ROUGE Metric Evaluation for Text Summarization Techniques

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

[a]*Department of Computer Science, University of Salerno, 84084, Fisciano (SA), Italy*

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 terms of how they work. The former extracts sentences from the document text directly, whereas the latter creates a summary by interpreting the text and rewriting sentences often with new words. So, it is important to assess and confirm how similar a summary is to the original text. The question is: *how can the quality of these methodologies of summaries be evaluated?* For the evaluation of text summarization results, various metrics and scores have been proposed in the literature, but the most used is ROUGE. Then, the primary goal of this study is to accurately estimate the behaviour of the ROUGE metric. We conducted a first experiment to compare the metric efficiency for evaluating Abstractive versus Extractive Text Summarization algorithms, 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. Our results have shown that: ROUGE does not obtain impressive results since it behaves in a similar way both on the Abstractive and Extractive algorithms; furthermore, in most cases, a multiple execution is better than a single one.

Keywords:
Automatic Text Summarization Algorithms, Extractive, Abstractive, ROUGE metric, Bert

1. Introduction

Today, there is a vast amount of textual data available from a variety of sources. In particular, extracting knowledge from long texts is becoming increasingly difficult for humans. The advancement of information technology, especially in the field of Artificial Intelligence (AI), has led to the formation of even more complex data management and processing tools. New algorithms for the analysis and extraction of the most important information from texts are constantly presented, even those created by humans.

These methodologies commonly referred to as Automatic Text Summarization, enable the production of summaries from any input text by combining their key concepts. There are two main groups of algorithms for extracting a summary from a text:

- *Extractive algorithms*: they select phrases from the input text, choosing those that best cover all the key information and discarding redundancy;
- *Abstractive algorithms*: they try to elaborate a new corpus, using different and more appropriate words and a different semantic composition, so as to output a simpler text.

Abstractive Automatic Text Summarization techniques (AATS) are far more fascinating than Extractive Automatic Text Summarization ones (EATS). In the literature, several techniques for both methodologies are suggested, which use both supervised and unsupervised algorithms [1]. The key issue is determining the quality of the summaries generated by these methods. Since it is difficult to compare one summary to another, is needed a metric that eases the comparison and is as unbiased as possible.

The most widely used subject evaluation metric is ROUGE. It is focused on the overlapping of n-grams (expressed as a numeric value) between the system and human summaries, without regard for their semantic

Email addresses: aauriemmacitarella@unisa.it (Alessia Auriemma Citarella), mbarbella@unisa.it (Marcello Barbella), mciobanu@unisa.it (Madalina G. Ciobanu), fdemarco@unisa.it (Fabiola De Marco), ldibiasi@unisa.it (Luigi Di Biasi), tortora@unisa.it (Genoveffa Tortora)

and syntactic accuracy.

The existing literature is more concerned with developing new summarization algorithms than with assessing the existing ones. Therefore this paper focuses on comparing different algorithms using a widely used dataset and a well-defined approach. The goal entails two separate research activities:

1. evaluating the performance of the ROUGE metric on the outputs of the Abstractive and Extractive algorithms;
2. evaluating the efficiency of its score on two alternative summarization methods.

EATS, by definition, uses sections of the original text to produce a summary, whereas AATS tends to introduce additional words, hence the former should do much better because there may be more overlapping of n-grams. To prove this idea, a first experiment was carried out. A second one, on the other hand, involves a comparison of two methodologies: the simple execution of a text summarization (TS) algorithm vs the multiple execution on the same text, in order to determine which methodology is better when the compression rate grows, using the specified metric. Preliminary results of this research are presented in [2].

The paper is structured as follows: the foundations for text representation, as well as similarity assessments, are introduced in Section 2, and the most recent technologies proposed by the literature on Abstractive and Extractive approaches are investigated in Section 3. Section 4 provides an overview of some of the most commonly used TS evaluation metrics, whilst Section 5 presents the experiment design and shows its organization, ensuring its replication. In order to generalize the results as much as possible, the first investigation involves several datasets including the CNN Daily-Mail[1], the most used dataset by researchers to test new summarization techniques. Because of the significant processing time, the second investigation, which is a follow-up, is only conducted on this dataset, since it is the most frequently used in text summarization studies. The experiment results are reported in Section 6, and the threats to validity are explored in Section 7. Finally, Section 8 summarizes the findings and offers some suggestions for further research. Table 1 lists all abbreviations used in the document for the convenience of the reader.

[1]CNN Daily-Mail: https://paperswithcode.com/dataset/cnn-daily-mail-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. Background

2.1. Text Representation

Some of the most widely used approaches to represent information of a text are mentioned in [3]. The following are the most commonly used vectorial representations:

a) **Bag of Words** This technique makes it possible to accurately describe the presence of terms within a document. It is called "*bag*" because all information referring to the position of a word inside the text is no longer taken into account. In fact, 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. 1). 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). Obviously, a phase of learning is required to obtain this form

of representation for each word. To achieve this scope, several algorithms are used, and the literature is always proposing new ideas.

2.2. 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 allow us to efficiently calculate the coverage of key information whilst removing the duplicate ones. Cosine similarity, Euclidean, Manhattan and Jaccard distances are the most frequently used measures in the TS field.

