document Frequency; it is the most widely used method for representing text in digital form. The input of BBMA-MDS is a set of sentences of all documents in the collection; each sentence is represented in TF-IDF form, where:

The weights of terms of each sentence in TF-IDF representation are given in formula (36).

$$W_{ij} = TF_{ij} * IDF_i \qquad (36)$$

TF_{ij}represents term frequency of the term $t_i$in sentence s_j, it's given in formula (37).

$$TF_{ij} = \frac{freq_{ij}}{\|s_j\|} \qquad (37)$$

Where, $freq_{ij}$ is the occurrences number of the term $t_i$in the sentence s_j.

$\|s_j\|$is the number of all terms in the sentence s_j.

Inverse Document Frequency (IDF), see equation (38), is the relation between the total number of sentences N in the collection and the number of sentences n_i containing the term t_i.

$$IDF_i = \log_{10}(N/n_i) \qquad (38)$$

4.2.2. Inter-sentences similarity

After having represented sentences in TF-IDF form in Equation (36), we calculate the similarity between sentences using the following cosine similarity formula (39).

$$sim(s_i, s_j) = \frac{\sum_{k=1}^{m} w_{ik} * w_{jk}}{\sqrt{\sum_{k=1}^{m} {w_{ik}}^2} * \sqrt{\sum_{k=1}^{m} {w_{jk}}^2}} \qquad (39)$$

The Equation (39) is used in BBMA-MDS to calculate the fitness function.

4.3. Proposed Binary Biology Migration Algorithm (BBMA)

The MDS problem is binary in nature, while the original BMA has been designed for continuous problems. Therefore, the BMA requires a binarization process to adapt it to binary problems. Two main methods are used in the literature for metaheuristics binarization: transfer functions and the modified position equation method using new binary operators (Krause et al. 2013). In the first method, the metaheuristic operates in continuous space and when we need to evaluate each particle's position a transfer function is applied to map the continuous vector to a binary one. In the second method, all operators in the position updating equations of the algorithm are transformed to binary operators using several technics, and then the particles are initialised and displaced in the binary space. The use of transfer function is widely used in binarization process due to their promising results obtained in solving many optimization problems before (Rahab, Houassi, and Laouid 2022), (Hichem et al. 2022). Hence, in this work, the transfer function binarization technic is used to propose a binary version of the BMA algorithm.

4.3.1. Solution encoding and population initialization

In the proposed approach for the MDS problem, a summary is a combination of a sub-set of sentences selected from the original *CD*. This problem is known as a combinatorial optimization problem which is of exponential complexity, what we motivated to use metaheuristics because they are more adapted to these complex problems.

The algorithm starts with a set of solutions called initial particle's positions P= (p_1, p_2, …, P_N), where N is the number of particles (swarm size) and p_i is the i^{th} particle in the swarm. Each particle p_i has a positionp_i^t at a time t that represents a solution of the optimized problem. In the MDS problem, a solution (summary) is represented as S-sized binary vector, where S is the total number of sentences in the original documents, the$p_i^t[j]$ is the j^{th} element in the vector p_i^t. As our problem is to select or not select sentences in the summary (solution), each element $p_i^t[j]$ can take a binary value "zero" or "one"; the value "zero" means that the corresponding sentence (j^{th} sentence) is not selected in the solution, and the value "one" means that the corresponding sentence is selected in the summary. For example, in Figure 8, a particle's position encoding is presented. In this solution, the second, fifth, sixth and ninth sentences of the original *CD* are selected in the summary.

0	1	0	0	1	1	0	0	1	0	0	.....	0	0	0

Figure 8: Particle's position encoding

4.3.2. Position updating

The BMA algorithm starts by generating a random population of *N* particles in continuous space. Each particle in the population updates his position at each iteration (displaced in the search space) using Equations (33) to (35) and the fitness function is called to evaluate the quality of the position until the maximum number of iterations.

It is impossible to use the BMA directly to tackle the MDS problem, because of the binary nature of MDS. Therefore, before the fitness evaluation of each position, a new binary position generated from the continuous position is necessary for the algorithm to be suitable for the MDS problem. Thus, a binary version of BMA (BBMA) is proposed based on the transfer function to tackle the MDS problem as a binary optimization problem. The transfer function is called in BBMA-MDS to convert the continuous position to binary position of particles, so that making the algorithm suitable for the MDS problem.

Several transfer functions are used in the literature; however, the sigmoid function is widely used and it proves promising results. So, in this approach, the sigmoid transfer function is used to BBMA-MDS for the position transformation. Equations (40) and (41) present the sigmoid function.

$$X_{sig} = \frac{1}{1 + e^{-x}} \quad (40)$$

$$X_b = \begin{cases} 1, & Random \leq x_{sig} \\ 0, & Random > x_{sig} \end{cases} \quad (41)$$

Where X_b is the converted binary value of the real vector solution, and *Random* is a random number used as the threshold.

4.3.3. Outlines of the binary biology migration algorithm for MDS problem

In the Algorithm 1 the pseudo-code of the proposed BBMA-MDS is presented.

4.4. Summary generation

The final step of our approach is to produce the summary by retrieving the fittest particle after the maximum number of iterations. The particle is represented as a vector of *D* elements, each element corresponding to a sentence in the original *CD*. The summary is generated by comparing the particle vector with a pre-recorded list of sentences. If an element in the particle vector has a value of 1, the corresponding sentence in the list is included in the summary. Otherwise, the sentence is omitted out the summary.

5. Experiment and results

This section offers a comprehensive analysis of the experimentation conducted to assess the effectiveness of our approach. It discusses several aspects of the programming environment, including the hardware and software components, the datasets used, evaluation metrics, control parameters particular to BBMA-MDS, and an examination of fitness function parameters. In addition, we present an example of a summary produced by our system and do a comparison evaluation with the reference summaries. The section closes by presenting the final results obtained from our approach, along with a comparative evaluation versus other studies that have utilised the same datasets.

5.1. Programming environment

The programming environment can be divided into two distinct components: the hardware environment and the software environment. Here we present an outline of the utilised environments.

5.1.1. Hardware

The hardware configuration is described as follows:

- ✓ Operating System: Windows 10, 64-bit, x64-based processor.
- ✓ CPU: Intel (R) Core (TM) i5-8350U CPU @ 1.70 GHz 1.90 GHz
- ✓ Memory: 8 GB RAM

Algorithm 1. Binary Biology Migration Algorithm

1. **Input:**
2. Collection of documents to summarizing CD = {s_1, s_2, …..s_D}
3. Number of particles (species) (NP)
4. Maximum number of iterations (T)
5. Switch probability (Pr)
6. Maximum number of cycles (C)
7. **Output:**
8. Sum_{best}: Best summary of the Collection of documents
9. **Begin**
10. Initialize randomly the population P = X_i (i = 1, 2, . . . , NP) in the continuous space
11. **for** each particle *i* in P **do**
12. Calculate the binary vector $XB_i(t)$ of the position $X_i(t)$ using Equations (40) and (41)
13. Feasibility verification
14. Generate the summary $Sum_i(t)$ that contains sentences of *D* that matching the "1" value in the binary vector$X_i(t)$
15. Calculate the fitness value $F_i(t)$ of $Sum_i(t)$ using Equation (32)
16. **end for**
17. S_1= the fittest summary obtained by the particles of the population P
18. t=1
19. **while** (t < T)
20. Calculate the step size L(t) using Equation (34)
21. **for** each particle *i*in the population P
22. (Migration phase)
23. X_{best} = particle that have the best fitness from the population P
24. **if** rand <Pr**then** // rand is a random value
25. Calculate the new position $X_i(t+1)$ using Equation (33)
26. **else**
27. Select two particles *i* and *j* randomly from the population P
28. Calculate the new position $X_i(t+1)$ using Equation (35)
29. **end if**
30. (Updating phase)
30. Calculate the binary vector $XB_i(t+1)$ of the position $X_i(t+1)$ using Equations (40) and (41)
31. Feasibility verification
32. Generate the summary $Sum_i(t+1)$ that contains sentences of *D* that matching the "1" value in the binary vector $X_i(t+1)$
33. Calculate the fitness value $F_i(t+1)$ of $Sum_i(t+1)$ using Equation (32)
34. **if** $F_i(t+1) > F_i(t)$ **then**
35. $X_i(t) = X_i(t+1)$
36. Cycle(i) = 0
37.**else**
38. Cycle(i) = Cycle(i)+1
39. **if** Cycle(i) ≥ C **then**
40. Generate randomly the position $X_i(t+1)$
41. **end if**
42. **end if**
43.**end for**
44. Sum_t = the fittest summary obtained by all the population P in the iteration t
45. $F_{best}(t+1)$ = the fitness of the best particle in the population P at iteration t
46. **if** $F_{best}(t+1) > F_{best}(t)$ **then**
47. $Sum_{best} = Sum_t$
48. t = t+1
49. **endwhile**
50. **Output** the summary Sum_{best}

5.1.2. Software

a) Python

We choose to use the Python programming language, particularly version 3.6, 64 bits, for implementing our approach. We were driven to adopt Python due to its notable features and qualities, including its readability, simplicity, and comprehensive standard library.

