Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Assessing the effectiveness of ROUGE as unbiased metric in Extractive vs. Abstractive summarization techniques

Alessia Auriemma Citarella[1], Marcello Barbella[1], Madalina G. Ciobanu[1], Fabiola De Marco[1], Luigi Di Biasi[1], Genoveffa Tortora*,[1]

Department of Computer Science, University of Salerno, 84084, Fisciano (SA), Italy

ARTICLE INFO

Keywords:
Automatic text summarization algorithms
Extractive
Abstractive
ROUGE metric

ABSTRACT

Approaches to Automatic Text Summarization try to extract key information from one or more input texts and generate summaries whilst preserving content meaning. These strategies are separated into two groups, Extractive and Abstractive, which differ in their work. The extractive summarization extracts sentences from the document text directly, whereas the abstractive summarization creates a summary by interpreting the text and rewriting sentences, often with new words. It is important to assess and confirm how similar a summary is to the original text independently of the particular TS algorithm adopted. The literature proposes various metrics and scores for evaluating text summarization results, and ROUGE (Recall-Oriented Understudy of Gisting Evaluation) is the most used. In this study, our main objective is to evaluate how the ROUGE metric performs when applied to both Extractive and Abstractive summarization algorithms. We aim to understand its effectiveness and reliability as an independent and unbiased metric in assessing the quality of summaries generated by these different approaches. We conducted a first experiment to compare the metric efficiency (ROUGE-1, ROUGE-2 and ROUGE-L) for evaluating Abstractive (word2vec, doc2vec, and glove) *versus* Extractive Text Summarization algorithms (textRank, lsa, luhn, lexRank), and a second one to compare the obtained score for two different summary approaches: a simple execution of a summarization algorithm *versus* a multiple execution of different algorithms on the same text. Based on our study, evaluating the ROUGE metric for Abstractive and Extractive algorithms revealed that it reaches similar results for the Abstractive and Extractive algorithms. Moreover, our findings indicate that multiple executions, based on the running of two text summarization algorithms sequentially on the same text, generally outperform single executions of a single text summarization algorithm.

1. Introduction

Today, a vast amount of textual data is available from various sources. In particular, extracting knowledge from long texts is becoming increasingly difficult for humans. The advancement of information technology, especially in Artificial Intelligence (AI), has led to the formation of even more complex data management and processing tools. New algorithms for analyzing and extracting the most important text information, even those created by humans, are constantly presented [1]. These methodologies, called Automatic Text Summarization (ATS), enable the production of summaries from any input text by combining their key concepts [2]. Text summarization is a critical process in information distillation. It plays a crucial role in condensing lengthy texts into concise forms while retaining essential information. This process enhances efficiency by enabling quick access to key content and supports decision-making across various domains.

There are two main groups of algorithms for extracting a summary from a text [3]:

- *Extractive Automatic Text Summarization* (EATS): they select phrases from the input text, choosing those that best cover all the key information and discarding redundancy;
- *Abstractive Automatic Text Summarization techniques* (AATS): they try to elaborate a new corpus, using different and more appropriate words and a different semantic composition, to output a simpler text.

* Corresponding author.

E-mail addresses: aauriemmacitarella@unisa.it (A. Auriemma Citarella), mbarbella@unisa.it (M. Barbella), mciobanu@unisa.it (M.G. Ciobanu), fdemarco@unisa.it (F. De Marco), ldibiasi@unisa.it (L. Di Biasi), tortora@unisa.it (G. Tortora).

[1] All authors have contributed equally to the paper.

https://doi.org/10.1016/j.jocs.2025.102571
Received 22 February 2024; Received in revised form 11 November 2024; Accepted 9 March 2025
Available online 18 March 2025
1877-7503/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

EATS, by definition, uses sections of the original text to produce a summary, whereas AATS tends to introduce additional words. Hence, extractive summarization should do much better because there may be more overlapping of n-grams. Several techniques for both methodologies are suggested in the literature, which uses both supervised and unsupervised algorithms [4].

Given the impracticality of manually evaluating text summarization for large-scale datasets, which can be labor-intensive, time-consuming, and subject to human bias, automatic evaluation methods have obtained significant attention [5–8]. These methods aim to provide scalable, consistent, and repeatable evaluations, facilitating the advancement of text summarization research and enabling the comparison of different algorithms on common ground.

The most widely used evaluation metric is ROUGE [9]. It focuses on the overlapping of n-grams (expressed as a numeric value) between the system and human summaries without regard for their semantic and syntactic accuracy. Thanks to its computational efficiency, ROUGE is particularly well-suited for large-scale evaluations, allowing for the rapid assessment of numerous summaries, especially in the context of AATS evaluation and combined with semantics [9,10].

Ensuring that summaries maintain the context and intent of the source material is crucial, as failing to do so can lead to misleading or incomplete summaries. Additionally, the ideal summary must be informative and comprehensive while also being coherent, readable, and concise [11]. Specific aspects of summarization performance that remain poorly understood include the ability to maintain semantic coherence and contextual accuracy in generated summaries. The types of performance metrics used can be both quantitative and qualitative, but the impact of different summarization techniques can create a gap in the evaluation of summaries, which should be both concise and comprehensive [12]. Extractive methods may retain unnecessary information, while abstractive methods risk losing key details. Assessing this balance accurately remains a difficulty in evaluation.

Addressing these challenges requires developing more sophisticated evaluation metrics or implementing robust methods for a fine and comprehensive assessment of summary quality [13].

This research focuses on determining how effective the ROUGE metric score is for assessing the quality of a summary, both for the EATS and AATS approaches, following two main research questions (RQs):

- **RQ1**: *How different is the ROUGE score for the EATS methods compared to the AATS ones? Can this metric score represent the quality of a summary generated by a TS algorithm?*
- **RQ2**: *How different is the multiple execution of a summary (execution of two TS algorithms in series on the same text, with the result of the first being used as input for the second) from the single execution (the summary is obtained with a single algorithm execution)? Is the ROUGE score appropriate in comparing the two methods?*

To this aim, the paper compares three different AATS (word2vec [14], doc2vec [15], and glove [16]) and four EATS (textRank [17], Latent Semantic Analysis-lsa [18], Luhn [19], lexRank) [20] on four widely used datasets: CNN Daily Mail, BBC News, HITG, and WCEP datasets to compare the metric efficiency of ROUGE-1, ROUGE-2, and ROUGE-L. The goal of our work is to provide a focused evaluation of ROUGE as an assessment tool specifically for the two distinct summarization paradigms: Extractive versus Abstractive summarization techniques. Our approach emphasizes the importance of using independent and unbiased metrics to ensure fair evaluation across these paradigms.

The paper is structured as follows. In Section, 2 we introduced an overview of the most commonly used TS evaluation metrics. Section 3 presents the experiment design and shows its organization, ensuring its replicability. The experiment results are reported in Section 4 and Section 5, while the threats to validity are explored in Section 6. Finally, Section 7 summarizes the findings and offers some suggestions for further research. For completeness, the Appendix covers fundamental concepts related to text representation and highlights recent advancements in Abstractive and Extractive techniques as reported in the literature. All abbreviations used are listed in Table 1.

Table 1
Abbreviations in this paper

Abbreviation	Explanation
AATS	Abstractive Automatic Text Summarization techniques
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
DL	Deep Learning
EATS	Extractive Automatic Text Summarization techniques
GRU	Gated Recurrent Unit
LCS	Longest Common Subsequence
LSTM	Long Short Term Memory
NER	Named entity recognition
NLP	Natural Language Processing
POS	Part of speech
RBM	Restricted Boltzmann Machine
RNN	Recurrent Neural Network
SCU	Summary Content Unit
SSAS	Semantic Similarity for Abstractive Summarization
TS	Text Summarization

2. State of the art of evaluation methods

An evaluation of a summary for a human is a frequent task. In fact, by reading and comparing two texts, one can determine the summary's quality by considering which is more specific, covers more key concepts, or at least is more readable and grammatically correct.

The process of creating a summary is more easily compared to an Abstractive TS task than to an Extractive one because when creating a summary, a human can try to express his thoughts with new words and phrases after carefully reading and understanding one or more source texts, attempting to cover as many topics as possible from the original text; this type of work requires a lot of creativity, and the results might vary greatly from person to person.

In conclusion, the lexical composition is crucial since a notion can be represented in various ways, utilizing distinct phrases and words. The various evaluation criteria are depicted in Fig. 1, classified into intrinsic and extrinsic types. This paragraph focuses on the intrinsic TS evaluation methods, including a detailed discussion of the ROUGE measure, which is the most used.

2.1. ROUGE

The acronym ROUGE stands for "*Recall-Oriented Understudy of Gisting Evaluation*," and refers to a collection of criteria for assessing automatically generated texts. It is usually used to evaluate the quality of the summary of a TS algorithm, thanks to its computational efficiency.

To operate, ROUGE compares a machine-generated summary (sometimes referenced to as a system summary) to one created by a human (sometimes called gold standard or reference summary). We can use *Precision* and *Recall* measures to estimate this metric.

- *Precision* determines how concise the system summary is and how many superfluous words are in the corpus.
- *Recall* determines how much of the system summary is covered by the reference summary.

The ROUGE metric refers to a set of distinct ways to quantify the quality of a system summary. ROUGE measures can be calculated using various methods, based on the different granularities. The following are the most commonly used:

1. ROUGE-N refers to the overlapping of N-grams (1-gram, 2-gram, 3-gram, and so on) between the system summary and the reference summary;
2. ROUGE-L measures the longest common word sequence, computed by the Longest Common Subsequence (LCS) algorithm;

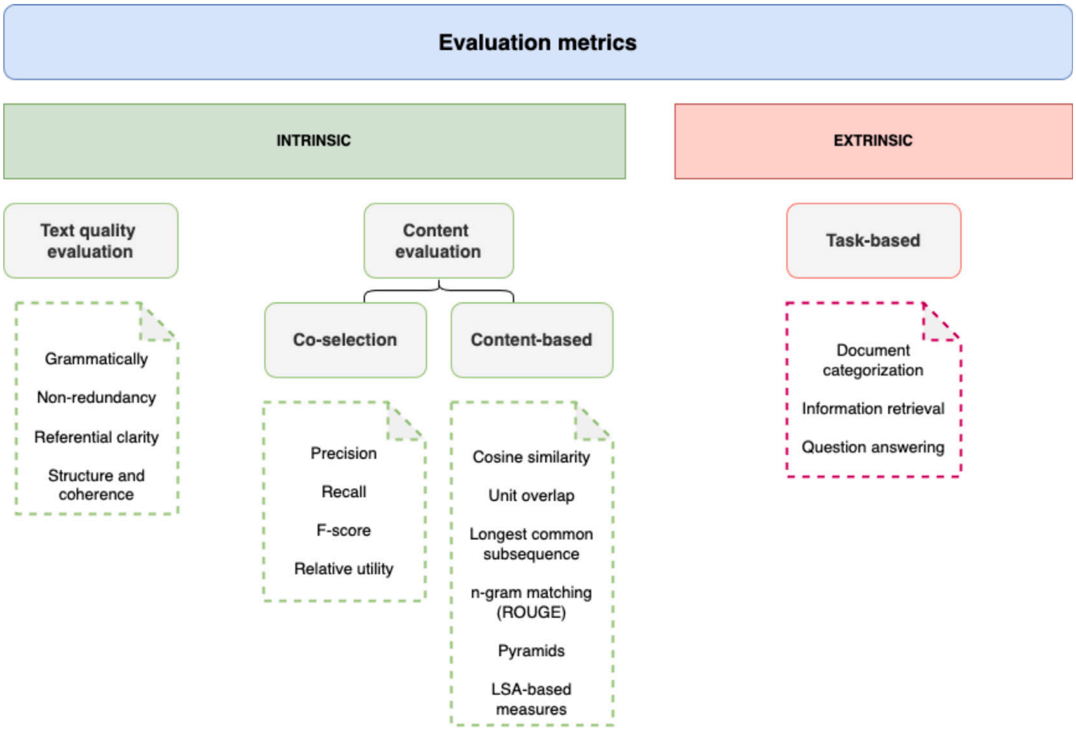


Fig. 1. Evaluation metrics overview.

3. ROUGE-S refers to a couple of words in an ordered sentence that allows some gaps. Sometimes, this measure is also called skip-gram;
4. ROUGE-SU is a weighted mean between ROUGE-S and ROUGE-L.

ROUGE-1, ROUGE-2 and ROUGE-L are the most commonly used metrics in the literature since they reflect the granularity of the studied texts. The overlap of unigrams (single words) and bigrams (two consecutive words) between the generated summary and the reference summary are measured, respectively, by ROUGE 1 and ROUGE-2.