3. State of the Art

In the following we will explore the most common techniques for the two types of TS approaches, *Extractive* and *Abstractive*, trying to understand what is the best way to evaluate a system-generated summary. As proved in [4], human evaluation of the quality of a summary is subjective, since it is dependent on individual criteria of relevance, comprehensibility, and readability.

As we can easily understand, the Abstractive methods are much more interesting than the Extractive ones, because they use AI techniques that are close to what a human really does; the studies in the literature analyzes in depth the algorithms and the mechanisms they are based on.

A further in-depth study is also performed to gain confidence on how we can automatically evaluate a summary generated by the system. Often the metrics used for this purpose mainly offer a statistical approach to evaluation, comparing the overlapping of words from both the summary and the comparison text, forgetting to evaluate the semantic meaning of what the text itself offers.

3.1. Extractive Method

Recent research, notably in the disciplines of Deep Learning (DL) and AI, has led to the consolidation of unique and more sophisticated EATS approaches. Some of the most promising ones in the literature are covered here, including those based on Neural Networks, Graphs, Fuzzy Logic, and Semantic methods.

The main steps of a TS task are a pre-processing phase, a sentence scoring phase and a final text extraction and summary generating phase [5]. The pre-processing phase is crucial since it forms the foundation for the ultimate result that a summary will produce. Stop-words are usually removed, and part-of-speech tagging, stemming, tokenization, and normalization are usually performed. The literature proposes some approaches for representing a text with the goal of text summarization, including the usage of vectors and matrices to represent the features extracted from the text.

3.1.1. Neural Network Approaches.

Neural networks are commonly utilized, particularly for the generation of complex features from text input. In [6] an overview is provided of the most widely used algorithms today.

The *Restricted Boltzmann Machines* (RBMs) [7] are neural networks composed of an input layer and a hidden layer, where connections are made only between neurons of different layers. Deep learning is used in [8] to generate complex features from simpler ones discovered during the extraction step. Finally, a vector of matching features is constructed for each sentence. The scores acquired from the vector of features for each sentence are used to generate the summary. Also [9] proposes a deep learning-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*, which is made up of various RBMs.

The *Variation Auto-Encoder* [10] for a TS task, instead, is based on a neural network composed of an encoder, a decoder, and a loss function. The encoder and decoder are two neural networks where the output of the encoder is the input of the decoder, whilst the output of the decoder is a probability distribution. 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.

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. In [11] is presented a RNN that can perform TS on a single document. The

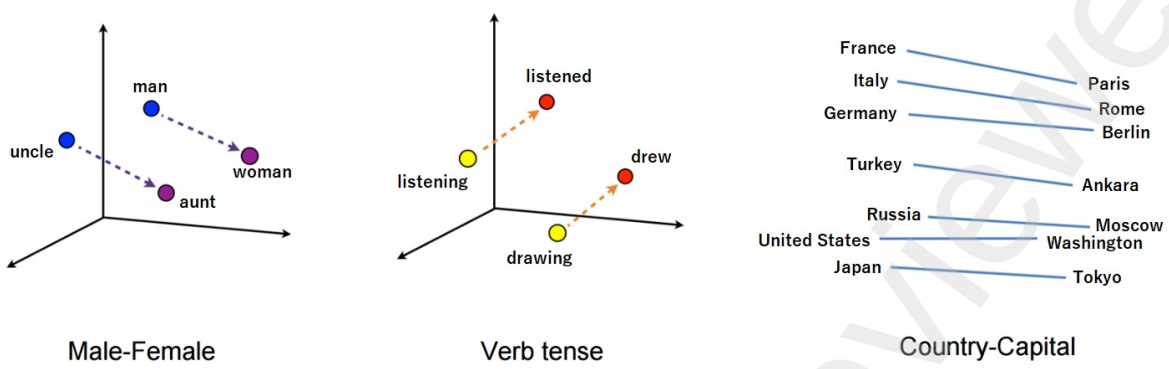

Figure 1: Multidimensional representation for word embedding.

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 in order to select summary sentences with greater accuracy and more correlations. The selection task is completed at the end of the network training phase.

3.1.2. Fuzzy Logic Based Approaches.

Defuzzifier, fuzzifier, fuzzy knowledge base and inference engine are the four fundamental components of the TS fuzzy logic approach. The fuzzy logic technique is used to feed sentence length, sentence similarity, and other textual features into the fuzzy system. [12] proposes a statistical feature-based model that uses a fuzzy model to deal with the imprecise and uncertain nature of feature weights. Redundancy removal using cosine similarity is presented as an additional enrichment, and the testing findings show that this methodology surpasses the other summarizers significantly. A fuzzy logic-based extractive summarization approach is presented in [13]. 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.

3.1.3. Graph Based Approaches.

A TS technique based on graphs is proposed in [3]. PageRank is a well-known TS model that uses a graph approach. It is built on *Google's Hits* algorithms [14]. The use of the graph as a semantic network between phrases [15] or as input for a convolutional network [16] are the two more relevant ideas in the literature.