Python is a programming language that is classified as high-level, meaning it is designed to be easily understandable and simple to put into practice. It is open source, implying that it is freely available for use, including for commercial purposes. Python is compatible with Mac, Windows, and Unix operating systems, and it has also been adapted to work on Java and NET virtual machines.

Python is extensively utilized in Natural Language Processing (NLP) due to its versatile nature, rich libraries, and user-friendly syntax. Key to its prominence is the availability of powerful NLP libraries like NLTK, SpaCy, TextBlob, and Gensim, offering pre-built functions for diverse NLP tasks.

Built-in functionalities for text processing, web scraping libraries like BeautifulSoup and Scrapy, and dedicated NLP tools like NLTK collectively position Python as a preferred language for researchers and developers in the NLP domain.

b) Visual studio

For editing our code, we employed Visual Studio Code; it's commonly known as VS Code, is a code editor introduced by Microsoft in 2015. Despite its lightweight nature, it is a robust and powerful tool designed to run on desktop environments, making it compatible with Windows, macOS, and Linux operating systems. VS Code provides built-in support for JavaScript, TypeScript, and Node.js, and it boasts a versatile ecosystem of extensions that cater to a variety of languages, including but not limited to C++, C#, Java, Python, PHP, and Go. Additionally, it supports various runtimes such as .NET and Unity, contributing to its widespread adoption among developers for a diverse range of programming tasks.

5.2. Dataset

The DUC (Document Understanding Conference) offers several datasets for text summarization. DUC datasets are centred on single-document and multi-document summarization problems. Each dataset contains a set of source documents in English together with Golden standard summaries created by human experts. To evaluate our approach, we must compare the summaries generated by BBMA-MDS approach with those proposed by the DUC. For the scope of this work, DUC2002 and DUC2004 datasets are used.

These datasets are collections of documents written in English, and each collection has 10 documents. These documents are press articles from two news agencies: the AP and the New York Times. For each collection, four reference summaries are provided.

Table 1 provides a short description of the used DUC datasets which are available in the link: https://www- nlpir.nist.gov/projects/duc/data

Table1: Description of DUC2002 and DUC2004 datasets

Dataset parameters	**DUC2002**	**DUC2004**
Number of collections	60	100
Document per collection	10	10
Data source	duc.nist.gov	TREC

5.3. Evaluation measures

Evaluation metrics are essential for determining the effectiveness of our technique and enabling comparisons with other methodologies. ROUGE is a generally acknowledged evaluation metric in the field of text summarization. In chapter one, we have furnished a comprehensive elucidation of the ROUGE metrics. These indicators are useful for quantifying the performance of our methodology, offering insights into its effectiveness and enabling a meaningful comparison with existing methods in the field of text summarization.

The metrics compare the automatically generated summary with the Golden summary (summary generated by an expert human). A higher ROUGE score indicates that the automatically generated summary is more similar to the reference one. ROUGE-1.5.5 tool

developed in (Lin 2004) includes various ROUGE metrics such as ROUGE-N, ROUGE-L, ROUGE-S and ROUGE-SU.

5.4. Controlling parameters

In this section, we will delve into the examination of algorithm parameters and fitness function parameters.

5.4.1. Algorithm parameters

In optimization algorithms, the values of control parameters vary depending on the application problem. Since there are no universal values that can be applied to all problems, it is necessary to conduct experiments to identify the most appropriate values for each problem.

To identify the most suitable values for the parameters in our problem, including the Maximum Number of Iterations (T), Number of Particles (NP), and Maximum Number of Cycles (C), we conducted several experiments.

Initially, our aim is to determine the best values for two parameters: the number of particles (NP), which represents the population size, and the number of iterations (T). For this experiment, we set the value of the maximum number of cycles (C) to 1 and tested three distinct values (10, 50, and 100) for each parameter (NP and T). For each combination, we calculate the value of the fitness function. The results are shown in Figure 9.

According to Figure 9, the optimal combination of parameters, number of iterations and number of particles, is (50, 100). While there is a little enhancement observed with the combination (100,100), this configuration adversely affects the overall execution time.

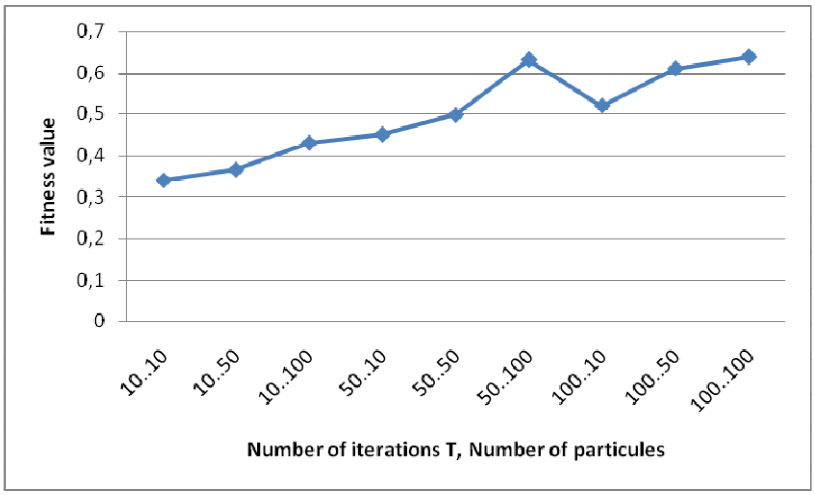

Figure 9: Variation of the fitness value vs. Combination of parameters (number of iterations, number of particles)

In the previous step, we identified the ideal combination of parameters, the number of iterations, and the number of particles, after setting the Maximum Number of Cycles (C) parameter. In this stage, we will keep the previously determined parameter combination of number of iterations and number of particles while testing various values of the Maximum Number of Cycles (C) parameter to determine the ideal value for our problem.

Table 2: Variation of fitness value Vs. Maximum number of Cycles(C)

Maximum Number of Cycles (C)	10	15	20	25
Fitness	0,6432	0,6021	0,5409	0,5018

According to Table 3, the optimal value for the parameter Maximum Number of Cycles (C), which results in the highest fitness value, is 10.

The values of T, NP and C that yield the highest fitness value constitute the most appropriate combination. Table 2 gives the most appropriate values for *T, NP* and C that will be used in the remainder of this work. Table 2 also contains the value of Switch probability (Pr); it is a fixed value in the original algorithm.

Table 3: Algorithm's parameters

Parameter	T	NP	C	Pr
value	50	100	10	0.5

5.4.2. Fitness function parameters

Our fitness function uses the weights α, β and γ respectively for the factors; coverage, cohesion and readability. To determine the most suitable combination of α, β and γ for our problem, several combinations of these weights have been tested. For each combination, we calculated the fitness value and evaluated the generated summary using ROUGE-1 and ROUGE-2. These experiments were performed on the two datasets; DUC2002 and DUC2004. The results are presented in Figures 10 and 11 for the fitness values, along with Tables 4 and 5 for ROUGE-1 and ROUGE-2.

Table 4: ROUGE score on DUC2004 dataset.

α	β	γ	ROUGE-1	ROUGE-2
0.33	0.33	0.33	0.3992	0.1593
0.7	0.2	0.1	0.4175	0.1667
0.8	**0.1**	**0.1**	**0.4379**	**0.1770**
0.6	0.2	0.2	0.3923	0.1353
0.7	0.1	0.2	0.4144	0.1523
0.6	0.1	0.3	0.3564	0.1000
0.6	0.3	0.1	0.3987	0.1429

Table 5: ROUGE score on DUC2002 dataset.