2.2. Pyramid

Pyramid is proposed as a novel way to analyze automatically generated texts [21]. The fundamental idea is to find some little units called Summary Content Units (SCU) that will be used to compare the data in the original text.

Every SCU is a one-sentence-long text. Each one is assigned a weight based on how frequently it appears in the various texts under consideration. It is reasonable to expect a small number of SCUs with a large weight and a growing number of SCUs with a small weight. This method works well when there are multiple texts from which SCUs can be derived. As it is a hierarchy, this structure suggests the method's name.

The pyramid's construction is depicted in Fig. 2. The approach can be synthesized in the following steps:

1. **Enumeration**: the SCUs for each sentence from the peer summary are listed;
2. **Pyramid generation**: each SCU is partitioned using a pyramidal hierarchy scheme, with the SCUs having the same weight at each level. A pyramid was also created for the reference summary;
3. **Scoring**: a ratio is calculated between the sum of the weights of the SCU of the system pyramid and the reference one. The resulting values range from 0 to 1, 1 indicating that most of the content has a high weight.

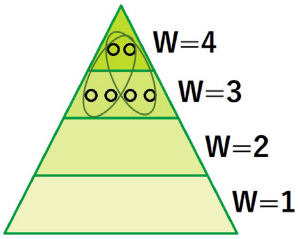


Fig. 2. The Pyramidal hierarchy.

Although this method appears to be extremely valid for evaluating the summary generated by TS Algorithms, it has not had much success compared to the ROUGE metric.

2.3. Semantic similarity for abstractive summarization (SSAS)

Unfortunately, statistical approaches fail to detect semantic inconsistencies within a text and other natural language elements like paraphrase and logic repercussions. So, novel techniques to deal with this challenge have been proposed in the literature, particularly for the AATS algorithms. SSAS is a metric that emphasizes the semantic relationships between the system summary and the reference one. Here, we can see a general overview of how the SSAS method is used to create a score:

- First, all SCUs from the system summary (S) and the reference one (R) are removed from the text. The two sets will have cardinality n and m, respectively;
- After that, the corpus is used to extract various natural language features, such as inference. A classification inference model is used to train the weights of various combinations of these qualities to arrive at a final score;
- Finally, a normalization is made, and the results are ranked.

However, this approach, like Pyramid, also failed to gain traction in the literature due to its high computational complexity.

2.4. Strengths & weaknesses

ROUGE is straightforward to implement and widely used in summarization tasks. It provides a standard metric for consistent comparison across different models and studies. It is also computationally efficient, making it suitable for large-scale evaluations. On the other hand, ROUGE primarily measures lexical overlap, which can disadvantage abstractive summarization methods that use different wordings, and it does not consider the semantic meaning, focusing only on word matches. Pyramid assesses the presence of SCUs in the summary, providing a subtle understanding of how well the summary covers the content. It aligns closely with human evaluation criteria by involving human annotators to identify key content units, enhancing its reliability. However, creating SCUs and evaluating summaries is time-consuming and resource-intensive. Identifying SCUs and their importance can be subjective, leading to potential inconsistencies. Moreover, applying the Pyramid method to large datasets is challenging because it requires manual annotation [22]. SSAS evaluates how well the original text's meaning is preserved in the summary, which is crucial for abstractive summarization. Semantic similarity can handle different wordings and phrasings, making it suitable for evaluating abstractive summaries. Leveraging modern Natural Language Processing (NLP) techniques, such as embeddings and transformer models, can enhance the accuracy of semantic similarity measures. One major open issue with SSAS is that calculating semantic similarity can be computationally intensive, especially with large datasets. The quality of the evaluation depends on the performance of the underlying NLP models, which may vary. Capturing deep contextual and slight meanings can still be challenging, and semantic similarity measures might not always align perfectly with human judgments. (rifrasare)Despite its weaknesses, the ROUGE metric remains a popular choice for evaluating text summarization for several reasons. It provides quick, initial assessments and is a standard for benchmarking and comparing models. Empirical validation of the ROUGE metric shows reasonable correlation with human judgments [9,23], as also recently reported by *Zhang et al.* [10]. It complements other metrics, offering a practical baseline performance measure that helps guide model development. Regarding the evaluation metrics mentioned above, we can identify their open issues and weaknesses, as summarized in Fig. 3. In conclusion, several open issues and challenges highlight the need for ongoing innovation and refinement in summarization evaluation metrics. To ensure these metrics provide comprehensive, reliable, and efficient assessments of summarization models, the main challenges include developing metrics that account for semantic meaning rather than just word matches and enabling consistent and objective evaluations. Additionally, balancing the need for computational efficiency with the demand for comprehensive and accurate quality assessments is crucial.

3. Design and organizations of the experiments

3.1. Experiment planning

The first RQ examines the ROUGE metric's validity and accuracy for the EATS and AATS algorithms. Instead, in the second RQ, single and multiple executions of TS algorithms will be compared to assess their efficacy using the ROUGE metric. The planning phase details the various steps of the experiment.

3.1.1. Hypotheses formulation

Two hypotheses have been proposed for the statistical analysis of the experiment:

RQ1 *Null Hypothesis*: The AATS methods perform differently than the EATS approaches.

$$H_0 : \mu_{ROUGE\_Ext} \neq \mu_{ROUGE\_Abs} \tag{1}$$

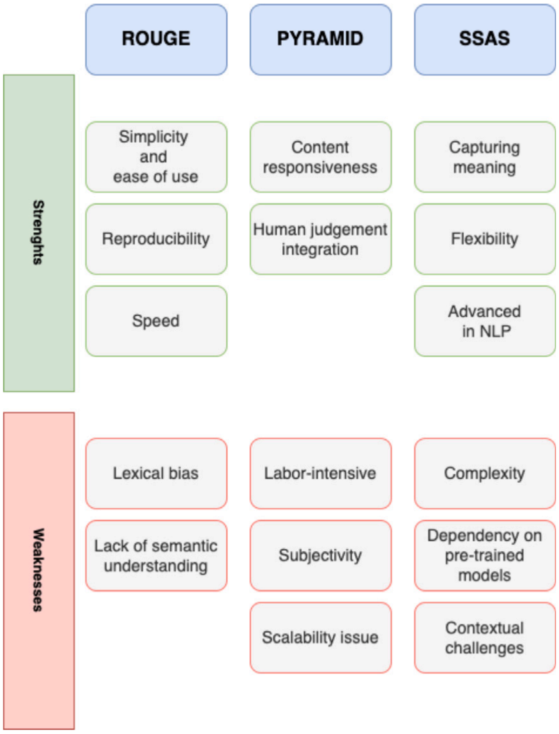


Fig. 3. Strengths and Weaknesses.

where μ is the mean and ROUGE the score of each summary.

This is because, in contrast to Abstractive methods, which utilize new words in the generated summary and therefore different N-grams, Extractive methods use sections of the original text in the output summary, which should provide a different overlap ratio of n-grams.

Alternative Hypothesis: The AATS methods perform almost as well as the EATS approaches.

$$H_A : \mu_{ROUGE\_Ext} = \mu_{ROUGE\_Abs} \tag{2}$$

where μ is the mean and ROUGE the score of each summary.

RQ2 (a) *Null Hypothesis*: A multiple execution of TS algorithms on the same text produces less or the same results as a single execution on the same text

$$H_0 : \mu_{ROUGE\_Multiple} \leq \mu_{ROUGE\_Single} \tag{3}$$

where μ is the mean and ROUGE the score of each summary;

(b) *Alternative Hypothesis*: A multiple execution of TS algorithms on the same text produces better results than a single execution on the same text

$$H_A : \mu_{ROUGE\_Multiple} > \mu_{ROUGE\_Single} \tag{4}$$

where μ is the mean and ROUGE the score of each summary.

3.1.2. Variable selection

The selection of variables is an important phase in the experiment planning process. The independent variables are those we control and can change during the experiment. The dependent variables, on the other hand, assess the experiment's impact on various combinations of independent variables.

For **RQ1** we set the *independent variables*, represented by the EATS and AATS approaches. Different EATS and AATS algorithms will be used.

For **RQ2**, we have that the *independent variables* are the TS techniques that included single and multiple executions of the summary. Different algorithms will be used in different combinations for each of them. For both RQ1 and RQ2 the *dependent variables* are represented by the ROUGE score for the output of each algorithm. To provide a single comparable measure, the findings will be averaged.

3.1.3. Subjects selection

When conducting an experiment, choosing subjects is essential since this is interconnected with generalizing the experiment's findings. The population must be represented in the selection through an appropriate sample size to accomplish this. Probability or non-probability approaches can be used for sampling. In our case, both RQ1 and RQ2 use the *Simple Random Sampling* model, in which subjects are randomly chosen from a population list. In particular: for RQ1 tests are performed on four different datasets: CNN Daily Mail, BBC News, HITG, and WCEP.

CNN Daily Mail contains editorial news and articles from CNN and the Daily Mail. Approximately 287.000 items, including summaries, comprise this dataset, first made available for Abstractive Summarization. It is one of the most widely used datasets for ATS algorithms evaluation [24].

BBC News is a popular extractive text summarization dataset that includes documents taken from the BBC news website regarding five main themes covered in publications from 2004 to 2005 [25].

HITG[2] is a dataset of abstracts with approximately 100.000 texts, each including different supplementary information to the publications and a brief summary.[3]

WCEP is composed of multi-document summaries obtained from the Current Events Portal of Wikipedia [26]. Short, human-written descriptions of new events are included in each summary, and each is paired with a selection of new items pertinent to the event.[4]

In each experiment, the summarized texts are chosen randomly from every referenced dataset. Every algorithm is executed on 40 blocks of 1000 texts each (40.000 summaries for each dataset).

For RQ2, due to the computational difficulty and the time required to complete the experiment, tests are conducted in this case only on the very large *CNN Daily Mail* dataset, considering 1000 documents and summaries.

3.1.4. Design type choice and tools

A sequence of tests makes up an experiment. The set of tests must be carefully planned and constructed to get the most out of the experiment. The way the tests are organized and executed is described in the experiment design. So, in this section, our test methodology is described.

Principle general design. The choice has been made to employ randomization and balancing approaches. Random blocks of data are used to conduct tests. Each test will be conducted using a block of 1000 texts to be examined for the balancing design principle. This allows very good results and statistically valid conclusions for each test.

Table 2
Descriptive statistics for the CNN Daily Mail dataset.

ROUGE Metric	Mean	Median	St Dev
ROUGE-1	0.205	0.194	0.002
ROUGE-2	0.059	0.041	0.002
ROUGE-L	0.204	0.189	0.003

Standard design type. As the Design Type for RQ1, a factor with two treatments was chosen. Indeed, we aim to compare the EATS and AATS techniques through these activities. The same design type is used for RQ2. We are particularly interested in comparing the performance of a single versus a multiple execution of summaries. All of the algorithms that have been considered are used in each experiment. In particular, we examine the execution of an EATS algorithm followed by the execution of an AATS algorithm for multiple tasks (and vice versa). Python-based software has been developed to execute the experiments.

3.2. Operation phase

The experiment operation phase consists of Preparation, Execution, and Data Validation.

Preparation. In this phase, the correctness of the code that will extract the random texts from the dataset, the ROUGE metric scores, and the algorithm setting must all be checked for the experiment to run. It is also essential to build the code to collect the results. The mean scores for each block of summaries are stored in a dataset containing all calculated scores.

Execution. Due to the calculation time required to execute tests, the experiment persisted for many days. The algorithms for RQ1 were run in parallel, grouped according to the TS technique, and given the same input texts. Sequential computation of the scores was adopted for RQ2. Initially, dataset texts were randomly chosen for both RQs and summarized using various algorithms. Finally, the ROUGE metric was applied to all of the summaries.

Data Validation. Data validation was carried out by randomly examining selected entries and ensuring that the CSV files were consistent. The ROUGE scores of the samples were also tested to see whether they met the criterion set by the algorithm creators.

In Fig. 4, we illustrated the work of the entire process. The workflow begins with the collection of text documents from four datasets. Various summarization techniques are applied, including extractive and abstractive methods. The generated summaries are assessed using multiple metrics, especially ROUGE scores (ROUGE-1, ROUGE-2 and ROUGE-L) to address RQ1. Subsequently, we conducted single and multiple executions to answer RQ2. Algorithms 1 and 2 describe the pseudocode for the experiments.

4. Results analysis for RQ1

This section describes, evaluates, and interprets the results of the two experiments, with some graphs highlighting their statistical validity. We started from the experiments on CNN Daily Mail. The experiments on the other three datasets (BBC News, HITG e WCEP) further supported the findings on the *CNN Daily Mail* dataset. For the experiments, 40 blocks were evaluated for these datasets, each consisting of 1000 summaries. To illustrate the results, we first calculated an average value within each block and then determined an overall average for each algorithm.