3.1.4. Semantic Approaches.

Most frequently used TS models rely on statistical methodologies that do not take into account the semantic or contextual meaning of the text being analyzed. Sentences in a document, on the other hand, are well clustered according to Semantic Role Labeling, making it easier to build groups of similar elements. The authors of [17] employed Semantic Analysis to represent sentences, yielding encouraging summaries. Moreover, it is proven in [18] that semantic knowledge of the text is extremely important in the AATS technique.

3.2. Abstractive Method

Recently, following the widespread adoption of neural networks and DL methodologies, researchers have developed a solid foundation for the development of Abstractive algorithms, no longer linked to the traditional approaches to Natural Language Processing (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.

3.2.1. 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 with the aim of generating a vector of features. In general, specific types of neural networks are used as internal components for the encoder and decoder in the literature. We can see the use of RNNs [19] by Gated Recurrent Unit (GRU) or LSTM. The latter is the most popular since

it allows for the determination of 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 (see Fig. 2);
    - *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 the difference of an instant of time.
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 internal state of the decoder is updated with new cell weights.

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

3.2.2. Transformer Network.

In [20], a new network design called Transformer is developed, which is based only on attention-mechanism. It avoids the use of recurrence, as it has been done previously with sequence modeling, and instead offers a new technique that enables the modeling of dependencies without taking into account their distance in the input or output sequence.

3.3. Bert

Bert (Bidirectional Encoder Representations from Transformers) is presented in [21]. Its model design is based on the implementation of [20] and consists of a bidirectional multilayer transformer encoder. Unlike recent language representation models, it aims to pre-train deep bidirectional representations from unlabeled text, by conditioning on both 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 a number of tasks, such as question answering and language inference, without requiring large task-specific architecture changes.

4. Summary 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 quality of the summary, by considering which is more specific or covers more key concepts, or at least, is more readable and grammatically correct.

Instead, 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 a variety of ways, utilizing distinct phrases and words. The various evaluation criteria are depicted in Figure 3, which are 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.

4.1. 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.

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 there in the corpus.
- *Recall* determines how much of the system summary is covered by the reference summary.

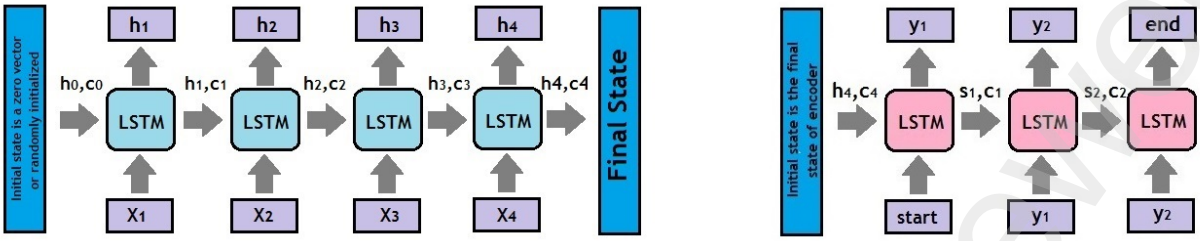

Figure 2: Encoder/Decoder representative models.

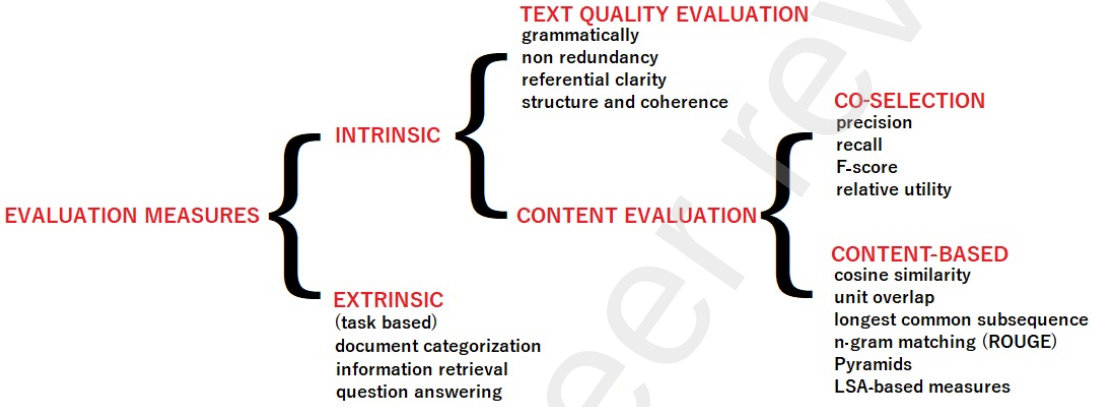

Figure 3: Evaluation methods overview.

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

1. ROUGE-N refers to the overlapping of N-grams (unigram, bigram, trigram 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;
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.

4.2. Pyramid

Pyramid is proposed as a novel way to analyze automatically generated texts [22]. 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 little piece of text that is no more than a sentence long. 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 SCU with a large weight and a growing number of SCU with a small weight. When there are multiple texts, from which SCUs can be derived, this method works well. As it is a hierarchy, this structure suggests the name of the method.