α	β	γ	ROUGE-1	ROUGE-2
0.33	0.33	0.33	0.4501	0.1993
0.7	0.2	0.1	0.4614	0.2291
0.8	**0.1**	**0.1**	**0.4851**	**0.2619**
0.6	0.2	0.2	0.4618	0.2249
0.7	0.1	0.2	0.4501	0.2192
0.6	0.1	0.3	0.4295	0.1867
0.6	0.3	0.1	0.4440	0.1979

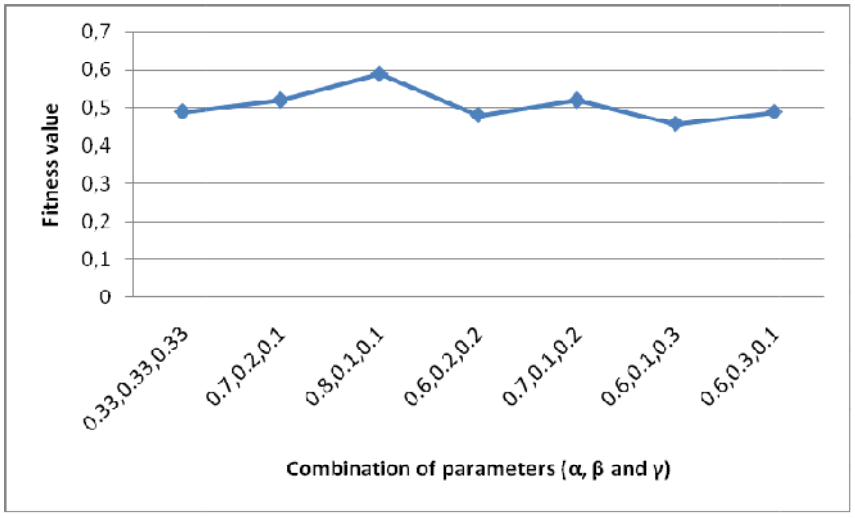

Figure 10: Variation of the fitness value vs. Combination of parameters (α, β and γ) on DUC2004

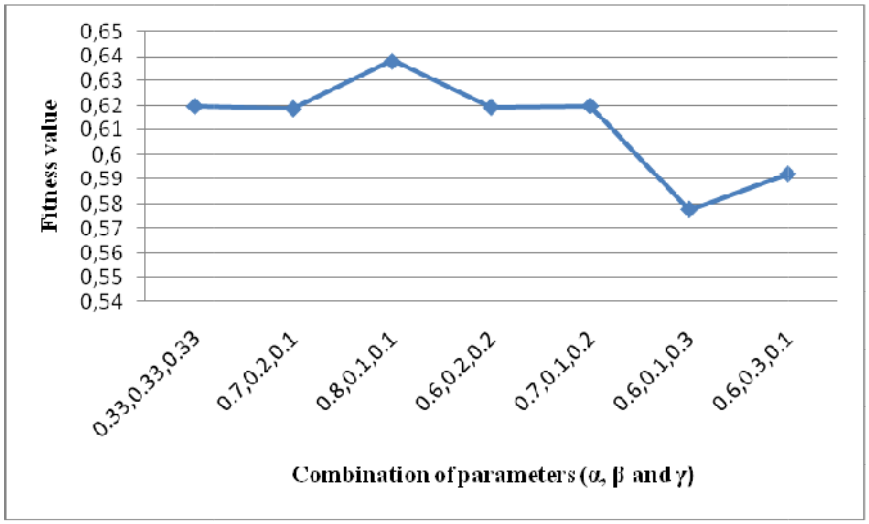

Figure 11: Variation of the fitness value vs. Combination of parameters (α, β and γ) on DUC2002

The fitness function is specifically designed to mirror the quality of the generated summary. As the fitness value increases, it signifies an enhancement in the overall quality of the summary. This observation aligns precisely with the results presented in Figures 9, 10, and Tables 3, 4, where higher fitness values correspond to improved ROUGE scores, indicating more effective and well-constructed summaries.

From the results of the experiments presented in Tables 4 and 5, our approach gives good results in most cases. Furthermore, its best results are obtained when the coefficients α, β and γ are fixed at the values 0.8, 0.1 and 0.1 respectively.

The results shown in Tables 4 and 5 suggest that setting α to 0.8, β to 0.1 and γ to 0.1 improve the ROUGE-1 and ROUGE-2 scores on the DUC2002 and DUC2004 datasets.

The increase in the ROUGE score with the weight of Coverage in the BBMA-MDS approach indicates that the algorithm places more emphasis on including relevant information from the input documents in the summary. As the weight of Coverage increases, the algorithm becomes more focused on ensuring that relevant information and the content coverage of all documents are well-represented in the generated summary. We conclude that the coverage is important to generate a good summary.

6. Results and comparison with other works

6.1. Results

The evaluation results of our approach using ROUGE-1, ROUGE-2 and ROUGE-SU4, with α set to 0.8, β set to 0.1 and γ set to 0.1, are summarized in Tables 6 and 7. The tables show the recall, precision and F-score for ROUGE-1 and ROUGE-2 on DUC2002 and DUC2004 datasets.

Table 6: Recall, Precision and F-score for the BBMA-MDS algorithm on DUC-2002

	Rouge-1	Rouge-2	Rouge -Su-4
Average Recall	0.4842	0.2611	0.2830
Average Precision	0.4862	0.2628	0.2847
Average F-score	0.4851	0.2619	0.2838

Table 7: Recall, Precision and F-score for the BBMA-MDS algorithm on DUC-2004

	Rouge-1	Rouge-2	Rouge -Su-4
Average Recall	0.4282	0.1726	0.1903
Average Precision	0.4489	0.1819	0.2000
Average F-score	0.4379	0.1770	0.1949

6.2. Comparison with other works

6.2.1. Comparison on DUC 2002

- ***Comparison with classical algorithms.***

Our approach, BBMA-MDS, is compared with several classical algorithms such as; LexRank (R Mihalcea and Tarau 2004) where a similarity graph is created with sentences as nodes and edges connecting sentences with a cosine similarity above a certain threshold. The PageRank score of each sentence is then determined within this graph. The most highly ranked sentences are then selected to form the summary; TF-IDF weighting (H. P. Luhn 1958) in this work the score of each sentence is calculated from the TF*IDF of its terms and the highest scoring sentences are chosen until the desired summary length is reached; JS-Greedy; KL-Greedy, and ICSI. These algorithms were implemented and evaluated using the DUC2002 dataset, as described in (Peyrard and Eckle-kohler 2016). The comparison results are shown in Table 8.

Table 8: Performance comparison of BBMA-MDS with classical algorithms on DUC2002.

Methods	ROUGE-1	ROUGE-2
TF-IDF	0.4072	0.1201
Lex Rank	0.4311	0.1388
KL-Greedy	0.3945	0.1125
JS-Greedy	0.4299	0.1455
ICSI	0.4434	0.1556
BBMA-MDS	**0.4851**	**0.2619**

The results show that our approach outperforms other methods in terms of ROUGE-1 and ROUGE-2 metrics. BBMA-MDS has the highest ROUGE-1 score of 0.4851 and the highest ROUGE-2 score of 0.2619.

- ***Comparison with optimization approaches.***

In this part, our approach is compared with FBTS (Tomer and Kumar 2022),FEOM (Song et al. 2011) and many metaheuristic-based approaches implemented on the DUC-2002 dataset in (Tomer and Kumar 2022); Firefly Algorithm with JS-Divergence, PSO using cosine similarity and JS-divergence as fitness functions, ant colony Optimization with cosine similarity, Genetic Algorithm with JS-divergence and cosine similarity.

The results in Table 9 and Figure 12 reveal that our approach, BBMA-MDS, has the highest ROUGE-1 score of 0.4851 and ROUGE-2 score of 0.2619, outperforming other methods. These results demonstrate the effectiveness of our approach in comparison to others.

Table 9: Performance comparison of BBMA-MDS with other metaheuristic based approaches on DUC-2002.

Methods	ROUGE-1	ROUGE-2
FA (JS -Divergence)	0.3262	0.1527
GA (JS-Divergence)	0.4117	0.1758
GA (Cosine Similarity)	0.3597	0.1374
PSO (JS-Divergence)	0.3759	0.1312
PSO (Cosine Similarity)	0.3235	0.1835
FEOM	0.4657	0.1249
ACO (Cosine Similarity)	0.3289	0.1589
FbTS(TRF, CF, RF)	0.4782	0.2295
BBMA-MDS (C, C, R)	**0.4851**	**0.2619**

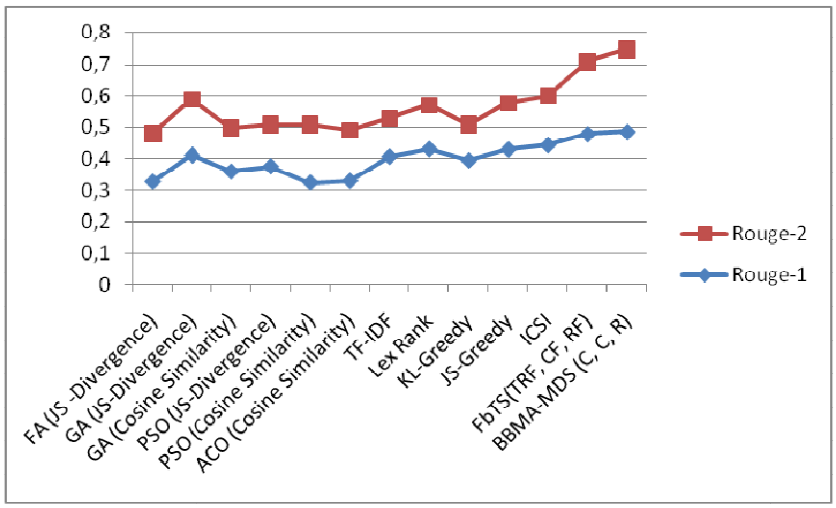

Figure 12: ROUGE-1 and ROUGE-2 comparison of BBMA-MDS with other works on DUC-2002 dataset.