4.1. Results on CNN daily mail

In this section, we get into the specifics of what our experiment on the *CNN Daily Mail* dataset revealed. Following that, the results for the final three datasets will be summarized. Considering that the texts are chosen randomly from the dataset, some main elements of the results

[2] https://www.kaggle.com/sunnysai12345/news-summary

[3] Only news pieces from the Hindu, Indian Times, and Guardian, as well as the compressed news from Inshorts, were scraped, from February to August 2017.

[4] These items are composed of content automatically retrieved from the Common Crawl News collection and sources cited by WCEP editors.

Algorithm 1 First Experiment - Single Execution

```
for each metrics in Metrics do:
    for each Extractive_Algorithm in ExtAlgorithm do:
        for each original_text do:
            for each Abstractive_Algorithm in AbsAlgorithm do:
                $ExtSum \leftarrow$ CalculateSummary($ExtAlgorithm[original\_text]$)
                $AbsOnExt[original\_text] \leftarrow$ CalculateSummary($AbsAlgorithm[ExtSum]$)
                $Score\_AbsOnExt[original\_text] \leftarrow$ CalculateScore($AbsOnExt[original\_text]$)
            end for
            $AbsOnExtMean[ExtAlgorithm] \leftarrow$ CalculateMean($Score\_AbsOnExt[]$)
        end for
        $MeanExt \leftarrow$ CalculateMean($MeanExtAlgorithm[]$)
    end for

    for each Abstractive_Algorithm in AbsAlgorithm do:
        for each original_text do:
            $AbsSum[original\_text] \leftarrow$ CalculateSummary($AbsAlgorithm[original\_text]$)
            $Score\_AbsSum[original\_text] \leftarrow$ CalculateScore($AbsSum[original\_text]$)
        end for
        $MeanbsAlgorithm[ExtAlgorithm] \leftarrow$ CalculateMean($Score\_AbsSum[]$)
    end for
    $MeanAbs \leftarrow$ CalculateMean($MeanAbsAlgorithm[]$)
    CompareScores($metric, MeanExt, MeanAbs$)
end for
```

Algorithm 2 Second Experiment - Multiple Execution

```
for each metrics in Metrics do:
    for each Extractive_Algorithm in ExtAlgorithm do:
        for each original_text do:
            $ExtSum[original\_text] \leftarrow$ CalculateSummary($ExtAlgorithm[original\_text]$)
            $Score\_ExtSum[original\_text] \leftarrow$ CalculateScore($ExtSum[original\_text]$)
        end for
        $AbsOnExtMean[ExtAlgorithm] \leftarrow$ CalculateMean($Score\_AbsOnExt[]$)
        $MeanExt \leftarrow$ CalculateMean($AbsOnExtMean[]$)
    end for

    for each Abstractive_Algorithm in AbsAlgorithm do:
        for each original_text do:
            for each original_text do:
                $AAbsSum \leftarrow$ CalculateSummary($AbsAlgorithm[original\_text]$)
                $ExtOnAbs[original\_text] \leftarrow$ CalculateSummary($ExtractiveAlgorithm[AbsSum]$)
                $Score\_ExtOnAbs[original\_text] \leftarrow$ CalculateScore($ExtOnAbs[original\_text]$)
            end for
        end for
        $ExtOnAbsMean[AbsAlgorithm] \leftarrow$ CalculateMean($Score\_ExtOnAbs[]$)
    end for
    CompareScores($metric, MeanExt, MeanAbs$)
end for
```

achieved are presented below, starting from the hypothesis of the same distribution for each block of summaries. A random execution of the textRank algorithm is investigated for this purpose. Table 2 presents mean, median, and standard deviation values for the three types of ROUGE measures.

First, a boxplot and a histogram of randomly generated results were used to assess the distribution of the outcomes (each one refers to a collection of 1000 summaries that can differ depending on the algorithm and input texts). The 1000 scores for ROUGE-1, ROUGE-2, and ROUGE-L obtained from the textRank algorithm execution are shown in Fig. 5.

We can note that ROUGE-2 values are smaller than ROUGE-1 and ROUGE-L. A notable number of outliers are present, especially at the top of the boxplot, for all three ROUGE types. In ROUGE-1 and ROUGE-L, the median line divides the box borders almost equally, indicating well-centered data in the interquartile range. ROUGE-2 shows positive skewness, evident from the median line near the bottom edge of the box and the shorter whisker. The symmetric whiskers and box edges in ROUGE-1 and ROUGE-L suggest a potential normal distribution for these measures. Instead, the subsequent Fig. 6 shows each ROUGE measure distribution by three representative histograms. As anticipated by the boxplots, ROUGE-1 and ROUGE-L approximate the normal distribution quite well. This guarantees the good distribution of data points along all the observations and allows us to consider the mean as a valid representation measure for them.

The first research question was to determine the usefulness of the ROUGE metric in evaluating TS algorithms. The experiment design guidelines used a random selection of texts to compare the results of both the EATS and AATS methodologies. Table 3 summarizes all the algorithms used in this experiment, reporting the relative mean and standard deviation for each of the three ROUGE measures.

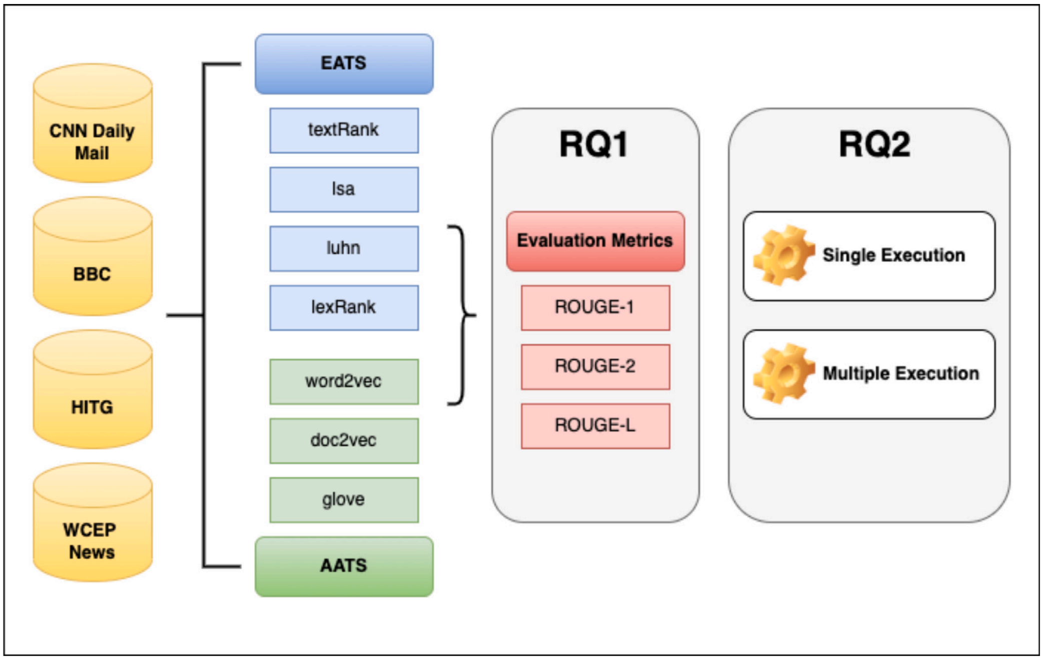


Fig. 4. Workflow of the process.

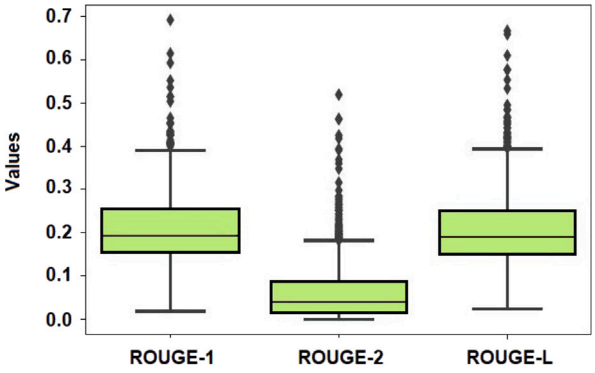


Fig. 5. Boxplot of ROUGE metric scores computed on 1000 summaries by the textRank algorithm for the CNN Daily Mail dataset.

Table 3
Mean and Standard Deviation for all the algorithms and ROUGE metrics used for the CNN Daily Mail dataset.

Algorithm	ROUGE-1		ROUGE-2		ROUGE-L	
Extractive	Mean	St Dev	Mean	St Dev	Mean	St Dev
textRank	0.205	0.002	0.060	0.002	0.204	0.003
lsa	0.223	0.004	0.056	0.003	0.205	0.004
luhn	0.220	0.003	0.066	0.002	0.220	0.003
lexRank	0.242	0.003	0.071	0.002	0.232	0.003
Mean	**0.22**		**0.06**		**0.21**	
Abstractive	Mean	St Dev	Mean	St Dev	Mean	St Dev
word2vec	0.213	0.003	0.058	0.002	0.205	0.003
doc2vec	0.215	0.002	0.059	0.002	0.206	0.002
glove	0.213	0.003	0.058	0.002	0.205	0.003
Mean	**0.21**		**0.06**		**0.21**	
Average	*0.219*	*0.003*	*0.06*	*0.002*	*0.211*	*0.003*

Fig. 7 shows the average score measured using ROUGE-1, ROUGE-2, and ROUGE-L. The first four algorithms are Extractive (textRank, lsa, luhn, and lexRank), whereas the last three (glove, word2vec, and doc2vec) are Abstractive.

For the Extractive methods, lexRank is generally the best-performing algorithm, scoring approximately 10% more than others. In particular, with the use of lexRank, ROUGE-1, ROUGE-2 and ROUGE-L achieved a mean of 0.242, 0.071 and 0.232, respectively. The Abstractive methods, on the other hand, have extremely comparable values, even if somewhat all of their scores are below the mean. Specifically, using doc2vec, ROUGE-1, ROUGE-2, and ROUGE-L reached mean scores of 0.215, 0.059, and 0.206, respectively. The average results for the EATS and AATS for the three ROUGE metrics are similar but lexRank outperforms all the algorithms. A statistical validity test was also performed to confirm or reject the hypothesis. To that purpose, a t-test was run on the distribution of findings for each summary, which was paired for Abstractive and Extractive. For this test, the freedom degrees are the same as the observed population of 40.000 summaries. The p-value for the statistical validity of the experiment is 2.2e−16, which is less than the needed 0.05. This supports the alternative hypothesis that the Extractive and Abstractive ROUGE scores are equivalent.

The assumption was that Extractive methods would perform far better than Abstractive methods. Instead, the findings revealed that this assumption is incorrect. Both AATS and EATS algorithms performed similarly in the majority of cases. Some considerations can be made on these results. From the findings, it is evident that ROUGE does not enable us to distinguish the summaries from EATS and AATS, depending on their value. Indeed, because ROUGE compares a system-generated summary to a human-written one, and the score is determined by a statistical computation based on the number of n-grams overlapping the two texts, the more the summaries utilize different words, the worse the ROUGE metric performs. However, this system ignores the semantics of statements. When comparing different summaries generated by humans from the same source text, the results can vary widely but still be valid and acceptable. These summaries typically have excellent readability and strong syntactic composition. However, calculating the ROUGE score between two gold standards might not yield high scores. This is because each human summary can use different words, text structures, and emphasize different subjects. ROUGE fails to account for these variations and thus may not accurately reflect the essential elements of a summary. According to the obtained ROUGE score, it does not allow us to distinguish the summaries from EATS and AATS.

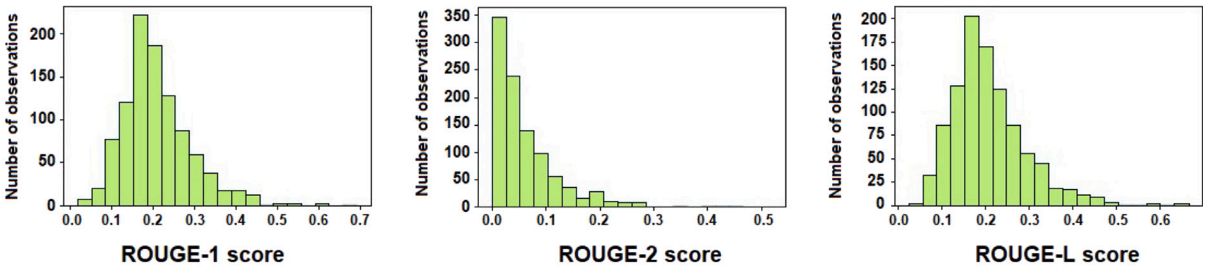


Fig. 6. Histogram showing the data distribution for ROUGE-1, ROUGE-2, and ROUGE-L scores using the textRank algorithm for the CNN Daily Mail dataset.