The construction of the pyramid is depicted in Figure 4. 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

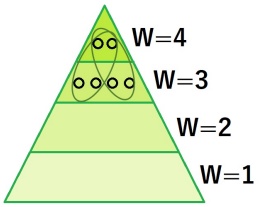

Figure 4: The Pyramidal hierarchy.

the reference one. The resulting values range from 0 to 1, with 1 indicating that the majority of the content has a high weight.

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

4.3. Semantic Similarity for Abstractive Summarization (SSAS)

Unfortunately, statistical approaches fail to detect semantic inconsistencies within a text, as well as other natural language elements like as paraphrase and logic repercussions. So, novel techniques to deal with this challenge have been proposed in the literature, particularly for the AATS algorithms. Actually, 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 both 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 a variety of natural language features, such as inference. A classification inference model is used to train the weights of various combinations of these qualities in order to arrive at a final score;
- Finally, a normalization is made, and the results are ranked.

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

5. Design and Organizations of Experiments

5.1. Research Questions

There are two research questions (RQs) in the proposed experiment. The first is motivated by a thorough review of the literature, which has uncovered some concerns about the most widely used metric for assessing the quality of Automatic Text Summarization algorithms.

This research focuses on determining how effective the ROUGE metric score is for assessing the quality of a summary, both for the Extractive and Abstractive approaches.

Based on the ROUGE scores, the second goal is to see how much better it is to employ a multiple execution of TS algorithms rather than a single execution. To summarize:

1. **RQ1:** How different is the ROUGE score for the EATS methods compared to the AATS ones? Can this metric score be representative of the quality of a summary generated by a TS algorithm?
   a) *Object of study* is the ROUGE score obtained in both the EATS and AATS algorithms;
   b) *Purpose* is to estimate the reliability and efficiency of this metric in both cases;
   c) *Perspective* is the point of view of a researcher;
   d) *Context* of experiment execution is the use of TS algorithms on various datasets of texts.

2. **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?
   a) *Object of study* is related to the multiple and single executions of TS algorithms;
   b) *Purpose* is to evaluate the quality of the generated summary by the two techniques against the ROUGE score;
   c) *Perspective* is the point of view of a researcher;
   d) *Context* of experiment execution is the use of TS algorithms on a standard dataset of texts.

5.2. Experiment Planning

The planning phase details the various steps of the experiment.

5.2.1. Context Selection

The off-line mode was used during the experiment. The purpose of the first RQ is to examine the validity and accuracy of the ROUGE metric for the two types of Automatic Text Summarization techniques. Instead, in the second RQ, two TS techniques will be compared to assess their efficacy using the same metric.

5.2.2. Hypotheses Formulation.

Two hypotheses are proposed for the statistical analysis of the experiment: the *null* and the *alternative* ones, to confirm or reject one of them. The following are formal explanations of both hypotheses for the two RQs, taking into account the ROUGE metric for comparison.

1. **RQ1**
   a) *Null Hypothesis*: The AATS methods perform differently than the EATS approaches. (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)

$$H_0 : \mu_{ROUGE\_Ext} \neq \mu_{ROUGE\_Abs} \quad (1)$$

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

   b) *Alternative Hypothesis*: The AATS methods perform almost as well as the EATS approaches. This could indicate that the ROUGE metric is not appropriate for the system-generated summary evaluation)

$$H_A : \mu_{ROUGE\_Ext} = \mu_{ROUGE\_Abs} \quad (2)$$

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

2. **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

$$H_0 : \mu_{ROUGE\_Multiple} \leq \mu_{ROUGE\_Single} \quad (3)$$

   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

$$H_A : \mu_{ROUGE\_Multiple} > \mu_{ROUGE\_Single} \quad (4)$$

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

5.2.3. Variable Selection.

The selection of variables is an important phase in the experiment planning process. The independent variables are those that we have control over and can change during the experiment. The dependent variables, on the other hand, assess the impact of the experiment on various combinations of independent variables. Our RQs are as follows:

1. **RQ1**
   a) *Independent variables*: the EATS and AATS approaches. Different algorithms will be used for each of these;
   b) *Dependent Variables*: The ROUGE score for the output of each algorithm. To provide a single comparable measure, the findings will be averaged;

2. **RQ2**
   a) *Independent variables*: TS techniques that included single and multiple executions of the summary. Different algorithms will be used in different combinations for each of them;
   b) *Dependent Variables*: The ROUGE score for the output of each algorithm.To provide a single comparable measure, the findings will be averaged.

5.2.4. Subjects Selection.

When conducting an experiment, the choice of subjects is essential since this is interconnected to the generalization of the findings of the experiment. To accomplish this, the population must be represented in the selection through an appropriate sample size. 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 chosen at random from a population list. In particular:

- For RQ1: tests are performed on four different datasets:

– **CNN/Daily Mail**, that contains editorial news and articles from CNN and the Daily Mail. Approximately 287.000 items, including summaries, make up this dataset, which was first made available for Abstractive Summarization. It is one of the most widely used datasets for ATS algorithms evaluation [19].

– **WCEP News**, a popular extractive text summarization dataset that includes documents taken from the BBC news website correlate to five main themes covered in publications from 2004 to 2005 [23].

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

– **WCEP**, a dataset composed of multi-document summaries obtained from the Current Events Portal of Wikipedia [24]. Short, human-written descriptions of news events are included in each summary, and each is paired with a selection of news items pertinent to the event.[4].