6.2.2. Comparison on DUC 2004

In this part our approach is compared with several meta heuristic-based approaches such as; Genetic algorithm with JS-divergence, ACO with cosine similarity, PSO with JS-divergence (Tomer and Kumar 2022). MCRMR-PSO with cosine similarity and MCRMR-SSO with cosine similarity, MMR-SE System 61 with centroid score and LexRank (Verma and Om 2019). The results of the comparison are presented in Table 10 and Figure 13.

Table 10: Performance comparison of BBMA-MDS with other methods on DUC-2004 dataset.

Methods	ROUGE-1	ROUGE-2
GA (JS-Divergence)	0.3546	0.0937
PSO (JS-Divergence)	0.3521	0.0935
ACO (CosineSimilarity)	0.3542	0.0837
MCRMR_PSO (CosineSimilarity)	0.385	0.139
MCRMR_SSO (CosineSimilarity)	0.410	0.136
MMR_SE	0.336	0.099
System 61(Centroid score)	0.359	0.091
Lex Rank	0.326	0.079
FbTS(TRF, CF, RF)	0.4244	0.1764
BBMA-MDS (C C R)	**0.4343**	**0.1770**

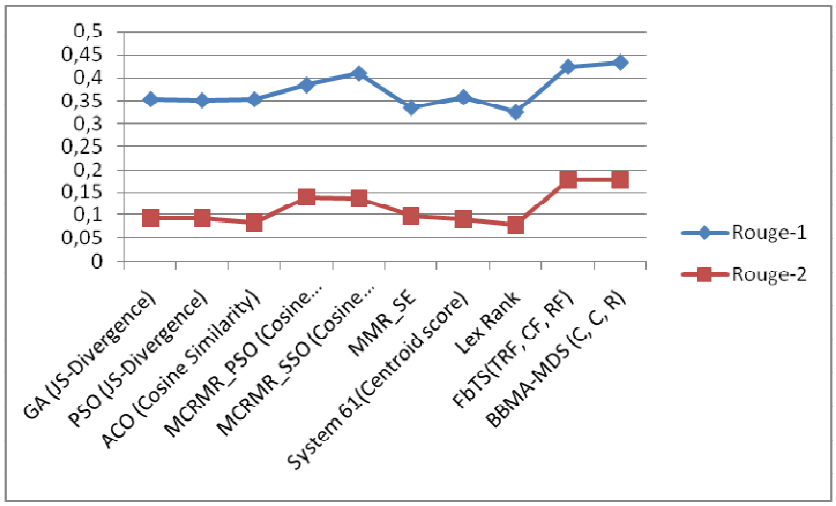

Figure 13: ROUGE-1 and ROUGE-2 comparison of BBMA-MDS with other works on DUC-2004 dataset.

Comparing the ROUGE-1 and ROUGE-2 scores from Table 10, our approach achieves the highest scores of 0.4343 and 0.1770 respectively, outperforming all other methods on DUC2004 dataset.

7. Interpretation and discussion of results

After conducting an analysis of the experimental results, it is clear that BBMA-MDS surpasses all the methods listed in Tables 8, 9, and 10. Notably, our approach outperforms FBTS, which achieved the best result in related works, in both the DUC2002 and DUC2004 datasets when assessed using ROUGE-1 and ROUGE-2 metrics. Specifically, compared with FBTS, BBMA-MDS demonstrate an improvement on DUC2002 of 0.0069 and 0.0324 in terms of ROUGE-1 and ROUGE-2 respectively. For DUC2004, the improvement is of 0.0099 and 0.0006 in ROUGE-1 and ROUGE-2 respectively.

When BBMA-MDS outperforms other methods on the ROUGE-1 and ROUGE-2 metrics, it indicates that BBMA-MDS is more effective at generating summaries that closely match the content of the source documents. Higher ROUGE-1 scores show that BBMA-MDS captures important words and maintains language fluency, while higher ROUGE-2 scores demonstrate its ability to maintain contextual coherence by capturing important word sequences. This superior performance implies that BBMA-MDS produces more informative, comprehensive, and coherent summaries, making it a promising approach for real-world applications.

On the other hand, theses outcomes on ROUGE metrics mean that the system achieves higher recall, precision, and F-scores. This indicates that BBMA-MDS can cover more relevant information from the source documents (higher recall), produce summaries with fewer unnecessary words (higher precision), and achieve a better balance between these two aspects (higher F-score). In summary, the superior recall, precision, and F-score results contribute to the overall effectiveness of BBMA-MDS in generating summaries that closely match the content of the reference summaries, as measured by the ROUGE evaluation.

These impressive outcomes can be attributed to the well-defined fitness function and the BBMA-MDS algorithm's ability to effectively balance between diversification and intensification.

8. Example of summary generated by our approach

In the following, we will provide an example of a summary generated by our system Figure 14, then display the reference summaries, Figures 15, 16, 17, and 18, these summaries are provided with the dataset, which are generated by human experts. We will then try to compare them. Our comparison will be approximate based on what we can observe.

Institute Director Raymundo Punongbayan said **Mount** Pinatubo was in the early eruptive stage and after that we will have bigger ones. Clark Air Base is in Angeles, a city of more than 300,000 people about 50 miles north of Manila. In all, about 9,800 people have been evacuated from communities at the foot of the mountain, said police in Nagasaki, 30 miles away. Scientists doubt it's more than coincidence that the eruption is occurring at the same time as the Mount Unzen volcano in Japan, he said. The last major eruption was the Taal volcano, about 50 miles south of Manila, in 1965. The initial explosions on Mount Pinatubo at 8:51 a.m.

Wednesday sent a 10-mile-high mushroom cloud of swirling ash and rock fragments into the skies over Clark Air Base, forcing the Air Force to evacuate hundreds of American volunteers who had stayed behind to guard it and to tend sensitive communications equipment. At least four people have died, 24 have been hurt and four have been listed as missing since Pinatubo began erupting Sunday, after lying dormant for six centuries, officials said. Friday, eight thunderous explosions shot a cloud of ash and steam nearly 19 miles high.

Figure 14: Summary generated by BBMA-MDS approach.

After slumbering for 600 years, the Mt. Pinatubo volcano erupted in June 1991. Mt. Pinatubo is on the Philippine Island of Luzon. It first erupted on 9 June, and the eruptions continued through the 14th, with as many as six eruptions on one day. The Philippine Institute of Vulcanology and Seismology, and the United States Geological Survey said the volcanic activity was increasing in a way that suggested a more massive eruption would occur. Heavy rain increased the danger of mudslides. The mountain shot out gases, rock and ash. Mushroom clouds ascended to heights of ten miles. The gray-green clouds turned day into night, and covered wide areas with ash. More than 19,000 panicked villagers within a 12-mile radius were evacuated. Clark Air Base, one of the largest U.S. overseas installations, is about ten miles from Pinatubo. Most airmen, dependents and civilian employees were evacuated to the U.S. Naval Base at Subic Bay on the 11th, following the first major eruption. The remaining personnel left a day or two later. Philippine President Aquino dismissed as a fabrication that she had been warned volcanic damage could cause a nuclear incident.

Figure 15: Reference summary 01.

The initial explosions on Mount Pinatubo at 8:51 a.m. Wednesday sent a 10-mile-high mushroom cloud of swirling ash and rock fragments into the skies over Clark Air Base, forcing the Air Force to evacuate hundreds of American volunteers who had stayed behind to guard it and to tend sensitive communications equipment. Philippine civil defense officials said that they have evacuated 19,369 people from villages within a 12-mile radius of the volcano by late Wednesday. The volcano, about 10 miles west of the air base, came to life last week after six centuries of dormancy. Three major new eruptions rocked Mount Pinatubo late Wednesday night and early today, forcing another emergency evacuation of Clark Air Base and increasing fears of even more violent explosions from the long-dormant volcano in the days ahead. The U.S. military began flying home hundreds of the 28,000 Americans who crowded onto Subic Bay Naval Base as a bizarre tropical blizzard of thick volcanic ash caused power failures across the base Friday during a third day of increasingly violent eruptions of Mount Pinatubo. At least four people have died, 24 have been hurt and four have been listed as missing since Pinatubo began erupting Sunday, after lying dormant for six centuries, officials said.

Figure 16: Reference summary 02.