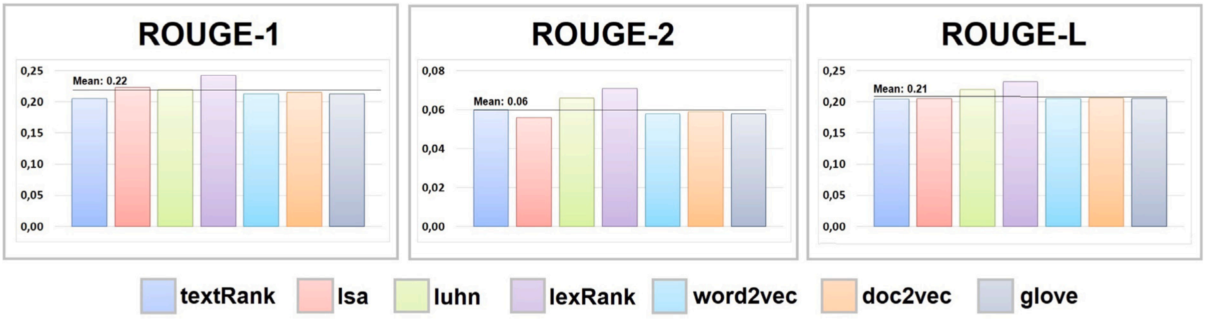


Fig. 7. ROUGE average scores of the experiment conducted on Abstractive and Extractive algorithms.

Table 4
Mean and Standard Deviation (and overall average) for all the algorithms and ROUGE metrics used for the BBC dataset.

Algorithm	ROUGE-1		ROUGE-2		ROUGE-L	
Extractive	Mean	St Dev	Mean	St Dev	Mean	St Dev
textRank	0.318	0.162	0.238	0.156	0.314	0.164
lsa	0.237	0.150	0.146	0.136	0.231	0.151
luhn	0.337	0.167	0.261	0.169	0.332	0.169
lexRank	0.298	0.152	0.220	0.143	0.294	0.153
Mean	**0.29**		**0.22**		**0.29**	
Abstractive	Mean	St Dev	Mean	St Dev	Mean	St Dev
word2vec	0.289	0.137	0.202	0.133	0.283	0.139
doc2vec	0.288	0.137	0.201	0.133	0.282	0.139
glove	0.296	0.135	0.207	0.133	0.290	0.138
Mean	**0.29**		**0.20**		**0.29**	
Average	*0.295*	*0.148*	*0.211*	*0.143*	*0.289*	*0.150*

Table 5
Mean and Standard Deviation (and overall average) for all the algorithms and ROUGE metrics used for the HITG dataset.

Algorithm	ROUGE-1		ROUGE-2		ROUGE-L	
Extractive	Mean	St Dev	Mean	St Dev	Mean	St Dev
textRank	0.363	0.261	0.127	0.174	0.315	0.240
lsa	0.352	0.266	0.124	0.174	0.306	0.244
luhn	0.342	0.258	0.116	0.169	0.297	0.235
lexRank	0.540	0.215	0.223	0.190	0.470	0.215
Mean	**0.40**		**0.15**		**0.35**	
Abstractive	Mean	St Dev	Mean	St Dev	Mean	St Dev
word2vec	0.408	0.259	0.148	0.180	0.353	0.239
doc2vec	0.381	0.260	0.134	0.176	0.329	0.239
glove	0.396	0.259	0.142	0.179	0.343	0.239
Mean	**0.40**		**0.14**		**0.34**	
Average	*0.397*	*0.254*	*0.145*	*0.178*	*0.345*	*0.236*

4.2. Results on BBC news

The results obtained from the experimentation on this dataset are shown below. The distribution of each ROUGE measure is shown in Fig. 8, by three representative histograms. We can note that ROUGE-1 and ROUGE-L approximate, albeit not perfectly, the normal distribution. This guarantees the good distribution of data points along all the observations and allows us to consider the mean as a valid representation measure for them. Table 4 shows the mean and standard deviation for the seven algorithms analyzed concerning the three ROUGE measures and their overall average in the last row. Fig. 9 provides a graphical representation of these results, which more easily highlights the minimal differences between the various scores obtained by the algorithms (the first four Extractive and the last three Abstractive).

All algorithms score very close to each other. For the Extractive methods, luhn is the best-performing algorithm, scoring about 14% above the average. In particular, ROUGE-1, ROUGE-2 and ROUGE-L reached a mean of 0.337, 0.261 and 0.332, respectively. For Abstractive methods, the best results are obtained by glove algorithm with a mean of 0.296, 0.207 and 0.290 with ROUGE-1, ROUGE-2 and ROUGE-L, respectively. The average results for both methods are very similar also in this scenario and, overall, the best algorithm is luhn.

4.3. Results on HITG

The results obtained from the experimentation on this dataset are shown below. The distribution of each ROUGE measure is shown in Fig. 10, by three representative histograms. In this case, we can note that none of the ROUGE measures approximate the normal distribution. So this dataset is unreliable according to the ROUGE metric, even if the results are not very different from the other datasets considered. Table 5 shows the mean and standard deviation for the seven algorithms analyzed concerning the three ROUGE measures and their overall average in the last row. Fig. 11 provides a graphical representation of these results, which more easily highlights the differences between the various scores obtained by the algorithms (the first four Extractive and the last three Abstractive). We can note that, for the Extractive methods, lexRank is the best algorithm. In particular, we reached a mean ROUGE-1, ROUGE-2 and ROUGE-L of 0.540, 0.223 and 0.470, respectively, for lexRank. For Abstractive methods, the best results are

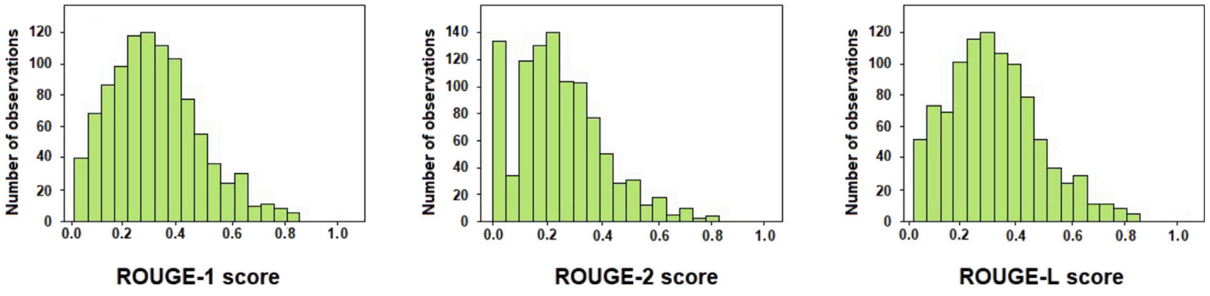


Fig. 8. Histogram showing the data distribution for ROUGE-1, ROUGE-2, and ROUGE-L scores using the textRank algorithm for the BBC dataset.

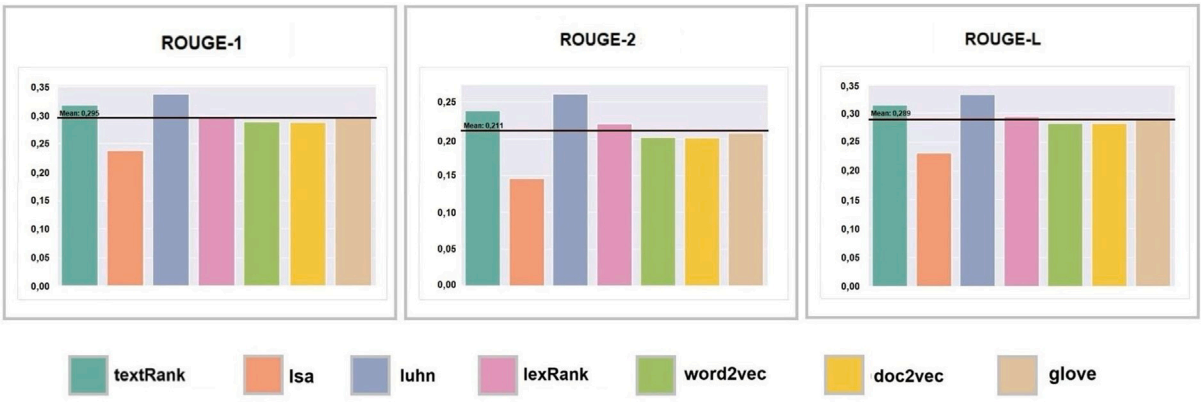


Fig. 9. ROUGE average scores of the experiment conducted on Abstractive and Extractive algorithms for the BBC dataset.

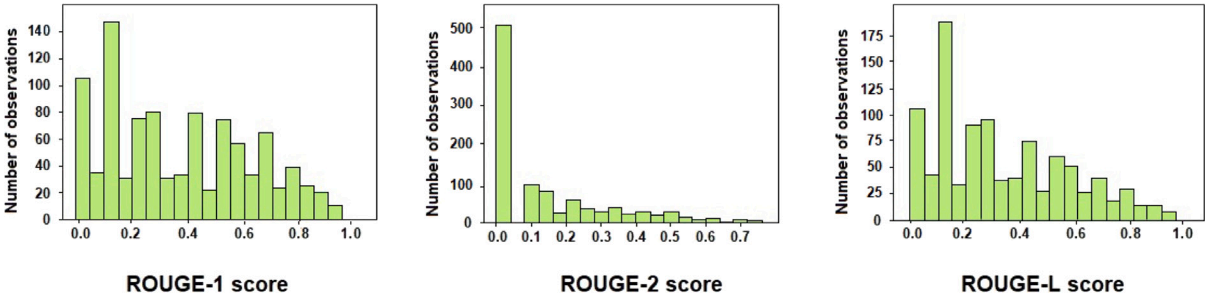


Fig. 10. Histogram showing the data distribution for ROUGE-1, ROUGE-2, and ROUGE-L scores using the textRank algorithm for the HITG dataset.

obtained by word2vec with a mean of 0.408, 0.148 and 0.353 with ROUGE-1, ROUGE-2 and ROUGE-L, respectively. Also in this case, the average results for both methods are very similar and, overall, the best algorithm is lexRank.

All algorithms score very close to each other (excluding lexRank, which shows an outlier score about 35% better than average).

4.4. Results on WCEP dataset

The results obtained from the experimentation on this dataset are shown below. The distribution of each ROUGE measure is shown in Fig. 12, by three representative histograms. We can note that ROUGE-1 and ROUGE-L approximate the normal distribution quite well. This guarantees the good distribution of data points along all the observations and allows us to consider the mean as a valid representation measure for them.

Table 6 shows the mean and standard deviation for the seven algorithms analyzed concerning the three ROUGE measures and their overall average in the last row. Fig. 13 provides a graphical representation of the results on the WCEP dataset, highlighting the minimal differences between the various scores obtained by the algorithms (the first four Extractive and the last three Abstractive).

Table 6
Mean and Standard Deviation (and overall average) for all the algorithms and ROUGE metrics used for the WCEP dataset.

Algorithm	ROUGE-1		ROUGE-2		ROUGE-L	
Extractive	Mean	St Dev	Mean	St Dev	Mean	St Dev
textRank	0.253	0.137	0.057	0.086	0.206	0.123
lsa	0.192	0.125	0.033	0.064	0.155	0.107
luhn	0.265	0.134	0.061	0,085	0.215	0.119
lexRank	0.253	0.140	0.060	0,088	0.206	0.125
Mean	**0.24**		**0.05**		**0.20**	
Abstractive	Mean	St Dev	Mean	St Dev	Mean	St Dev
word2vec	0.237	0.129	0.049	0.079	0.191	0.114
doc2vec	0.239	0.127	0.048	0.078	0.194	0.113
glove	0.242	0.129	0.049	0.080	0.195	0.114
Mean	**0.24**		**0.05**		**0.19**	
Average	*0.240*	*0.132*	*0.051*	*0.080*	*0.195*	*0.116*

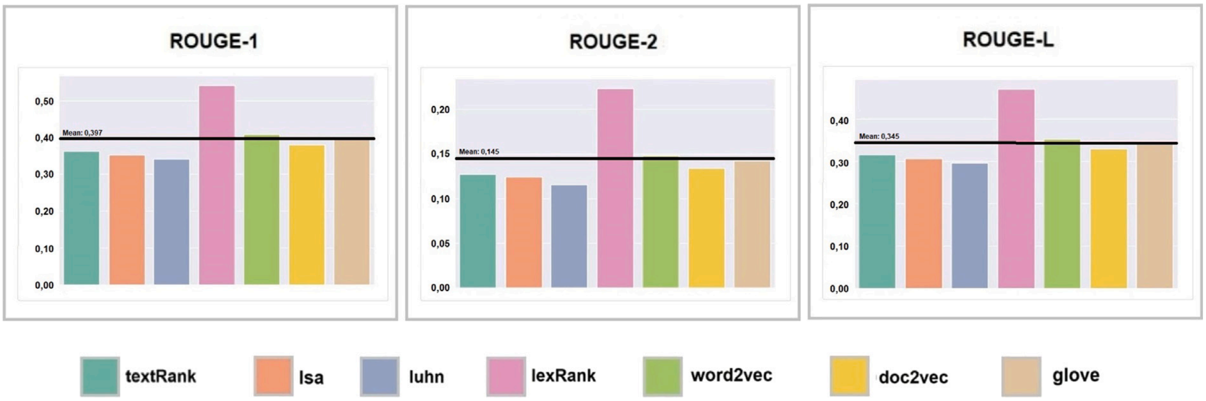


Fig. 11. ROUGE average scores of the experiment conducted on Abstractive and Extractive algorithms for the HITG dataset.