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

- For RQ2: Due to the computational difficulty and the amount of time required to complete the experiment, tests are conducted in this case only on the very large *CNN Daily Mail* dataset, taking into account a total of 1000 documents and summaries.

5.2.5. Design Type Choice and Tools.

A sequence of tests makes up an experiment. The set of tests must be carefully planned and constructed in order 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.

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 too. 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 for the execution of the experiments.

5.3. Operation Phase

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

5.3.1. Preparation.

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 in this phase for the experiment to run. It is also essential to build up the code that will collect the results. The mean scores for each block of summaries are stored in a dataset containing all calculated scores.

5.3.2. 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. In the beginning, dataset texts were chosen at random for both RQs and summarized using various algorithms. Finally, the ROUGE metric was applied to all of the summaries.

5.3.3. 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 if they met the criterion set by the algorithm creators.

[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.

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

6. Results Analysis

The results of the two experiments are described, evaluated, and interpreted in this section, with some graphs highlighting their statistical validity.

6.1. Results

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 at random from the dataset, some main elements of the results 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. To further understand the next graphs, 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 Figure 5. Instead, the subsequent Figure 6 shows each ROUGE measure distribution by three representative histograms. As anticipated by boxplots, ROUGE-1 and ROUGE-L approximate quite well 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.

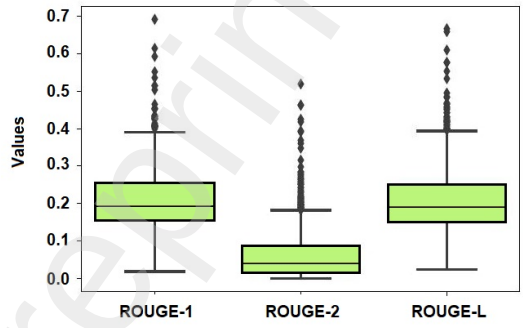

Figure 5: Boxplot of ROUGE metric scores computed on 1000 summaries by the textRank algorithm for the CNN Daily Mail dataset.

6.1.1. RQ1 results.

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.

To this aim, 40 blocks were evaluated, each consisting of 1000 summaries. To illustrate the results, an average inside each block was computed, followed by an overall average for each algorithm. Figure 8 shows the average score for the seven algorithms analyzed in relation to the three ROUGE measures. The first four algorithms are Extractive (textRank, lsa, luhn, and lexRank), whereas the last three (glove, word2vec, and doc2vec) are Abstractive.

All the algorithms have almost the same mean. LexRank is the best-performing algorithm in general, scoring approximately 10% more than the others. The Abstractive methods, on the other hand, have extremely comparable values and, even if somewhat, all of their scores are below the mean.

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 al-

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
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

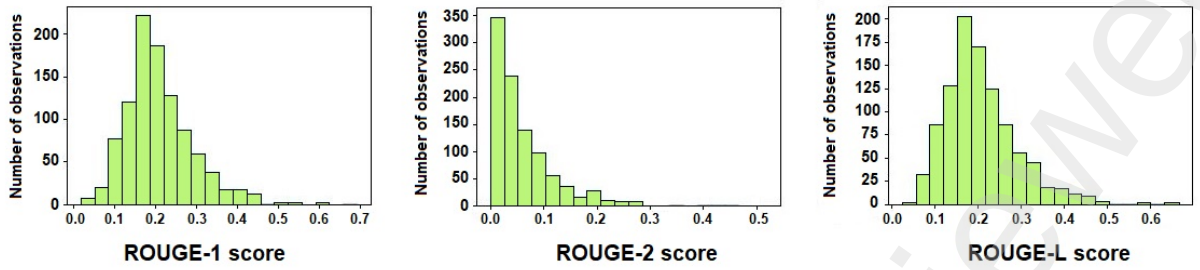

Figure 6: Histogram showing the data distribution for ROUGE-1, ROUGE-2, ROUGE-L scores using the textRank algorithm for the CNN Daily Mail dataset.

ternative 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 algorithms performed similarly in the majority of cases. ROUGE is not an appropriate metric to evaluate TS algorithms for a variety of reasons, all of which support the assumption that it is not a useful metric to analyze TS algorithms.

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 that overlap between the two texts, the more the summaries utilize different words, the worse the ROUGE metric performs. However, the semantics of statements are ignored by this system. As a result, Abstractive approaches would be at a significant disadvantage, and it may be concluded that algorithms that extract random phrases from the original text work extremely well.

We can also examine the gold standard, which is the human-generated summary, to confirm the claim that ROUGE is not very representative. It should be the best summary of a text that is currently available (and is the optimal target of our algorithms). So, when comparing several human-generated summaries based on the same source text, the outcomes can be very dissimilar, but they are all valid and acceptable. However, we would not be satisfied if we calculated the ROUGE score between two gold standards. ROUGE does not consider all of these factors and may lead us to results that do not reflect the key quality of a summary. In terms of the ROUGE score, the experiments revealed that Abstractive algorithms perform comparably to Extractive algorithms. This demonstrates that ROUGE is a poor method for evaluating TS algorithm-generated summaries.