Since roaring to life about two weeks ago, explosions at Mount Pinatubo, located 10 miles from Clark AFB in the Philippines, have become increasingly severe. The first eruptions shot gas, steam and ash 12,000 feet above the mountain. Today's eruptions spewed matter over 12 miles high, and a possible lava flow was seen for the first time. Clark, evacuated nearly a week ago, is covered in ash, as is Subic Bay naval base, which received the 16,000 Clark evacuees. U.S. officials, while not commenting on whether nuclear weapons are at the abandoned base, denied the possibility of radioactive contamination if molten rock hits the weapons depots. At least four deaths and 24 people injured have been recorded since Pinatubo began erupting after six centuries of dormancy. About 84,000 have been evacuated. Heavy rains today have raised fears of catastrophic mudflows. Pinatubo is in the "Ring of Fire" that circles the Pacific. Mount Unzen in Japan, also in the ring, recently erupted, killing 38. President Aquino toured refugee centers and authorized the release of $1.42 million for relief efforts. According to Foreign Secretary Manglapus, the eruptions should not effect the stalled talks with the U.S. on new Clark and Subic lease agreements.

Figure 17: Reference summary 03.

Thousands of Americans piled into cars and buses with dogs, cats and duffel bags and left Clark Air Base today after scientists warned that the eruption of a nearby volcano could turn catastrophic. The decision to evacuate one of the largest overseas U.S. military bases was made early today after searing gases, ash and rock shot out of 4,795-foot Mount Pinatubo on Sunday at speeds of up to 60 mph . Angeles, a city of 300,000, is adjacent to Clark Air Base, but authorities had not ordered a general evacuation there. President Corazon Aquino helicoptered to refugee centers near Angeles briefly before returning to Manila for an Independence Day rally and parade at Luneta Park. She authorized the release of $1.42 million for relief efforts. U.S. officials have refused to say whether nuclear weapons are stored at Clark, but they denied news reports suggesting a radiation danger if weapons depots on the base were hit by molten rock. Four people have died, 24 have been hurt and four have been listed as missing since Pinatubo began erupting Sunday, after lying dormant for six centuries. About 84,000 people, including Americans, have been evacuated.

Figure 18: Reference summary 04.

Upon conducting a concise comparison, we observed that our summary, along with the other four summaries, related to the Pinatubo volcano. The words highlighted with the same colour are found in both our summary and the reference summaries. The segments of texts coloured with the same colour in the summary generated by our system and the other summaries have comparable meanings, although they may have been rephrased by human experts.

In conclusion, the summary produced by our algorithm closely aligns with the other four summaries, but with subtle distinctions, as human experts employ cognitive processes and possess distinct writing styles. In contrast, our system employs an extractive approach, whereby the produced summary consists of existing sentences from the original text.

9. Conclusion

This chapter introduced the innovative proposal of BBMA-MDS, an approach dedicated to automatic multi-text document summarization, based on the BMA swarm intelligence algorithm. The importance of swarm intelligence algorithms in solving combinatorial optimization problems was highlighted, with a specific application to the complex MDS problem. By considering MDS as an optimization problem, we have established a solid theoretical basis for our approach. The second part of the chapter, devoted to the experimental results, played a crucial role in the calibration of the parameters of the algorithm and the coefficients of the objective function. The results obtained clearly demonstrated the superior effectiveness of our approach compared to other existing methods, thus paving the way for future improvements. These findings reinforce our confidence in the relevance of BBMA-

MDS as an innovative and efficient solution for automatic multi-document summarization, justifying the envisaged avenues for future development and enhancement of this approach

General conclusion and perspectives

In conclusion, we will provide a summary of the work accomplished as part of this thesis project. We will provide a brief overview of the contributions made to the field of multi-document text summarization, as well as a discussion of the outcomes acquired. The proposed approach is then used to provide perspectives for further development.

In this thesis, we developed an approach for automatic text summarization. More precisely, we provide a novel extraction approach for multi-document text summarization known as the Binary Biology Migration Algorithm for Multi-document Text Summarization, or BBMA-MDS, which is based on the swarm intelligence algorithm BMA. To achieve our purpose, we initially set out to model the multi-document summarization problem as a combinatorial optimization problem. The second step involved the design and implementation of our proposed approach.

Our important contributions are emphasised below:

For the first time, we use the Biology Migration Algorithm (BMA) to solve MDS problems. This application of BMA increased the quality of the resulting summary and demonstrated the efficacy of swarm intelligence approaches in addressing the issues of multi-document summarization.

In comparison to other approaches, our approach provides a novel enhancement for the binarization step. We employ a sigmoid function to improve the discretization of the continuous aspect of the original BMA. This function converts real numbers ranging from 0 to 1 into discrete values and assigns them to either 0 or 1.

We present a new objective function designed to assess the quality of the produced summaries. This function includes three key features: coverage, cohesion, and readability. Significantly, the accompanying parameters or weights are not chosen at random; rather, they are carefully selected to produce relevant and useful summary results. This thorough parameter attribution improves the precision and effectiveness of our approach.

In our experiments, we used two datasets: DUC2002 and DUC2004. To compare the results of our approach with several baseline algorithms such as TF-IDF, LexRenk, and several

metaheuristic-based approaches such as FBTS and PSO, we employed the ROUGE metrics, which are widely used in the field of automatic text summarization.

The comparison of BBMA-MDS to other approaches employing ROUGE-1 and ROUGE-2 metrics demonstrates that it is more effective in producing summaries that are closely aligned with the source document content. High scores show ability in catching key sentences while conserving contextual coherence. The results show higher recall, precision, and F-scores, demonstrating BBMA-MDS's capacity to provide significant information while reducing redundant words and striking a good balance. These accomplishments originate from a well-defined fitness function and the algorithm's balance between diversification and intensification, establishing BBMA-MDS as a potential solution for thorough and coherent summarization in a variety of applications.

Although we have achieved significant advancements, our work is inherently constrained by certain limits. The evaluation of the BBMA-MDS approach has been limited to the DUC2002 and DUC2004 datasets, which presents doubts about its applicability to additional datasets. In addition, the evaluation of our approach has mostly depended on the ROUGE metrics. This indicates the necessity for an additional set of evaluation measures to guarantee a comprehensive and thorough assessment of its effectiveness across different aspects.

In terms of future directions, we aim to delve deeper into the optimization of the BBMA-MDS algorithm by employing a learning method to fine-tune the weights (α, β, and γ) within the fitness function. This approach seeks to optimize the algorithm's performance in both diversification and intensification aspects. Additionally, we plan to explore enhancements and extensions to the BBMA-MDS algorithm, possibly integrating other metaheuristics or combining diverse bio-inspired algorithms to further refine its capabilities. To rigorously evaluate and generalize the solution, we aspire to create specialized datasets across varied contexts. Moreover, our research will extend into adapting the approach to automatically generate comprehensive and up-to-date scientific state-of-the-art summaries, providing valuable insights for researchers in specific domains. These diverse perspective and propositions aim to advance the effectiveness and applicability of our approach in addressing complex multi-document text summarization challenges.

Publications

Boussalem, Mohamed, Samia Aitouche, Moumen Hamouma, Hichem Haouassi, Hichem Rahab, and Abdelaali Bekhouche. 2023. “BBMA-MDS: Binary Biology Migration Algorithm for Multi-Document Text Summarization.” *Revue d’Intelligence Artificielle* 37 (5).

Bibliography

"" Http://Www.Metaheuristics.Net." n.d.

Abo-Elghit, Amira Hamed, Aya Mohammed Al-Zoghby, and Taher Tawfeek Hamza. 2020. "Textual Similarity Measurement Approaches: A Survey (1)." *The Egyptian Journal of Language Engineering* 7 (2): 41–62.

Agrawal, P, H F Abutarboush, T Ganesh, and A W Mohamed. 2021. "Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019)." *IEEE Access* 9: 26766–91. https://doi.org/10.1109/ACCESS.2021.3056407.

Alguliev, Rasim M, Ramiz M Aliguliyev, Makrufa S Hajirahimova, and Chingiz A Mehdiyev. 2011. "MCMR: Maximum Coverage and Minimum Redundant Text Summarization Model." *Expert Systems with Applications* 38 (12): 14514–22. https://doi.org/https://doi.org/10.1016/j.eswa.2011.05.033.

Alzuhair, Abeer, and Mohammed Al-Dhelaan. 2019. "An Approach for Combining Multiple Weighting Schemes and Ranking Methods in Graph-Based Multi-Document Summarization." *IEEE Access* 7: 120375–86.

Appel, Orestes, Francisco Chiclana, Jenny Carter, and Hamido Fujita. 2016. "A Hybrid Approach to the Sentiment Analysis Problem at the Sentence Level." *Knowledge-Based Systems* 108: 110–24.

Babar, S A, and Pallavi D Patil. 2015. "Improving Performance of Text Summarization." *Procedia Computer Science* 46: 354–63. https://doi.org/https://doi.org/10.1016/j.procs.2015.02.031.

Back, Thomas. 1993. "Evolutionary Programming and Evolution Strategies: Similarities and Differences." In *Proc. of the Second Ann. Conf. on Evolutionary Programming*, 11–22.