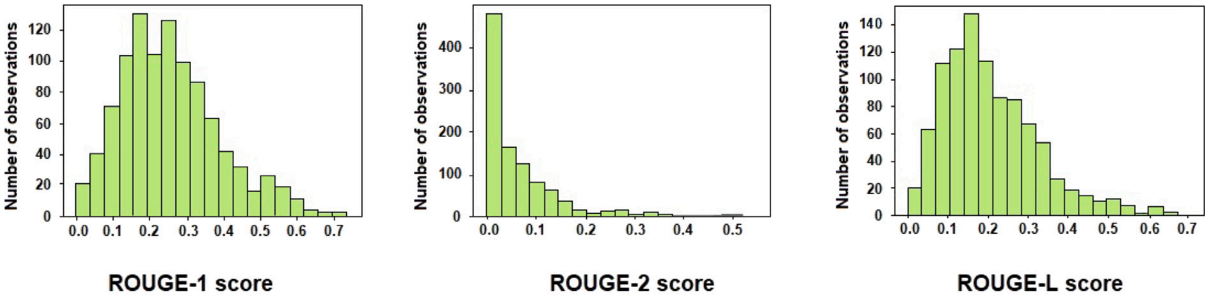


Fig. 12. Histogram showing the data distribution for ROUGE-1, ROUGE-2, and ROUGE-L scores using the textRank algorithm for the WCEP dataset.

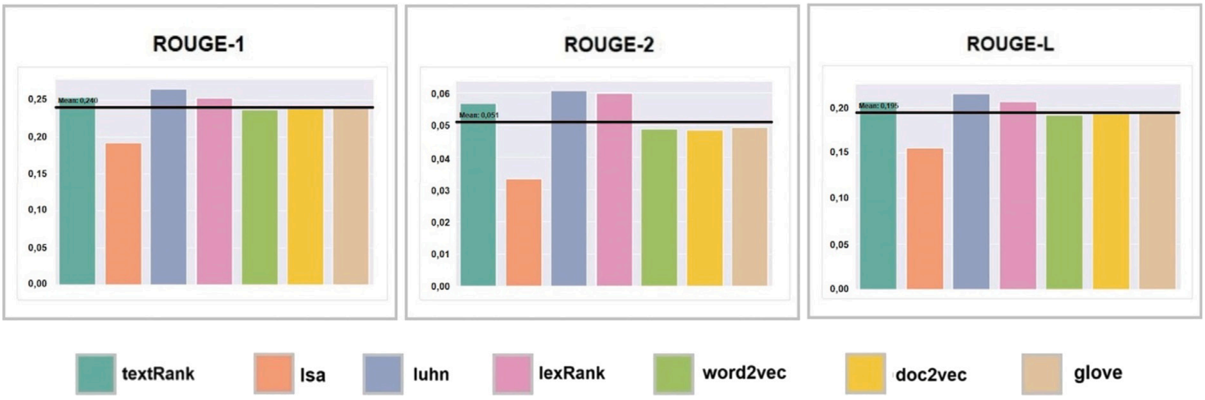


Fig. 13. ROUGE average scores of the experiment conducted on Abstractive and Extractive algorithms for the WCEP dataset.

For the Extractive methods, luhn is the top-performing algorithm, achieving mean ROUGE-1, ROUGE-2, and ROUGE-L scores of 0.265, 0.061, and 0.215, respectively. For the Abstractive methods, the glove algorithm delivers the best results, with mean scores of 0.242, 0.049, and 0.195 for ROUGE-1, ROUGE-2, and ROUGE-L, respectively. Overall, the average results for both methods are quite similar, with lexRank emerging as the best algorithm. All algorithms score very close to each other. In this case, luhn is the best-performing algorithm, scoring about 10% above the average.

Initially, it was hypothesized that EATS methods would significantly outperform AATS approaches. However, the results discredited this assumption, revealing that both methods generated similar outcomes in most cases. Consequently, we can infer that ROUGE metric does not enable us to distinguish between AATS and EATS approaches so suggesting its use for evaluating the absolute quality of a summary independently on the specific approach used.

5. Results analysis for RQ2

The second research question aimed to compare the impacts of a single and multiple summary executions of TS algorithms. The ROUGE score average on both a summary and the total of compared summaries is computed for each block of summaries. Multiple executions were contemplated in two ways:

1. **Extractive** algorithms on Abstractive input;
2. **Abstractive** algorithms on Extractive input.

Table 7 shows the results of the Extractive algorithms performed on the output of an Abstractive one and vice versa. The first two values in each row of the tables are the percentages with which the approach (single or multiple) has achieved better results on the total number of samples, and the last two values are the average scores for each of the two methodologies.

Table 7
Comparison between a single and multiple executions of Extractive (resp Abstractive) algorithms on the input of Abstractive (resp Extractive) ones.

	Performances		Mean	
Extractive Algorithm	Single execution	Multi-exec (on Abs)	Single execution	Multi-exec (on Abs)
textRank	34.88%	65.13%	0.2100	0.2396
lsa	43.95%	56.05%	0.2235	0.2399
luhn	41.65%	58.35%	0.2241	0.2396
lexRank	52.50%	47.50%	0.2566	0.2424
Abstractive Algorithm	Single execution	Multi-exec (on Ext)	Single execution	Multi-exec (on Ext)
word2vec	39.00%	61.00%	0.2145	0.2391
doc2vec	41.16%	58.84%	0.2177	0.2388
glove	38.60%	61.40%	0.2133	0.2389

In this case, a t-test is also used to establish statistical validity in both types of experiments. The distinctions between the EATS and AATS techniques were considered when conducting this test. The numerous execution techniques of a summary are contrasted with the single execution strategy for each experiment. The population consists of 1000 paired summaries.

The results are quite dissimilar: the t-test for the Extractive approach produced a p-value of 0.4, indicating that this experiment has no statistical validity. Instead, the t-test of the Abstractive method gives a p-value of 0.018, which is less than the required 0.05 for statistical validity, confirming the alternative hypothesis that multiple executions surpass single executions. These findings are remarkable, as they reveal that the multiple execution approach performed better than the single execution method in almost all algorithms. The compression ratio derived from multiple algorithm runs could be one explanation: a first iteration can remove redundant information, whilst a second compresses important concepts into a higher-scoring summary. This indicates how the compression ratio has caused algorithms to save as much information from the source text as possible to include it in the output summary. Because the ratio of n-grams overlapping, especially if properly selected across two algorithms, might lead to misleading results, having a more compressed reference summary can benefit from the ROUGE score. We must realize that, whilst the ROUGE score is excellent for multiple executions, the summary readability must also be considered. As a result, the alternative hypothesis was confirmed in this experiment. The multiple execution of algorithms almost always outperforms the single ones.

6. Validity evaluation and threats discussion

The validity of the results of an experiment can be compromised by various types of threats: Conclusion, Internal, Construct, and External Validity. This section shows these threats for RQ1 (extendable to RQ2).

6.1. Conclusion validity

The threat of having *low statistical power* was excluded. All the experiments have been completed, and the collected results are based on solid scientific and statistical security. Furthermore, the metric used to compare the two methods performed in the experiment returns a well-defined numerical score, which is comparable and can be analyzed without losing validity.

The *violation hypothesis of statistical tests* is a problem that can affect the dataset used. In fact, there may be fluctuations in each computed score of an algorithm, depending on multiple factors, including the syntactic and semantics of the input text. This is reduced by running many tests for each algorithm and averaging the result of each block (1000 randomly chosen summaries).

For the work done, the amount of performed tests is only for statistical purposes, so *Fishing* is excluded.

As for the *reliability of the measurements*, there is no doubt about the correctness of the obtained scores. All ROUGE metrics are computed using a standard Python library, and since it is a well-defined algorithm, the results are reproducible.

For the *reliability of treatment implementation*, it was chosen a standard execution methodology, in combination with the subject selection and the design type, that avoids an incorrect execution and then this type of threat.

Random irrelevances in the experimental setting are not to be considered because the execution is done in a controlled environment without the possibility of any interference from external phenomena. The *random heterogeneity of subjects* is also attenuated by averaging the result of the execution over 1000 texts.

6.2. Internal validity

Historical threats are avoided because there is no risk that, by running the same experiment with the same algorithm in a different time interval, the result will change.

The same can be said for the *maturation threats* because the algorithms do not store information over time and in the various executions of the experiment, so when time passes, results will always be the same.

The equal assumption for the *testing threats*, since whilst a human may give different answers during the test due to some knowledge about the procedure, an algorithm does not have this problem, so results are consistent across all tests.

Instrumentation threats can lead to some issues. In particular, external libraries that contain the algorithms used during the experiment can have different types of problems, such as bugs or errors during the implementation. In addition, the developed software may also have errors, especially in the implementation phase, which may include both the computation of summaries and the Rouge score storage. To mitigate this type of issue, a deep study was conducted on the package documentation of the software used. Firstly, thorough code reviews and rigorous testing of external libraries and internally developed software were conducted. Secondly, maintaining version control and dependency management ensured consistency and traceability of software components. In addition, continuously benchmarking algorithm performance against established standards provided by the developers aids in detecting deviations or issues promptly, facilitating timely adjustments. So, each algorithm's average performance was compared with the average standard quality, and this average performance was the average of all the average results on each considered sample used in the experiments. So the result is as general as possible.

During the validity evaluation phase, factors such as statistical regression, selection biases, and mortality threats associated with human behaviors are not considered. These elements can significantly impact the outcomes but are often overlooked in the assessment process.

6.3. Construct validity

Construct validity concerns the generalization of the experiment result to the concept or theory behind it. The *inadequate preoperative explanation of constructs* refers to the possibility that the construct may not be well defined. For example, saying "one is better than the other" can have many meanings because "better" is poorly defined. In this case, the used metric allows a perfect mathematical comparison between numerical values.

The *mono-method bias* implies that the use of a single type of measurement or observations gives back the risk that if this measurement provides a measurement bias[5] the experiment will be misleading. A

[5] "Measurement bias" refers to any systematic or non-random error that occurs in a study data collection. Another generic term for this type of bias is "detection bias".

solution could be using different types of measures to cross-check between them. But, as we are evaluating a measure for the quality of the text summaries, it is impossible to use a second metric.

Regarding *Confounding constructs and Construct Levels*, sometimes the problems are not primarily the presence or absence of the construct but the levels that the construct assumes. In the presented experiment, considering the first research question, this is addressed using different TS algorithms for each method to have a reliable statistical meaning. Each algorithm corresponds to the construct levels. Finally, all algorithms are averaged to compare the two measures to obtain the final score. A similar approach was also conducted for the second research question. In fact, for each of the examination methodologies (multiple execution and single execution), different algorithms will be used to analyze each portion of the text.

Interaction of different treatments and between tests and treatments are threats closely related to human behavior in the experiment. This is impossible for these experiment tests.

6.4. External validity

External validity threats limit the ability to generalize the experiment results. The *interaction of selections and treatments* refers to the effect of having a non-representative sample of the population to generalize. This type of threat is very important for this work because the dataset used refers to a generic text that needs to be summarized. If the experiment's goal is more focused on a single topic, for example in the medical field, the results can be very different. This depends on the algorithms used and the different sets of words used in the training phase. But this is an open question, too. The aim to be achieved in this case is to have results on standard datasets to compare literature-based data.