Table 4: Mean and Standard Deviation for Bert algorithm and ROUGE metrics.

	ROUGE-1		ROUGE-2		ROUGE-L	
Algorithm	Mean	St Dev	Mean	St Dev	Mean	St Dev
Bert Ext	**0.284**	0.099	**0.093**	0.091	**0.268**	0.106
Bert Abs	**0.288**	0.099	**0.101**	0.087	**0.257**	0.100

RQ1: Bert results

Bert is an algorithm that requires a different explanation, because it is one of the most cutting-edge TS algorithms in recent years, and produces superior performances. It is accessible for both the Extractive and Abstractive techniques, with extremely similar results for both. The results of Bert are detailed in Table 4. As it can be seen in the table, it outperforms the other algorithms. In fact, BERT Extractive obtained a ROUGE-1 score of 0.284, whilst BERT Abstractive got a score of 0.288. So, it improves by roughly 30% when compared to the mean of the previous algorithms (0.22 for ROUGE-1, 0.06 for ROUGE-2 and 0.021 for ROUGE-L). In this case, the standard deviation can be an important factor that should be taken into account because it is very different. A boxplot resuming BERT scores on a single block of summaries is presented in Figure 7. We can also conclude from the figure that the boxes are slightly different. BERT produces a very good distribution of results for the Abstractive approach, but few outlier values and a more skewed distribution for the Extractive method.

RQ1: results on *BBC*, *HITG* and *WCEP* datasets

The experiments performed on the other three datasets served to further support the findings made using the *CNN Daily Mail* dataset. These are displayed below with comparison tables and diagrams. Also for

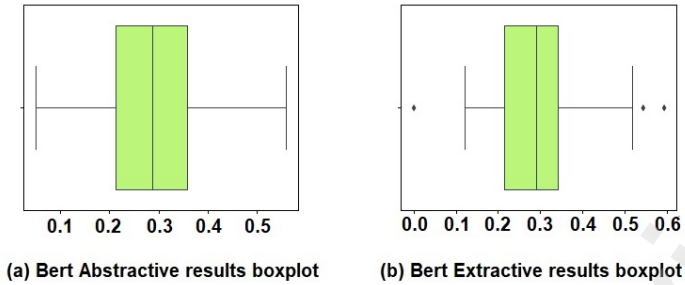

Figure 7: Comparison of results for Bert algorithm (one block of summaries).

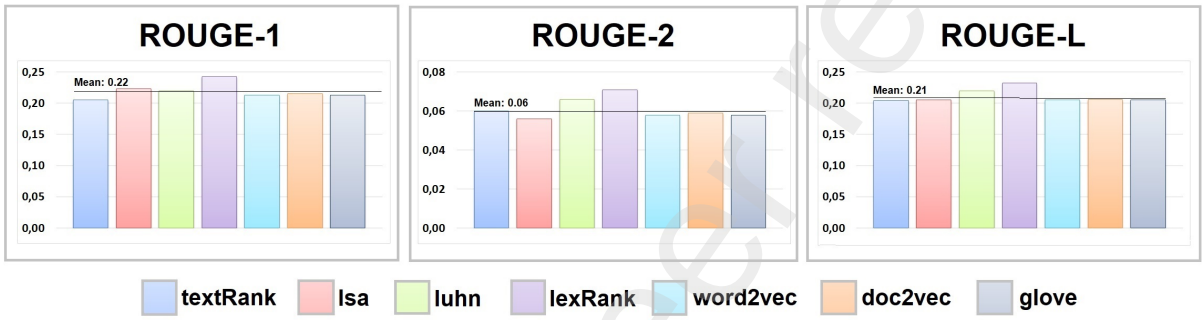

Figure 8: ROUGE average scores of the experiment conducted on Abstractive and Extractive algorithms.

Table 5: 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
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
Average	*0,295*	*0,148*	*0,211*	*0,143*	*0,289*	*0,150*

these datasets, 40 blocks were evaluated, each consisting of 1000 summaries. To illustrate the results, an average inside each block was computed, followed by an overall average for each algorithm. The conclusions derived from the examination of the *CNN Daily Mail* dataset are supported by this new evidence.

1. BBC dataset outcomes

The results obtained from the experimentation on this dataset are shown below. The distribution of each ROUGE measure is shown in Figure 9, 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. The final results of the experimentation are shown below where:

- Table 5 shows the mean and standard deviation for the seven algorithms analyzed in relation to the three ROUGE measures, and in the last row, their overall average.
- Figure 10 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. In this case Luhn is the best-performing algorithm, scoring about 14% above the average.