Barros, Cristina, Elena Lloret, Estela Saquete, and Borja Navarro-Colorado. 2019. "NATSUM: Narrative Abstractive Summarization through Cross-Document Timeline Generation." *Information Processing & Management* 56 (5): 1775–93.

Baxendale, Phyllis B. 1958. "Machine-Made Index for Technical Literature—an Experiment." *IBM Journal of Research and Development* 2 (4): 354–61.

Bernier., H. Borko and C. L. 1975. “Abstracting Concepts and Methods.” *Academic Press, London.*

Bhaskar, Pinaki, and Sivaji Bandyopadhyay. 2010. “A Query Focused Multi Document Automatic Summarization.” In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, 545–54.

Bhat, Iram Khurshid, Mudasir Mohd, and Rana Hashmy. 2018. “Sumitup: A Hybrid Single-Document Text Summarizer.” In *Soft Computing: Theories and Applications: Proceedings of SoCTA 2016, Volume 1*, 619–34. Springer.

Binwahlan, Mohammed Salem, Naomie Salim, and Ladda Suanmali. 2010. “Fuzzy Swarm Diversity Hybrid Model for Text Summarization.” *Information Processing & Management* 46 (5): 571–88.

Bonabeau, Eric, Marco Dorigo, and Guy Theraulaz. 1999. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford university press.

Boudin, Florian, and Juan-Manuel Torres-Moreno. 2009. “Résumé Automatique Multi-Document et Indépendance de La Langue: Une Première Évaluation En Français.” In *Actes de La 16ème Conférence Sur Le Traitement Automatique Des Langues Naturelles. Articles Courts*, 321–30.

Boussalem, Mohamed, Samia Aitouche, Moumen Hamouma, Hichem Haouassi, Hichem Rahab, and Abdelaali Bekhouche. 2023. “BBMA-MDS: Binary Biology Migration Algorithm for Multi-Document Text Summarization.” *Revue d’Intelligence Artificielle* 37 (5).

Brants, Thorsten. 2003. “Natural Language Processing in Information Retrieval.” *CLIN* 111.

Chakraborty, Amrita, and Arpan Kumar Kar. 2017. “Swarm Intelligence: A Review of Algorithms.” In *Nature-Inspired Computing and Optimization*, edited by Srikanta Patnaik, Xin-She Yang, and Kazumi Nakamatsu, 475–94. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-50920-4_19.

Chali, Yllias, and Shafiq Joty. 2008. “Improving the Performance of the Random Walk Model for Answering Complex Questions.” In *Proceedings of ACL-08: HLT, Short Papers*, 9–12.

Conroy, John M, Judith D Schlesinger, Jeff Kubina, Peter A Rankel, and Dianne P O'Leary. 2011. "CLASSY 2011 at TAC: Guided and Multi-Lingual Summaries and Evaluation Metrics." *Tac* 11: 1–8.

Darwin, Charles. 1876. *On the Origin of Species by Means of Natural Selection; or, the Preservation of Favored Races in the Struggle for Life*. John murray.

Deneubourg, J -L, Serge Aron, Simon Goss, and Jacques M Pasteels. 1990. "The Self-Organizing Exploratory Pattern of the Argentine Ant." *Journal of Insect Behavior* 3: 159–68.

Deza, Elena, Michel Marie Deza, Michel Marie Deza, and Elena Deza. 2009. *Encyclopedia of Distances*. Springer.

Donis-Díaz, Carlos A, Rafael Bello, and Janusz Kacprzyk. 2015. "Using Ant Colony Optimization and Genetic Algorithms for the Linguistic Summarization of Creep Data." In *Intelligent Systems' 2014: Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014, September 24-26, 2014, Warsaw, Poland, Volume 1: Mathematical Foundations, Theory, Analyses*, 81–92. Springer.

Dorigo, Marco. 1991. "Positive Feedback as a Search Strategy." *Technical Report*, 16–91.

———. 1992. "Optimization, Learning and Natural Algorithms." *Ph. D. Thesis, Politecnico Di Milano*.

Dorigo, Marco, and Christian Blum. 2005. "Ant Colony Optimization Theory: A Survey." *Theoretical Computer Science* 344 (2–3): 243–78.

Dorigo, Marco, Vittorio Maniezzo, and Alberto Colorni. 1996. "Ant System: Optimization by a Colony of Cooperating Agents." *IEEE Transactions on Systems, Man, and Cybernetics, Part b (Cybernetics)* 26 (1): 29–41.

Dorigo, Marco, and Thomas Stützle. 2019. *Ant Colony Optimization: Overview and Recent Advances*. Springer.

Eberhart, Russ, Pat Simpson, and Roy Dobbins. 1996. *Computational Intelligence PC Tools*. Academic Press Professional, Inc.

Edmundson, Harold P. 1969. "New Methods in Automatic Extracting." *Journal of the ACM (JACM)* 16 (2): 264–85.

Eiben, Agoston E, and James E Smith. 2015. *Introduction to Evolutionary Computing*. Springer.

Erkan, Günes, and Dragomir R Radev. 2004. "Lexrank: Graph-Based Lexical Centrality as Salience in Text Summarization." *Journal of Artificial Intelligence Research* 22: 457–79.

Fattah, Mohamed Abdel, and Fuji Ren. 2009. "GA, MR, FFNN, PNN and GMM Based Models for Automatic Text Summarization." *Computer Speech & Language* 23 (1): 126–44.

Favre, Benoit, Evgeny Stepanov, Jérémy Trione, Frédéric Béchet, and Giuseppe Riccardi. 2015. "Call Centre Conversation Summarization: A Pilot Task at Multiling 2015." In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 232–36.

Filippova, Katja. 2009. "Dependency Graph Based Sentence Fusion and Compression." Technische Universität.

Fogel, David B. 1991. *System Identification through Simulated Evolution: A Machine Learning Approach to Modeling*. Ginn Press.

———. 1998. *Artificial Intelligence through Simulated Evolution*. Wiley-IEEE Press.

Gambhir, Mahak, and Vishal Gupta. 2017. "Recent Automatic Text Summarization Techniques: A Survey." *Artificial Intelligence Review* 47: 1–66.

Gao, Shen, Xiuying Chen, Piji Li, Zhaochun Ren, Lidong Bing, Dongyan Zhao, and Rui Yan. 2019. "Abstractive Text Summarization by Incorporating Reader Comments." In *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:6399–6406.

Genest, Pierre-Etienne, and Guy Lapalme. 2011. "Framework for Abstractive Summarization Using Text-to-Text Generation." In *Proceedings of the Workshop on Monolingual Text-to-Text Generation*, 64–73.

Giannakopoulos, George. 2013. "Multi-Document Multilingual Summarization and Evaluation Tracks in Acl 2013 Multiling Workshop." In *Proceedings of the Multiling 2013 Workshop on Multilingual Multi-Document Summarization*, 20–28.

Giannakopoulos, George, Mahmoud El-Haj, Benoit Favre, Marina Litvak, Josef Steinberger, and Vasudeva Varma. 2011. "TAC 2011 MultiLing Pilot Overview."

Gillick, Dan, and Benoit Favre. 2009. “A Scalable Global Model for Summarization.” In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, 10–18.

Glover, Fred. 1986. “Future Paths for Integer Programming and Links to Artificial Intelligence.” *Computers & Operations Research* 13 (5): 533–49.

Goldberg, David E. 1989. “Cenetic Algorithms in Search.” *Optimization, Machine Learning*.

Goldstein, Jade, Vibhu O Mittal, Jaime G Carbonell, and Mark Kantrowitz. 2000. “Multi-Document Summarization by Sentence Extraction.” In *NAACL-ANLP 2000 Workshop: Automatic Summarization*.

H. P. Luhn. 1958. “The Automatic Creation of Literature Abstracts.” *IBM Journal of Research and Development* 2 (April): 159–65. https://doi.org/10.1147/rd.22.0159.

Hassel, Martin. 2003. “Exploitation of Named Entities in Automatic Text Summarization for Swedish.” In *NODALIDA’03–14th Nordic Conferenceon Computational Linguistics, Reykjavik, Iceland, May 30–31 2003*, 9.

Hichem, Haouassi, Merah Elkamel, Mehdaoui Rafik, Maarouk Toufik Mesaaoud, and Chouhal Ouahiba. 2022. “A New Binary Grasshopper Optimization Algorithm for Feature Selection Problem.” *Journal of King Saud University - Computer and Information Sciences* 34 (2): 316–28. https://doi.org/https://doi.org/10.1016/j.jksuci.2019.11.007.

HLTCOE, J H U. n.d. “HLTCOE Submission at TREC 2013: Temporal Summarization.”

Holland, JohnH. 1975. “Adaptation in Natural and Artificial Systems, Univ. of Mich. Press.” *Ann Arbor* 7: 390–401.

Hong, Kai, Mitch Marcus, and Ani Nenkova. 2015. “System Combination for Multi-Document Summarization.” In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 107–17.