The *interaction of settings and treatments* refers to not having an available representative experimental environment or material. For the proposed research, this may be related to the algorithms and datasets used for tests. The algorithms used are imperfect because the available computing power is insufficient to perform experiments with more complex algorithms. For this reason, different algorithms were used to be able to have an average of their performances, to generalize the result as much as possible, allowing other researchers to perform the experiment in the same environment using one of the most used data sets to evaluate the performance of TS algorithms in the literature.

From the point of view of *interaction of history and treatment*, the only threat that can affect the experiment results is the release of new and more powerful TS methods or updated versions of the used datasets.

7. Conclusions and future work

The primary objective of this study is to evaluate the ROUGE metric for TS algorithms and then determine whether a single execution of an algorithm produces better results than a multiple one. We conclude from our research that ROUGE shows similar results when applied to AATS and EATS and then that a multiple execution yields better results than a single execution (also when evaluated by ROUGE). Our findings highlight the importance of ROUGE to function as an independent and unbiased metric across different summarization approaches.

In future studies, we aim to expand our analysis to include other evaluation metrics, as reported in [27], and lesser-known algorithms, exploring their effectiveness in summarization tasks. Our goal is to identify new evaluation methods that move beyond traditional statistical metrics, potentially leveraging advanced NLP algorithms for deeper text comprehension and evaluation of summary quality.

Most existing studies employ uni-modal evaluation techniques such as ROUGE scores. Multi-modal evaluation metrics in summarization techniques could allow for a thorough assessment of text-based summaries, considering factors beyond traditional uni-modal metrics like ROUGE. This includes aspects such as coherence, informativeness, and relevance, which are crucial for assessing the quality of textual summaries. They could serve as a benchmarking tool for evaluating different text summarization models and techniques.

Additionally, we plan to investigate the impact of focusing algorithms specifically on single-topic summaries. By training these algorithms on datasets from specific limited interest fields, we anticipate producing summaries that are more accurate and more engaging to readers. This approach aims to enhance the relevance and applicability of summarization techniques in specialized domains.

8. Open issues & challenges

In the field of NLP, TS methods, such as AATS and EATS, have achieved great results but together with these results we are also trying to improve the automatic evaluation of the summaries. The challenges and unresolved issues surrounding the evaluation of text summarization techniques, despite using ROUGE which is a widely used metric in this scenario, are diverse and intricate. There is a growing need for evaluation metrics that incorporate semantic understanding, beyond surface-level lexical matching, to fully capture the quality of a summary in terms of coherence and informativeness. In addition, the majority of current research relies on unimodal assessment techniques. A significant gap exists in multimodal evaluation approaches that could provide a comprehensive assessment of summary quality. Multimodal metrics would encompass factors beyond conventional measures, including coherence, informativeness, and relevance, which are crucial for thorough evaluation of textual summaries. Furthermore, the problem of developing algorithms trained on domain-specific datasets is essential to generate more precise and relevant summaries in fields such as medicine, law, and technical disciplines, and consequently also, the evaluation metrics should adapt to the specificity of the context. An additional hard challenge is the computational complexity involved. Advanced evaluation methods like Pyramid and SSAS offer a finer understanding of summary quality by considering semantic relationships and content units. However, these methods require substantial computational resources. In summary, while the ROUGE metric offers a standardized approach to evaluate textual summaries, its limitations underscore the necessity for more sophisticated, multifaceted evaluation methods that account for semantic meaning, domain specificity, and computational feasibility. Addressing these challenges and unresolved issues is paramount to advancing the field of TS and to identify beyond traditional statistical metrics.

CRediT authorship contribution statement

Alessia Auriemma Citarella: Writing – review & editing. **Marcello Barbella:** Writing – review & editing. **Madalina G. Ciobanu:** Writing – review & editing. **Fabiola De Marco:** Writing – review & editing. **Luigi Di Biasi:** Writing – review & editing. **Genoveffa Tortora:** Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Genoveffa TORTORA reports financial support was provided by European Union Next-GenerationEU. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors sincerely wish to remember Michele Risi, who actively contributed to this work but passed away at a young age.

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013).

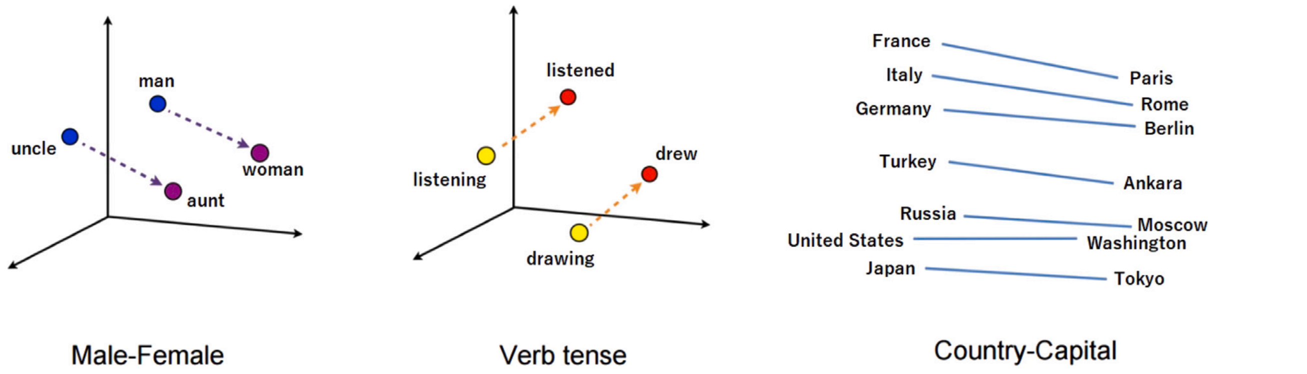


Fig. A.14. Multidimensional representation for word embedding.

Appendix A

A.1. Text representation

In the process of text summarization, the pre-processing phase is crucial since it forms the foundation for the ultimate result that a summary will produce. Pre-processing involves cleaning and preparing the text data, such as removing stop words, stemming, and tokenization [28]. Once the text is pre-processed, it can then be represented using, several techniques, such as vectors or matrices to capture its features for summarization [29].

Some of the most widely used approaches to represent text information are mentioned in [30]. The following are the most commonly used vectorial representations:

(a) **Bag of Words** This technique makes it possible to describe the presence of terms within a document accurately. It is called "*bag*" because all information referring to a word's position inside the text is no longer considered. It focuses on determining whether or not a word appears in the analyzed text. This entails having a vocabulary or set of known words, expressed by a list of clear terms and a metric for their presence. This metric is frequently the number of times the words appear in a vector. Each place corresponds to a single word, and the number in a given location is the total number of occurrences of that word in the text. This type of vectorial representation is called "*sparse vector*".

(b) **Word Embedding** Word Embedding is another approach for representing words in a multi-dimensional space (see Fig. A.14). It enables the representation of words with similar meanings in the same way. This method is based on a data structure known as a "*dense vector*", which is extremely computationally efficient where each word is represented by a vector of real numbers (usually of tens or hundreds of dimensions). A learning phase is required to obtain this representation for each word. Several algorithms are used to achieve this scope, and the literature always proposes new ideas.

A.2. Text similarity

Different features and metrics can be used to calculate the similarity of sentences. Both statistical and linguistic computations are utilized to extract the most commonly used elements from the text. Several measures are used: term frequency, inverse document frequency, sentence position, sentence length, cue words, verbs, and nouns, pos (*part of speech*) and ner (*named entity recognition*) tagging, and so on.

We must explore a metric for similarity measures that allows us to efficiently calculate the coverage of key information while removing the duplicates. Cosine similarity, Euclidean, Manhattan, and Jaccard distances are the most frequently used measures in the TS field.

Appendix B. Recent in TS algorithms

Automatic text summarization is a fundamental NLP task that employs extractive, abstractive, and hybrid methods, depending on how these approaches are combined [2]. Following the synthesis process, text classification categorizes the content into predefined groups or labels. This step involves analyzing the synthesized text and assigning it to specific categories based on its content, which helps in organizing and structuring the information for further use or analysis [31].

Several review studies have explored text summarization methods. Using the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) framework, the authors in [1] have selected articles between 2000 and 2023 that address text synthesis techniques, evaluations, and challenges [3]. The authors presented an inclusive review of extractive and abstractive summarization techniques for various inputs. Their study showed a comparative study of different models, classified based on the selected techniques. The review included the parametric evaluation of these techniques and their challenges were also presented. Another review study shows an in-depth survey of the recent AATS, EATS and hybrid techniques and open challenges [32]. This study categorizes summarization methods into extractive, abstractive, and hybrid techniques. For this technique, the study evaluates common metrics like ROUGE, METEOR, and CIDEr, highlighting their limitations in accurately assessing summary quality. The review aimed to highlight issues and propose a research direction focused on overcoming these limitations to enhance text summarization's efficiency, accuracy, and scalability.

In the following, we will explore the most common techniques for the two types of TS approaches, *Extractive* and *Abstractive*, trying to understand the best way to evaluate a system-generated summary. As proved in [33], human evaluation of the quality of a summary is subjective since it depends on individual criteria of relevance, comprehensibility, and readability. Therefore, the metrics used for this purpose mainly offer a statistical approach to evaluation, comparing the overlapping words from both the summary and the comparison text, forgetting to evaluate the semantic meaning of what the text offers [34,35].

B.1. Extractive method

Extractive text summarization methods involve extracting sentences from a source document based on various keywords and features. These features can be derived using statistical, semantic, and graphical methods [36]. In addition, recent research in AI, specifically in Deep Learning (DL) [37], has consolidated unique and more sophisticated EATS approaches. This paper uses some of the most promising ones in the literature, including those based on Neural Networks, Graphs, Fuzzy Logic, and Semantic methods, which are detailed below.

B.1.1. Neural network approaches

Neural Networks (NN) are commonly used to generate complex text input features. In [38], an overview of today's most widely used algorithms is provided, highlighting that the approaches are classified into three categories based on DL techniques, which are Restricted Boltzmann Machine, Variation Auto-Encoder, and Recurrent Neural Network.

The *Restricted Boltzmann Machines* (RBMs) are NN composed of an input layer and a hidden layer, where connections are made only between neurons of different layers [5]. Preeja et al. [39] explores various recent DL models that have been applied to the summarization and generation of text, such as RBM, Generative Adversarial Networks (GAN), Variational Auto Encoders (VAE), and Recurrent Neural Networks (RNN). Also, the authors in [40] propose a DL-based approach for multi-document text summarization. A matrix containing this information is provided to a neural network following the first step of normalization and feature extraction. The network type is a Deep Believe Network comprising various RBMs.

The *Variation Auto-Encoder* for a TS task is based on a NN composed of an encoder, a decoder, and a loss function. The encoder and decoder are two neural networks where the encoder's output is the decoder's input, whilst the decoder's output is a probability distribution [41]. The features from the text are extracted during the pre-processing step. Statistical methods or the count of the most frequently used terms are usually used, and a matrix comprising these data is then constructed. Each text is semantically analyzed, allowing a vector of characteristics to be created as input for the training phase. The sentences with the highest cosine similarity are extracted in the last stage [42].

The *Recurrent Neural Networks* (RNNs) are composed of a series of hidden layers, each of which receives a sequence of words as input, with the summary words becoming the output [43]. In [44], an RNN is presented that can perform TS on a single document. The suggested network is based on an encoder and an extractor model. The first uses Long Short-Term Memory (LSTM) cells to create the features. The extractor builds a weighted representation of each sentence in the input document to select summary sentences with greater accuracy and more correlations. The selection task is completed at the end of the network training phase. The authors of the study [45] propose an extractive synthesis model trained employing employed the Model-Agnostic Meta-Learning (MAML) algorithm on Bidirectional Long Short-Term Memory (BiLSTM) and subsequently evaluate with ROUGE. The use of meta-learning improves performance even if the model has some limitations such as its inability to capture correlations between words in the text. In addition, to improve the interpretability of their model, they created an extensive framework that integrates multiple techniques such as SHAP (SHapley Additive exPlanations), linear regression, decision trees, and input modification.

In [46], the summarization technique employs a DNN-based learning model that determines which sentences should be included in a summary by evaluating both their saliency and the overall diversity of the summary. This approach ensures that the summary is not only representative of the most important content but also varied in its coverage. To achieve this, the model is trained using two distinct sets of features: saliency features, which capture the importance of each sentence, and diversity features, which promote a well-rounded and diverse summary. This dual-feature approach helps create summaries that are both comprehensive and engaging.

B.1.2. Fuzzy logic based approaches

Defuzzifier, fuzzifier, fuzzy knowledge base, and inference engine are the four fundamental components of the TS fuzzy logic approach [47]. The fuzzy logic technique feeds sentence length, similarity, and other textual features into the fuzzy system.