2. HITG dataset outcomes

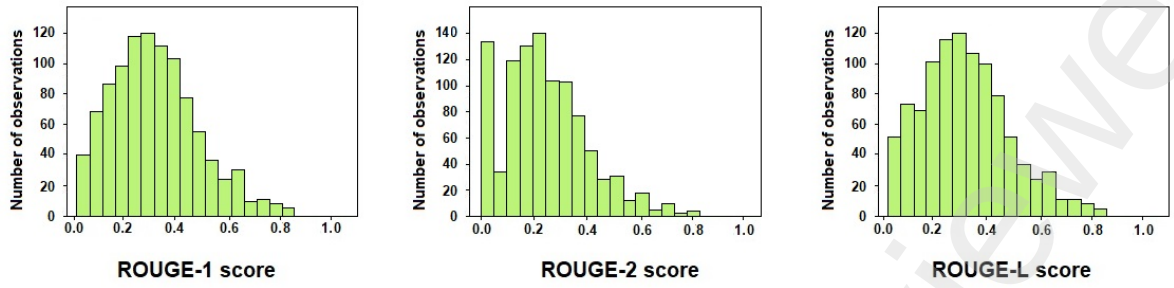

Figure 9: Histogram showing the data distribution for ROUGE-1, ROUGE-2, ROUGE-L scores using the textRank algorithm for the BBC dataset.

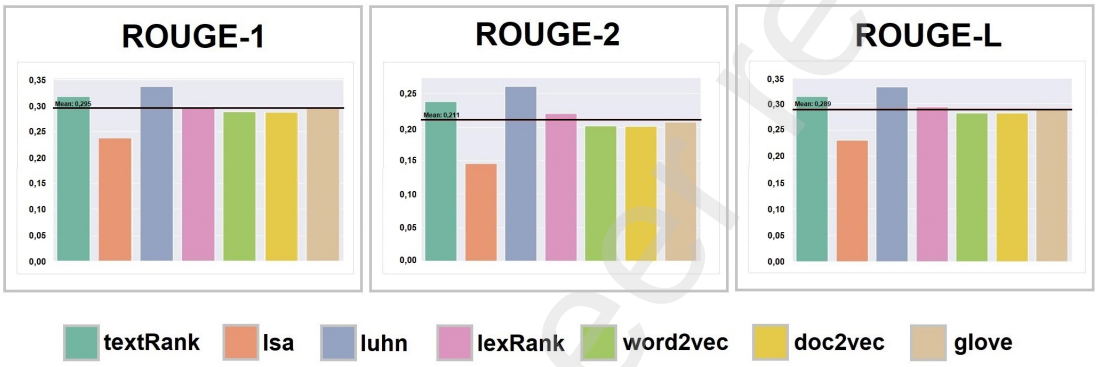

Figure 10: ROUGE average scores of the experiment conducted on Abstractive and Extractive algorithms for the BBC dataset.

The results obtained from the experimentation on this dataset are shown below. The distribution of each ROUGE measure is shown in Figure 11, by three representative histograms. We can note that none of the ROUGE measures, in this case, approximate the normal distribution. So this dataset is not very reliable according to the ROUGE metric, even if the results are not very different from the other datasets considered. The final results of the experimentation are shown below where:

- Table 6 shows the mean and standard deviation for the seven algorithms analyzed in relation to the three ROUGE measures, and in the last row, their overall average.
- Figure 12 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).

Table 6: 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
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
Average	*0,397*	*0,254*	*0,145*	*0,178*	*0,345*	*0,236*

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

3. WCEP dataset outcomes

The results obtained from the experimentation on this dataset are shown below. The distribution of each ROUGE measure is shown in Figure 13, by three representative histograms. We can note that ROUGE-1 and ROUGE-L approximate quite well the normal distribution. This guarantees the good distribution of data

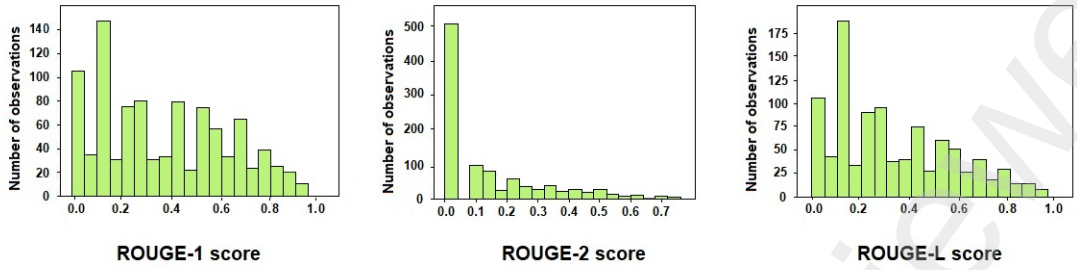

Figure 11: Histogram showing the data distribution for ROUGE-1, ROUGE-2, ROUGE-L scores using the textRank algorithm for the HITG dataset.

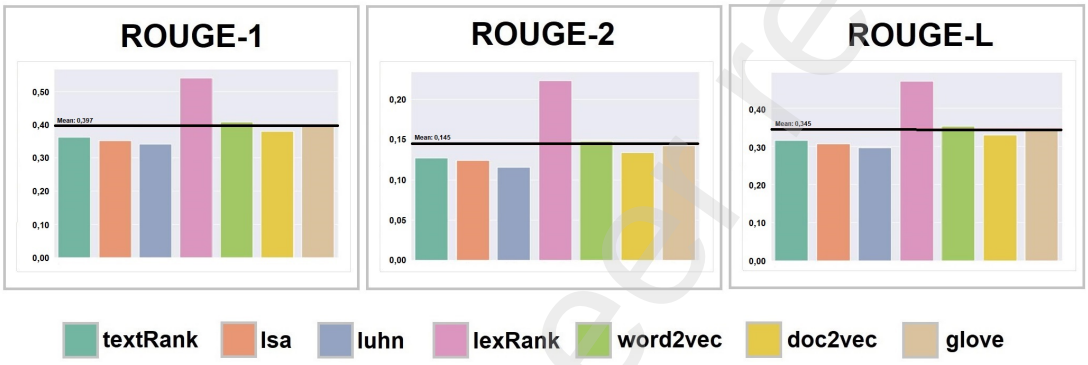

Figure 12: ROUGE average scores of the experiment conducted on Abstractive and Extractive algorithms for the HITG dataset.