Hovy, Eduard H, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. 2006. “Automated Summarization Evaluation with Basic Elements.” In *LREC*, 6:604–11.

Hovy, Eduard, and Chin-Yew Lin. 1998. “Automated Text Summarization and the SUMMARIST System.” In *TIPSTER TEXT PROGRAM PHASE III: Proceedings of a*

Workshop Held at Baltimore, Maryland, October 13-15, 1998, 197–214.

Hynek, Jiri, and Karel Jezek. 2003. "Practical Approach to Automatic Text Summarization." In *ELPUB*. Citeseer.

Ishikawa, Kai. 2001. "A Hybrid Text Summarization Method Based on the TF Method and the Lead Method." In *Proceedings of the Second NTCIR Workshop Meeting on Evaluation of Chinese & Japanese Text Retrieval and Text Summarization*, 325–30.

Jaccard, Paul. 1912. "The Distribution of the Flora in the Alpine Zone. 1." *New Phytologist* 11 (2): 37–50.

Jones, Karen Sparck, and Julia R Galliers. 1995. "Evaluating Natural Language Processing Systems: An Analysis and Review."

Kabadjov, Mijail, Josef Steinberger, Emma Barker, Udo Kruschwitz, and Massimo Poesio. 2015. "Onforums: The Shared Task on Online Forum Summarisation at Multiling'15." In *Proceedings of the 7th Annual Meeting of the Forum for Information Retrieval Evaluation*, 21–26.

Kan, Min-Yen, Kathleen R McKeown, and Judith L Klavans. 2001. "Applying Natural Language Generation to Indicative Summarization." *ArXiv Preprint Cs/0107019*.

Karen, SPARCK-JONES. 1999. "Automatic Summarizing: Factors and Directions," in" Advances in Automatic Text Summarization." *Evaluations*, 6–7.

Kazantseva, Anna. 2006. "An Approach to Summarizing Short Stories." In *Student Research Workshop*, 55–62.

Kennedy, James, and Russell Eberhart. 1995. "Particle Swarm Optimization." In *Proceedings of ICNN'95-International Conference on Neural Networks*, 4:1942–48. IEEE.

Kennedy, James, and Rui Mendes. 2002. "Population Structure and Particle Swarm Performance." In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, 2:1671–76. IEEE.

Khan, Atif, and Naomie Salim. 2014. "A Review on Abstractive Summarization Methods." *Journal of Theoretical and Applied Information Technology* 59 (1): 64–72.

Khan, Atif, Naomie Salim, and Yogan Jaya Kumar. 2015. "A Framework for Multi-Document Abstractive Summarization Based on Semantic Role Labelling." *Applied Soft*

Computing 30: 737–47. https://doi.org/https://doi.org/10.1016/j.asoc.2015.01.070.

Koza John, R. 1992. “Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems).” A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England.

Krause, Jonas, Jelson Cordeiro, Rafael Parpinelli, and Heitor Lopes. 2013. “A Survey of Swarm Algorithms Applied to Discrete Optimization Problems.” In *Swarm Intelligence and Bio-Inspired Computation*, 169–91. https://doi.org/10.1016/B978-0-12-405163-8.00007-7.

Kubina, Jeff, John Conroy, and Judith D Schlesinger. 2013. “Acl 2013 Multiling Pilot Overview.” In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-Document Summarization*, 29–38.

Kumar, A Anil, and S Chandrasekhar. 2012. “Text Data Pre-Processing and Dimensionality Reduction Techniques for Document Clustering.” *International Journal of Engineering Research & Technology (IJERT)* 1 (5): 1–6.

L. J. Fogel, A. J. Owens, & M. J. Walsh. 1966. “Artificial Intelligence through Simulated Evolution.”

Lawler, Eugène L, Jan Karel Lenstra, and A H G Rinnooy Kan. 1985. “DB, Shmoys,\The Traveling Salesman Problem.” *A Guided Tour of Combinatorial Optimization," Wiley*.

Li, Chen, Xian Qian, and Yang Liu. 2013. “Using Supervised Bigram-Based ILP for Extractive Summarization.” In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1004–13.

Li, Wei, and Hai Zhuge. 2019. “Abstractive Multi-Document Summarization Based on Semantic Link Network.” *IEEE Transactions on Knowledge and Data Engineering* 33 (1): 43–54.

Lin, Chin-Yew. 2004. “ROUGE: A Package for Automatic Evaluation of Summaries.” *In: Text Summarization Branches*, 74–81.

Lin, Chin-Yew, and Eduard Hovy. 1997. “Identifying Topics by Position.” In *Fifth Conference on Applied Natural Language Processing*, 283–90.

———. 2003. “Automatic Evaluation of Summaries Using N-Gram Co-Occurrence

Statistics." In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 150–57.

Ling, Jeffrey. 2017. "Coarse-to-Fine Attention Models for Document Summarization."

Madnani, Nitin, and Bonnie J Dorr. 2010. "Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods." *Computational Linguistics* 36 (3): 341–87.

Mani, Inderjeet, David House, Gary Klein, Lynette Hirschman, Therese Firmin, and Beth M Sundheim. 1999. "The TIPSTER SUMMAC Text Summarization Evaluation." In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, 77–85.

Mani, Inderjeet, Gary Klein, David House, Lynette Hirschman, Therese Firmin, and Beth Sundheim. 2002. "SUMMAC: A Text Summarization Evaluation." *Natural Language Engineering* 8 (1): 43–68.

Marcus, Mitchell, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. "Building a Large Annotated Corpus of English: The Penn Treebank."

McCreadie, Richard, Craig Macdonald, and Iadh Ounis. 2014. "Incremental Update Summarization: Adaptive Sentence Selection Based on Prevalence and Novelty." In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 301–10.

McDonald, Ryan. 2007. "A Study of Global Inference Algorithms in Multi-Document Summarization." In *European Conference on Information Retrieval*, 557–64. Springer.

McKeown, Kathleen, and Dragomir R Radev. 1995. "Generating Summaries of Multiple News Articles." In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 74–82.

Mihalcea, R, and P Tarau. 2004. "Textrank: Bringing Order into Texts. Association for Computational Linguistics." *EECS News*.

Mihalcea, Rada. 2004. "Graph-Based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization." In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, 170–73.

Mihalcea, Rada, and Paul Tarau. 2004. "Textrank: Bringing Order into Text." In *Proceedings*

of the 2004 Conference on Empirical Methods in Natural Language Processing, 404–11.

———. 2005. “A Language Independent Algorithm for Single and Multiple Document Summarization.” In *Companion Volume to the Proceedings of Conference Including Posters/Demos and Tutorial Abstracts*.

Mitkov, Ruslan, and Machine Translation Unit. 1993. “Automatic Abstracting in a Limited Domain.” *Proceedings of PACFoCoL I*.

Morris, Andrew H, George M Kasper, and Dennis A Adams. 1992. “The Effects and Limitations of Automated Text Condensing on Reading Comprehension Performance.” *Information Systems Research* 3 (1): 17–35.

Nallapati, Ramesh, Bowen Zhou, Caglar Gulcehre, and Bing Xiang. 2016. “Abstractive Text Summarization Using Sequence-to-Sequence Rnns and Beyond.” *ArXiv Preprint ArXiv:1602.06023*.

Narendra, and Fukunaga. 1977. “A Branch and Bound Algorithm for Feature Subset Selection.” *IEEE Transactions on Computers* 100 (9): 917–22.

Nenkova, Ani, and Kathleen McKeown. 2011. “Automatic Summarization.” *Foundations and Trends® in Information Retrieval* 5 (2–3): 103–233.

Nenkova, Ani, and Rebecca J Passonneau. 2004. “Evaluating Content Selection in Summarization: The Pyramid Method.” In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: Hlt-Naacl 2004*, 145–52.

Nenkova, Ani, Rebecca Passonneau, and Kathleen McKeown. 2007. “The Pyramid Method: Incorporating Human Content Selection Variation in Summarization Evaluation.” *ACM Transactions on Speech and Language Processing (TSLP)* 4 (2): 4-es.

Neto, Joel Larocca, Alex A Freitas, and Celso A A Kaestner. 2002. “Automatic Text Summarization Using a Machine Learning Approach.” In *Advances in Artificial Intelligence: 16th Brazilian Symposium on Artificial Intelligence, SBIA 2002 Porto de Galinhas/Recife, Brazil, November 11–14, 2002 Proceedings 16*, 205–15. Springer.

Nobata, Chikashi, and Satoshi Sekine. 2004. “CRL/NYU Summarization System at DUC-2004.” In *Proceedings of DUC*.

Nunberg, Geoffrey. 1990. *The Linguistics of Punctuation*. Center for the Study of Language (CSLI).

Paice, Chris D. 1980. "The Automatic Generation of Literature Abstracts: An Approach Based on the Identification of Self-Indicating Phrases." In *Proceedings of the 3rd Annual ACM Conference on Research and Development in Information Retrieval*, 172–91.