The authors in [48] present and implement key summarization techniques for multi-document summarization. Their summarization process incorporates rule-based fuzzy logic for scoring sentences. After scoring all sentences, they are ordered in descending sequence according to their scores derived from the fuzzy inference system. A major challenge addressed in multi-document summarization is the elimination of redundant information. To tackle this, cosine similarity measures are employed to filter out sentences with overlapping content from the extracted salient sentences, thereby refining the final summary. The proposed approach was evaluated using the DUC 2004 dataset with ROUGE 2 and ROUGE 4 metrics, demonstrating superior performance compared to other systems. In [49], the authors propose a model for extractive text summarization based on the fuzzy logic, specifically employing a triangular membership function. The sentences with the highest scores are chosen and incorporated into the summary. The authors [50] propose an application of Fuzzy Bi-GRU to remove similarity or redundancy from extracted sentences, generating an abstract summary. From comparison with other models, this proposed model outperforms in terms of ROUGE-N and L scores. Patel et al. [48] developed a statistical feature-based model that uses a fuzzy model to deal with feature weights' imprecise and uncertain nature. Redundancy removal using cosine similarity is presented as an additional enrichment, and the testing findings show that this methodology outperforms the other summarizers significantly. Instead, in the work of Sharaff et al. [49] the summary sentences are selected based on several criteria, such as the frequency of the terms and their position in the text, enhancing the quality of the produced summaries of all lengths.

However, in the literature we could see that the use of fuzzy logic is employed for several techniques and tasks related to text mining. The articles presents various methodologies, various research topics for text mining, feature extraction methods and related application fields [51].

B.1.3. Graph based approaches

A TS technique based on graphs is proposed in [30]. PageRank is a well-known TS model that uses a graph approach. It is built on *Google's Hits* algorithms [52]. The two more relevant ideas in the literature are using the graph as a semantic network between phrases [53] or as input for a convolutional network [54]. These approaches have been applied to carry out various studies such as extractive synthesis in the healthcare field [55]. The authors in [56] used machine learning techniques such as k-means, TextRank and Latent Semantic Analysis to identify and extract important information to produce the summarized texts. The methods were subsequently evaluated using the ROUGE-1, ROUGE-2, and ROUGE-N metrics.

B.1.4. Semantic approaches

The most frequently used TS models rely on statistical methodologies that do not account for the semantic or contextual meaning of the text being analyzed. On the other hand, sentences in a document are well clustered according to Semantic Role Labeling, making it easier to build groups of similar elements [57]. The authors in [58] employed Semantic Analysis to represent sentences, yielding encouraging summaries. Moreover, it is proven in [59] that semantic knowledge of the text is extremely important in the AATS technique.

B.2. Abstractive method

Abstractive summarization is a technique in which the summary is generated by creating novel sentences through rephrasing or using new words, rather than merely extracting important sentences from the source text [60]. Following the widespread adoption of DL methodologies, researchers have developed a solid foundation for abstractive algorithms, which are no longer linked to the traditional approaches to NLP. DL models, such as those based on seq2seq and attention-model, have elevated the research of AATS to a new level, sometimes outperforming extractive approaches [61]. For example, in the medical field, documents often do not have abstracts or summaries. Some researchers have faced the challenge of summarizing information from clinical documents using different abstract synthesis techniques (T5, BART and PEGASUS). Performance was then compared using the ROUGE metric [62].

B.2.1. Seq2seq model

The encoder–decoder design, in which the lengths of the input and output sequences differ, is used in the Seq2seq neural network model. For a TS task, the encoder examines the whole input sequence to generate a vector of features [63]. In general, specific types of NN are used as internal components for the encoder and decoder in the literature. We can see the use of RNNs by Gated Recurrent Unit (GRU) or LSTM [24]. LSTM is the most popular since it allows for determining long-term dependencies whilst avoiding the gradient problem. The Encoder–Decoder work can be divided into two phases: training and inference.

1. *Training phase*: here, the encoder and decoder will be trained to predict the target sequence, with a different time frame;
   - *Encoder*: this module uses LSTM to read the whole input sequence and inject a word into the encoder at any instant of time;
   - *Decoder*: this module is composed of LSTM cells too. It reads the whole sequence word by word and tries to predict the same sequence with an instant time difference.
2. *Inference phase*: the model is evaluated on some novel sequences for which the target sequence is known during this step. Briefly:

   (a) the encoder processes the entire sequence given in input, and the decoder is initialized with the internal state of the encoder;
   (b) the token start is sent as the first input of the decoder;
   (c) the decoder is begun for an instance of time at the time;
   (d) the output is the probability of the subsequent word;
   (e) the word with the highest probability is picked and pushed as input to the next instant of time;
   (f) meanwhile, the decoder's internal state is updated with new cell weights.

   Steps 3–5 are performed until the token end is read.

The importance of tokens is highlighted in the work [64] where they propose an advanced sentence embedding method based on a pre-trained language model. The token importance is calculated by combining the eXplainable module with a text summarization model, and the final sentence embedding is derived using weighted pooling.

B.2.2. Transformer network

A Transformer network is a deep learning model for sequential data, particularly effective in NLP. It uses a self-attention mechanism to capture long-range dependencies, overcoming limitations of RNNs. Key components include multi-head attention, positional encoding, and an encoder–decoder structure with feed-forward networks, layer normalization, and residual connections [65]. Transformers enable advanced NLP tasks like machine translation, text summarization, question answering, and text generation [66]. BERT (Bidirectional Encoder Representations from Transformers) is one of the most important transformers used in NLP [67]. Its model design is based on the implementation of [65] and consists of a bidirectional multilayer transformer encoder. Unlike recent language representation models, it aims to pre-train deep bidirectional representations from the unlabeled text by conditioning on both the left and right context in all layers. As a result, with just one additional output layer, the pre-trained BERT model may be fine-tuned to produce state-of-the-art models for several tasks, such as question answering and language inference, without requiring large task-specific architecture changes. BERT is a cutting-edge NLP model with great power. The BERT model provides an efficient response but still has limitations and drawbacks [68]. Despite its limitations, it remains a model that has allowed us to carry out an interesting study on the selection of articles on the Covid topic, using the CORD-19 dataset. The results were evaluated by applying the rouge metrics on recall, precision and score [69]. Given its limitations, the authors [70] conducted a study on a sparse attention mechanism called BIGBIRD applied to text summarization of long documents. The AFBB-LDTS they proposed achieves better ROUGE and BLEU scores than state-of-the-art systems and is also less complex.

Data availability

No data was used for the research described in the article.

References

[1] Supriyono, A. Wibawa, Suyono, K. Fachrul, A survey of text summarization: Techniques, evaluation and challenges, Nat. Lang. Process. J. 7 (2024) 100070, http://dx.doi.org/10.1016/j.nlp.2024.100070.

[2] W.S. El-Kassas, C.R. Salama, A.A. Rafea, H.K. Mohamed, Automatic text summarization: A comprehensive survey, Expert. Syst. Appl. 165 (2021) 113679.

[3] G. Mishra, N. Sethi, A. Loganathan, Inclusive review on extractive and abstractive text summarization: Taxonomy, datasets, techniques and challenges, in: Intelligent Systems Design and Applications, 2023, pp. 65–80.

[4] V. Dalal, L. Malik, A survey of extractive and abstractive text summarization techniques, in: 6th Inlt. Conf. on Emerging Trends in Eng. and Tech., IEEE, 2013, pp. 109–110.

[5] N. Zhang, S. Ding, J. Zhang, Y. Xue, An overview on restricted Boltzmann machines, Neurocomputing 275 (2018) 1186–1199.

[6] M. Akter, N. Bansal, S.K. Karmaker, Revisiting automatic evaluation of extractive summarization task: Can we do better than rouge? in: Findings of the Association for Computational Linguistics, ACL 2022, 2022, pp. 1547–1560.

[7] R. Akula, I. Garibay, Sentence pair embeddings based evaluation metric for abstractive and extractive summarization, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, 2022, pp. 6009–6017.

[8] W. Lin, S. Li, C. Zhang, B. Ji, J. Yu, J. Ma, Z. Yi, SummScore: A comprehensive evaluation metric for summary quality based on cross-encoder, in: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, Springer, 2022, pp. 69–84.

[9] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, 2004, pp. 74–81.

[10] M. Zhang, C. Li, M. Wan, X. Zhang, Q. Zhao, ROUGE-SEM: Better evaluation of summarization using ROUGE combined with semantics, Expert. Syst. Appl. 237 (2023) 121364.

[11] E. Lloret, L. Plaza, A. Aker, The challenging task of summary evaluation: an overview, Lang. Resour. Eval. 52 (2018) 101–148.

[12] E. Davoodijam, M. Alambardar Meybodi, Evaluation metrics on text summarization: comprehensive survey, Knowl. Inf. Syst. (2024) 1–22.

[13] M. Barbella, M. Risi, G. Tortora, A.A. Citarella, Different metrics results in text summarization approaches, in: DATA, 2022, pp. 31–39.

[14] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv preprint arXiv:1301.3781.

[15] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, PMLR, 2014, pp. 1188–1196.

[16] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1532–1543.

[17] R. Mihalcea, P. Tarau, Textrank: Bringing order into text, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004, pp. 404–411.

[18] Y. Gong, X. Liu, Generic text summarization using relevance measure and latent semantic analysis, in: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2001, pp. 19–25.

[19] H.P. Luhn, The automatic creation of literature abstracts, IBM J. Res. Dev. 2 (2) (1958) 159–165.

[20] G. Erkan, D.R. Radev, Lexrank: Graph-based lexical centrality as salience in text summarization, J. Artif. Intell. Res. 22 (2004) 457–479.

[21] A. Nenkova, R.J. Passonneau, Evaluating content selection in summarization: The pyramid method, in: Human Lang. Tech. Conf. of the North American Ch. of the Assoc. for Comput. Ling., HLT-NAACL, 2004, pp. 145–152.

[22] D. Harman, P. Over, The effects of human variation in duc summarization evaluation, in: Text Summarization Branches Out, 2004, pp. 10–17.

[23] F. Liu, Y. Liu, Exploring correlation between ROUGE and human evaluation on meeting summaries, IEEE Trans. Audio, Speech, Lang. Process. 18 (1) (2009) 187–196.

[24] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al., Abstractive text summarization using sequence-to-sequence rnns and beyond, in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, 2016, pp. 280–290.

[25] D. Greene, P. Cunningham, Practical solutions to the problem of diagonal dominance in kernel document clustering, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 377–384.

[26] D. Gholipour Ghalandari, C. Hokamp, N.T. Pham, J. Glover, G. Ifrim, A large-scale multi-document summarization dataset from the wikipedia current events portal, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 1302–1308.

[27] G. Mishra, N. Sethi, A. Loganathan, Y.-H. Lin, Y.-C. Hu, Attention free BIGBIRD transformer for long document text summarization, Int. J. Comput. Inf. Syst. Ind. Manag. Appl. 16 (2) (2024) 20–20.

[28] A. Kathuria, A. Gupta, R. Singla, A review of tools and techniques for preprocessing of textual data, in: Computational Methods and Data Engineering: Proceedings of ICMDE 2020, vol. 1, Springer, 2021, pp. 407–422.

[29] R. Patil, S. Boit, V. Gudivada, J. Nandigam, A survey of text representation and embedding techniques in nlp, IEEE Access 11 (2023) 36120–36146.

[30] P. Janjanam, C.P. Reddy, Text summarization: An essential study, in: Intl. Conf. on Computational Intelligence in Data Science, ICCIDS, IEEE, 2019, pp. 1–6.

[31] N. Pandey, S. Kumar, V. Ranjan, M. Ahamed, A.K. Sahoo, Analyzing extractive text summarization techniques and classification algorithms: A comparative study, in: 2024 International Conference on Advancements in Smart, Secure and Intelligent Computing, ASSIC, 2024, pp. 1–5.

[32] S.D. Myla, E.R. Saini, E.N. Kapoor, Auto text summarization in natural language processing: Review, in: 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things, IDCIoT, 2024, pp. 1258–1267.

[33] P.C.F. de Oliveira, How to Evaluate the 'Goodness' of Summaries Automatically (Ph.D. thesis), University of Surrey, 2005.

[34] T. Stanisz, S. Drożdż, J. Kwapień, Complex systems approach to natural language, Phys. Rep. 1053 (2024) 1–84.

[35] R. Mao, K. He, X. Zhang, G. Chen, J. Ni, Z. Yang, E. Cambria, A survey on semantic processing techniques, Inf. Fusion 101 (2024) 101988.

[36] G. Sharma, D.H. Sharma, Automatic text summarization methods: A comprehensive review, SN Comput. Sci. 4 (1) (2023) 33, http://dx.doi.org/10.1007/S42979-022-01446-W.