Parpinelli, Rafael S, and Heitor S Lopes. 2011. "New Inspirations in Swarm Intelligence: A Survey." *International Journal of Bio-Inspired Computation* 3 (1): 1–16.

Peyrard, Maxime, and Judith Eckle-kohler. 2016. "A General Optimization Framework for Multi-Document Summarization Using Genetic Algorithms and Swarm Intelligence," 247–57.

Porter, Martin F. 1980. "An Algorithm for Suffix Stripping." *Program* 14 (3): 130–37.

———. 2001. "Snowball: A Language for Stemming Algorithms, 2001."

Radev, Dragomir, Simone Teufel, Horacio Saggion, Wai Lam, John Blitzer, Hong Qi, Arda Celebi, Danyu Liu, and Elliott Franco Drabek. 2003. "Evaluation Challenges in Large-Scale Document Summarization." In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 375–82.

Rahab, Hichem, Hichem Houassi, and Abdelkader Laouid. 2022. "Rule-Based Arabic Sentiment Analysis Using Binary Equilibrium Optimization Algorithm." *Arabian Journal for Science and Engineering*, September. https://doi.org/10.1007/s13369-022-07198-2.

Rautray, Rasmita, and Rakesh Chandra Balabantaray. 2017. "Cat Swarm Optimization Based Evolutionary Framework for Multi Document Summarization." *Physica A: Statistical Mechanics and Its Applications* 477 (July): 174–86. https://doi.org/10.1016/j.physa.2017.02.056.

———. 2018. "An Evolutionary Framework for Multi Document Summarization Using Cuckoo Search Approach: MDSCSA." *Applied Computing and Informatics* 14 (2): 134–44. https://doi.org/10.1016/j.aci.2017.05.003.

Rechenberg, Ingo. 1965. "Cybernetic Solution Path of an Experimental Problem." *Roy. Aircr. Establ., Libr. Transl.* 1122.

Ren, Pengjie, Furu Wei, Zhumin Chen, Jun Ma, and Ming Zhou. 2016. "A Redundancy-Aware Sentence Regression Framework for Extractive Summarization." In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 33–43.

Republic, Czech. 2009. "EVALUATION MEASURES FOR TEXT SUMMARIZATION Josef Steinberger , Karel Je˘ Zek" 28: 1001–25.

Reynar, Jeffrey C, and Adwait Ratnaparkhi. 1997. "A Maximum Entropy Approach to Identifying Sentence Boundaries." *ArXiv Preprint Cmp-Lg/9704002*.

Reynolds, Craig W. 1987. "Flocks, Herds and Schools: A Distributed Behavioral Model." In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, 25–34.

Riley, Michael. 1989. "Some Applications of Tree-Based Modelling to Speech and Language." In *Speech and Natural Language: Proceedings of a Workshop Held at Cape Cod, Massachusetts, October 15-18, 1989*.

S. Voß, S. Martello, I.H. Osman and C. Roucairol. 1999. "'MetaHeuristics - Advances and Trends in Local Search Paradigms for Optimization'." *Kluwer Academic Publishers*.

Salton, Gerard, and Christopher Buckley. 1988. "Term-Weighting Approaches in Automatic Text Retrieval." *Information Processing & Management* 24 (5): 513–23.

Salton, Gerard, Amit Singhal, Chris Buckley, and Mandar Mitra. 1996. "Automatic Text Decomposition Using Text Segments and Text Themes." In *Proceedings of the the Seventh ACM Conference on Hypertext*, 53–65.

Salton, Gerard, and Chung-Shu Yang. 1973. "On the Specification of Term Values in Automatic Indexing." *Journal of Documentation* 29 (4): 351–72.

Schiffman, Barry, Ani Nenkova, and Kathleen McKeown. 2002. "Experiments in Multidocument Summarization."

Schwefel, Hans-Paul. 1981. *Numerical Optimization of Computer Models*. John Wiley & Sons, Inc.

Selvan, R. Senthamizh, and K. Arutchelvan. 2021a. "An Efficient Social Spider Optimization Algorithm Based Multi-Document Summarization Model." *Proceedings of the 6th*

International Conference on Inventive Computation Technologies, ICICT 2021, 855–60. https://doi.org/10.1109/ICICT50816.2021.9358521.

———. 2021b. "Improved Cuckoo Search Optimization Algorithm Based Multi-Document Summarization Model." *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, no. Iccmc: 735–39. https://doi.org/10.1109/ICCMC51019.2021.9418473.

Shen, Chao, and Tao Li. 2010. "Multi-Document Summarization via the Minimum Dominating Set." In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, 984–92.

Shi, Lei, Furu Wei, Shixia Liu, Li Tan, Xiaoxiao Lian, and Michelle X Zhou. 2010. "Understanding Text Corpora with Multiple Facets." In *2010 IEEE Symposium on Visual Analytics Science and Technology*, 99–106. IEEE.

Song, Wei, Lim Cheon Choi, Soon Cheol Park, and Xiao Feng Ding. 2011. "Fuzzy Evolutionary Optimization Modeling and Its Applications to Unsupervised Categorization and Extractive Summarization." *Expert Systems with Applications* 38 (8): 9112–21.

Steinberger, Josef, and Karel Jezek. 2009. "Evaluation Measures for Text Summarization." *Computing and Informatics* 28 (January): 251–75.

Sun, Shiliang, Chen Luo, and Junyu Chen. 2017. "A Review of Natural Language Processing Techniques for Opinion Mining Systems." *Information Fusion* 36: 10–25.

Tan, Min-Yen Kan Chew-Lim. 2011. "Swing: Exploiting Category-Specific Information for Guided Summarization." *TAC*.

Thomas, Justine Raju, Santosh Kumar Bharti, and Korra Sathya Babu. 2016. "Automatic Keyword Extraction for Text Summarization in E-Newspapers." In *Proceedings of the International Conference on Informatics and Analytics*, 1–8.

Tomer, Minakshi, and Manoj Kumar. 2022. "Multi-Document Extractive Text Summarization Based on Firefly Algorithm." *Journal of King Saud University - Computer and Information Sciences* 34 (8): 6057–65. https://doi.org/10.1016/j.jksuci.2021.04.004.

Trandabat, Diana. 2011. "Using Semantic Roles to Improve Summaries." In *Proceedings of*

the 13th European Workshop on Natural Language Generation, 164–69.

Tsai, Chun-Wei, and Ming-Chao Chiang. 2023. “Chapter Two - Optimization Problems.” In *Uncertainty, Computational Techniques, and Decision Intelligence*, edited by Chun-Wei Tsai and Ming-Chao B T - Handbook of Metaheuristic Algorithms Chiang, 13–27. Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-44-319108-4.00015-0.

Valdez, Fevrier. 2021. “Swarm Intelligence: A Review of Optimization Algorithms Based on Animal Behavior.” In *Recent Advances of Hybrid Intelligent Systems Based on Soft Computing. Studies in Computational Intelligence*, 915:273–98. Springer International Publishing. https://doi.org/10.1007/978-3-030-58728-4_16.

Vent, W. 1975. “Rechenberg, Ingo, Evolutionsstrategie—Optimierung Technischer Systeme Nach Prinzipien Der Biologischen Evolution. 170 S. Mit 36 Abb. Frommann-Holzboog-Verlag. Stuttgart 1973. Broschiert.” Wiley Online Library.

Verma, Pradeepika, and Hari Om. 2019. “MCRMR : Maximum Coverage and Relevancy with Minimal Redundancy Based Multi-Document Summarization” 120: 43–56.

Yang, Xin She, and Xingshi He. 2013. “Firefly Algorithm: Recent Advances and Applications.” *International Journal of Swarm Intelligence* 1 (1): 36. https://doi.org/10.1504/ijsi.2013.055801.

Yang, Zhen, Fei Yao, Huayang Sun, Yun Zhao, Yingxu Lai, and Kefeng Fan. 2013. “BJUT at TREC 2013 Temporal Summarization Track.” In *TREC*.

Zhang, Qingyang, Ronggui Wang, Juan Yang, Andrew Lewis, Francisco Chiclana, and Shengxiang Yang. 2019. “Biology Migration Algorithm: A New Nature-Inspired Heuristic Methodology for Global Optimization.” *Soft Computing* 23 (16): 7333–58. https://doi.org/10.1007/s00500-018-3381-9.

Zhang, Yong, Meng Joo Er, Rui Zhao, and Mahardhika Pratama. 2016. “Multiview Convolutional Neural Networks for Multidocument Extractive Summarization.” *IEEE Transactions on Cybernetics* 47 (10): 3230–42.

Zhong, Sheng-hua, Yan Liu, Bin Li, and Jing Long. 2015. “Query-Oriented Unsupervised Multi-Document Summarization via Deep Learning Model.” *Expert Systems with Applications* 42 (21): 8146–55.