[37] G. Mishra, N. Sethi, L. Agilandeeswari, Two phase ensemble learning based extractive summarization for short documents, in: Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition, SoCPaR 2022, Springer, 2023, pp. 129–142.

[38] D. Suleiman, A.A. Awajan, Deep learning based extractive text summarization: Approaches, datasets and evaluation measures, in: 6th Intll. Conf. on Social Networks Analysis, Manag. and Sec., SNAMS, IEEE, 2019, pp. 204–210.

[39] P. V, T. Subashini, J. J. S, Exploring deep learning approaches for text summarization and text generation, in: 2023 Third International Conference on Ubiquitous Computing and Intelligent Information Systems, ICUIS, 2023, pp. 154–160.

[40] A. Rezaei, S. Dami, P. Daneshjoo, Multi-document extractive text summarization via deep learning approach, in: 5th Conf. on Knowledge Based Engineering and Innovation, KBEI, IEEE, 2019, pp. 680–685.

[41] M. Yousefi-Azar, L. Hamey, Text summarization using unsupervised deep learning, Expert Syst. Appl. 68 (2017) 93–105.

[42] B. Ghojogh, M. Crowley, F. Karray, A. Ghodsi, Variational autoencoders, in: Elements of Dimensionality Reduction and Manifold Learning, Springer International Publishing, 2023, pp. 563–576, Ch. 20.

[43] A.C. Tsoi, Recurrent neural network architectures: an overview, in: International School on Neural Networks, Initiated by IIASS and EMFCSC, Springer, 1997, pp. 1–26.

[44] L. Chen, M. Le Nguyen, Sentence selective neural extractive summarization with reinforcement learning, in: 11th Intl. Conf. on Knowl. and Sys. Eng., KSE, IEEE, 2019, pp. 1–5.

[45] S.-N. Vo, T.-T. Vo, B. Le, Interpretable extractive text summarization with meta-learning and BI-LSTM: A study of meta learning and explainability techniques, Expert. Syst. Appl. 245 (2023) 123045, http://dx.doi.org/10.1016/j.eswa.2023.123045.

[46] S. Chowdhury, K. Sarkar, A new method for extractive text summarization using neural networks, SN Comput. Sci. 4 (2023) http://dx.doi.org/10.1007/s42979-023-01806-0.

[47] R. Czabanski, M. Jezewski, J. Leski, Introduction to fuzzy systems, in: Theory and Applications of Ordered Fuzzy Numbers: a Tribute to Professor Witold Kosiński, Springer International Publishing, 2017, pp. 23–43.

[48] D. Patel, S. Shah, H. Chhinkaniwala, Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique, Expert Syst. Appl. 134 (2019) 167–177.

[49] A. Sharaff, A.S. Khaire, D. Sharma, Analysing fuzzy based approach for extractive text summarization, in: 2019 International Conference on Intelligent Computing and Control Systems, ICCS, 2019, pp. 906–910, http://dx.doi.org/10.1109/ICCS45141.2019.9065722.

[50] G. Mishra, N. Sethi, L. Agilandeeswari, Fuzzy bi-GRU based hybrid extractive and abstractive text summarization for long multi-documents, in: Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition, SoCPaR 2022, Springer, 2023, pp. 153–166.

[51] Y.-W. Lai, M.-Y. Chen, Review of survey research in fuzzy approach for text mining, IEEE Access PP (2023) http://dx.doi.org/10.1109/ACCESS.2023.3268165, 1–1.

[52] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web, Technical Report 1999–66, Stanford InfoLab, 1999, URL http://ilpubs.stanford.edu:8090/422/.

[53] X. Han, T. Lv, Z. Hu, X. Wang, C. Wang, Text summarization using FrameNet-based semantic graph model, Sci. Prog. 2016 (2016).

[54] Z. Jalil, J.A. Nasir, M. Nasir, Extractive multi-document summarization: A review of progress in the last decade, IEEE Access 9 (2021) 130928–130946, http://dx.doi.org/10.1109/ACCESS.2021.3112496.

[55] K. Ramani, K. Bhavana, A. Akshaya, K.S. Harshita, C.R. Thoran Kumar, M. Srikanth, An explorative study on extractive text summarization through k-means, LSA, and TextRank, in: 2023 International Conference on Wireless Communications Signal Processing and Networking, WiSPNET, 2023, pp. 1–6.

[56] W. Liu, Y. Sun, B. Yu, H. Wang, Q. Peng, M. Hou, H. Guo, H. Wang, C. Liu, Automatic text summarization method based on improved TextRank algorithm and K-means clustering, Knowl.- Based Syst. 287 (2024) 111447, http://dx.doi.org/10.1016/j.knosys.2024.111447.

[57] B. Mutlu, E.A. Sezer, Enhanced sentence representation for extractive text summarization: Investigating the syntactic and semantic features and their contribution to sentence scoring, Expert. Syst. Appl. 227 (2023) 120302.

[58] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, N.A. Smith, Toward abstractive summarization using semantic representations, in: Proceedings of the 2015 Conference of the North American, Association for Computational Linguistics, 2018, pp. 1077–1086.

[59] A. Khan, N. Salim, H. Farman, M. Khan, B. Jan, A. Ahmad, I. Ahmed, A. Paul, Abstractive text summarization based on improved semantic graph approach, Int. J. Parallel Program. 46 (5) (2018) 992–1016.

[60] S. Gupta, S. Gupta, Abstractive summarization: An overview of the state of the art, Expert Syst. Appl. 121 (2019) 49–65.

[61] N. Giarelis, C. Mastrokostas, N. Karacapilidis, Abstractive vs. extractive summarization: An experimental review, Appl. Sci. 13 (13) (2023) 7620.

[62] E. Lalitha, K. Ramani, D. Shahida, E.V.S. Deepak, M.H. Bindu, D. Shaikshavali, Text summarization of medical documents using abstractive techniques, in: 2023 2nd International Conference on Applied Artificial Intelligence and Computing, ICAAIC, 2023, pp. 939–943.

[63] F. Wu, K. Kim, S. Watanabe, K.J. Han, R. McDonald, K.Q. Weinberger, Y. Artzi, Wav2seq: Pre-training speech-to-text encoder-decoder models using pseudo languages, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2023, pp. 1–5.

[64] Y. Cha, Y. Lee, Advanced sentence-embedding method considering token importance based on explainable artificial intelligence and text summarization model, Neurocomputing 564 (2023) 126987, http://dx.doi.org/10.1016/j.neucom.2023.126987.

[65] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017) 5998–6008.

[66] N. Patwardhan, S. Marrone, C. Sansone, Transformers in the real world: A survey on nlp applications, Inf. 14 (4) (2023) 242.

[67] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: North American Chapter of the Association for Computational Linguistics, 2019, pp. 4171–4186.

[68] A. Kumar, BERT: Extractive text summarization, in: 2023 5th International Conference on Advances in Computing, Communication Control and Networking, ICAC3N, 2023, pp. 1060–1066, http://dx.doi.org/10.1109/ICAC3N60023.2023.10541662.

[69] N. Darapaneni, R. Prajeesh, P. Dutta, V.K. Pillai, A. Karak, A.R. Paduri, Abstractive text summarization using BERT and GPT-2 models, in: 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication, IConSCEPT, 2023, pp. 1–6.

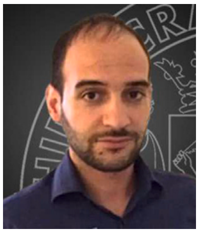
[70] G. Mishra, N. Sethi, A. Loganathan, Y.-H. Lin, Y.-C. Hu, Attention free BIGBIRD transformer for long document text summarization, Int. J. Comput. Inf. Syst. Ind. Manag. Appl. 16 (2) (2024) 20.

Alessia Auriemma Citarella graduated cum laude in Biology at the University of Salerno in July 2017. After the graduation, she was the winner of a scholarship from the University of Salerno with research activities within the project Methodologies and techniques of Visual Analytics and Data Warehousing applied to life sciences. In April 2022, she got Ph.D. in Computer Science at the same University and actually she is an Assistant Professor at UNISA as part of the project "D3 4 Health Digital Driven Diagnostics, prognostics and Therapeutics for Sustainable Health Care". In particular, her research interests are oriented toward the detection of melanoma through Machine Learning and Deep Learning techniques, with the help of biological skills in the detection phase of characteristic patterns of this pathology. Her field of application concerns Pattern Recognition, the analysis of proteomic and genomic Big Data, Information Visualization and modeling by Game Theory of systems responsible for the epidemiological trend of biomedical data.

Marcello Barbella has a Ph.D. in Computer Science from the University of Salerno, specializing in Data Analytics, Exploratory Data Analysis, Text Analytics, and Natural Language Processing (NLP). He has extensive experience in using statistical methods and advanced machine learning techniques to extract insights and facilitate decision-making. During his research career, Marcello Barbella applied AI solutions to solve real-world challenges. His work includes optimizing processes through datadriven strategies and developing scalable AI systems. Additionally, he has studied AI techniques such as neural networks, reinforcement learning, and deep learning, enhancing his ability to innovate and implement cutting-edge solutions across various fields.

Madalina Ciobanu is a Research Fellow at the Department of Computer Science at University of Salerno. She graduated in Computer Science at the University of Salerno and she holds a Ph.D. in Information Sciences and Technologies, Complex Systems and the Environment at the University of Salerno in May 2018. Her research interests are particularly oriented towards artificial intelligence techniques for the analysis of biomedical data, Computer Science, Android Operating System, Cybersecurity. The goal is to support the software engineer during the development and verification phases, proposing a semi-automatic method based on techniques code instrumentation and static model verification. Through modeling techniques checking, the activity aimed at detecting tampering or a cyber attack within a database, data warehouse and big data archives. She is very careful to promote awareness of the importance of the participation of girls and women in STEM disciplines. She is involved in Italian Team Staff of Italian Olympiads in Informatics since 2009.

Fabiola De Marco is an Assistant Professor at the Department of Computer Science at University of Salerno. She graduated cum laude in Computer Science at the University of Salerno in July 2020 and works on projects concerning biomedical data analysis. Her research interests are particularly oriented toward analyzing biomedical data through Machine Learning and Deep Learning techniques. Her field of application concerns Pattern Recognition, the analysis of biological Big Data, Information Visualization, and image analysis on different pathologies. By developing and implementing algorithms that can identify and interpret patterns within this data, she contributes to the development of predictive models and diagnostic tools with far-reaching implications in healthcare.



Luigi Di Biasi is an Assistant Professor in Computer Science at the University of Salerno. He obtained his Ph.D. in Computer Science on 03/04/2023 at the University of Salerno with an Excellent grade. Also, he obtained a Master's Degree in Computer Science from the University of Salerno on 03/28/2014, scoring 107/110. After obtaining the title, he won a Research Grant at the University of Salerno, Department of Pharmacy, relating to the project "Trusted Distributed Computing of Free Alignment Algorithms for Genomic Analysis", for which he carried out research from February 2015 to February 2016. In 2019, he participated in the establishment, as a founding partner, of the university spinoff SOFTMING SRL, active in digital health and drug analysis. He currently carries out research activities as part of the project "D3 4 Health - Digital Driven Diagnostics, prognostics and Therapeutics for Sustainable Health Care" at the Department of Computer Science, focusing his main activities on the problem of Diabet Detection and secondary activities on the Melanoma Detection on Clinical and dermoscopic Images, analysis of Electroencephalograms, Electrocardiograms, BigData and Learning Technologies, Text Mining and Digital Forensics. He has a background oriented towards bioinformatics and actively develops tools for molecular docking and the analysis of proteomic sequences. He is co-author of Sm-Covid-19, the first Italian app authorized for Contact Tracing and is the author of several articles published in scientific journals.

Genoveffa Tortora is a full professor of Computer Science (DI) since 1990, holder of the courses of Data Bases (Bachelor's Degree in Computer Science) and Data Bases II (Master's Degree in Computer Science) and IoT Data Analytics (Internet of Things Curriculum of the Master's Degree in Computer Science). She is the Context-Aware Intelligent Systems Laboratory (CAIS Lab) director at the Department of Informatics. She is the author or co-author of over 290 articles published in journals and proceedings of international conferences. Member of the editorial board of high-quality international scientific journals. Member of the Steering Committee of the IEEE Symposium on Visual Languages. Program Chair and member of the Program Committee of several international conferences. Senior Member of the IEEE Computer Society, member of the ACM Special Interest Group on Computer–Human Interaction. Member of the EATCS (European Association of Theoretical Computer Science) Member of the IAPR (International Association of Pattern Recognition). Reviewer for several international scientific journals. Member of the editorial board of various international journals. Since 2002 he has been in the MIUR Register of Experts and, as project reviewer, has held numerous positions by the MIUR and MISE ministries and by Regions.