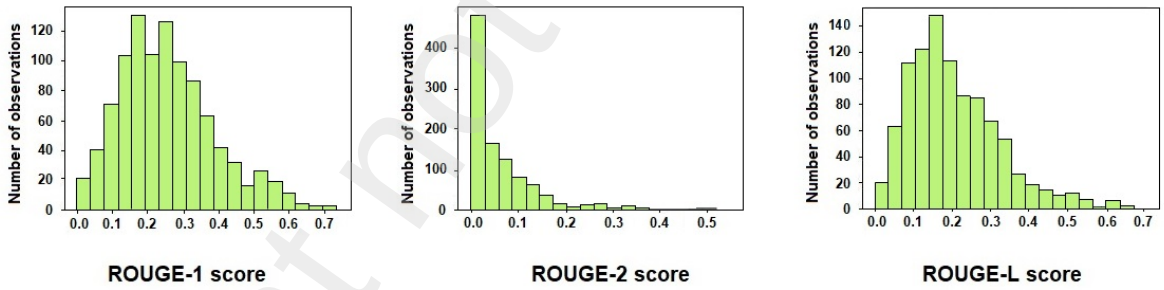

Figure 13: Histogram showing the data distribution for ROUGE-1, ROUGE-2, ROUGE-L scores using the textRank algorithm for the WCEP dataset.

points along all the observations and allows us to consider the mean as a valid representation measure for them. The final results of the experimentation are shown below where:

- Table 7 shows the mean and standard deviation for the seven algorithms analyzed in relation to the three ROUGE measures, and in the last row, their overall average.
- Figure 14 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. In this case, Luhn is the best-performing algorithm, scoring about 10% above the average.

Remembering that it was assumed that Extractive methods would perform significantly better than Abstractive approaches, the results showed that this assumption was unfounded. In most cases, the two algorithms gave similar results, so we can deduce that ROUGE is not an efficient metric to evaluate TS algorithms.

6.1.2. RQ2 results.

The second research question aimed to compare the impacts of a single and a multiple summary execution 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 8 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.

Figure 15 shows the two sheets (for ease of viewing, only the ROUGE-1 is considered). We can clearly note that multiple algorithms executions (in red) performed better than single ones (in blue) almost always.

Also in this case, a t-test is used to establish statistical validity in both types of experiments. The distinctions between the Extractive and Abstractive techniques were taken into account when conducting this test. The numerous execution technique of a summary is contrasted with the single execution strategy for each experiment. A total of 1000 paired summaries make up the population.

Table 7: 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
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
Average	*0,240*	*0,132*	*0,051*	*0,080*	*0,195*	*0,116*

Table 8: Comparison between a single and multiple execution 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

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 taken into account. As a result, the alternative hypothesis was confirmed in this experiment. Multiple execution of algorithms outperforms single ones almost always.

RQ2: Bert Results. One major exception occurs with Bert that, on the other hand, gets different results. As confirmed from the average of the ROUGE score shown in Table 9, it is higher than any other algorithms. Of course, Bert is one of the best methods that the state of the art in text summarization has to offer. It employs a novel deep neural network architecture that differs from the other algorithms investigated in this research. Un-

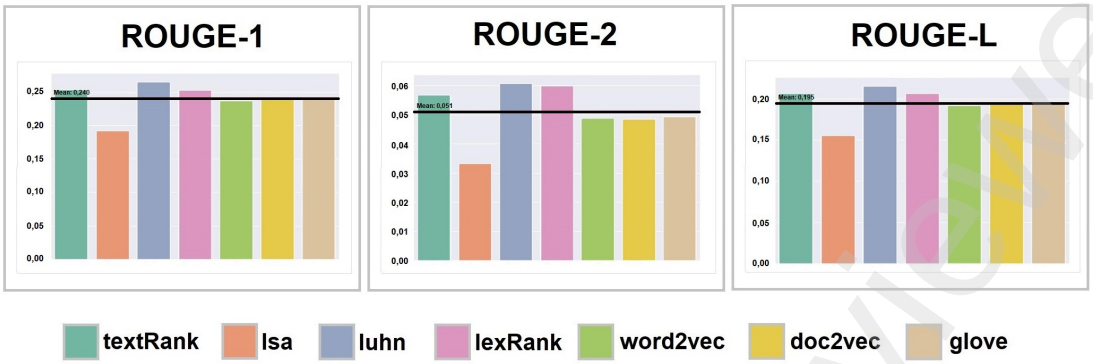

Figure 14: ROUGE average scores of the experiment conducted on Abstractive and Extractive algorithms for the WCEP dataset.

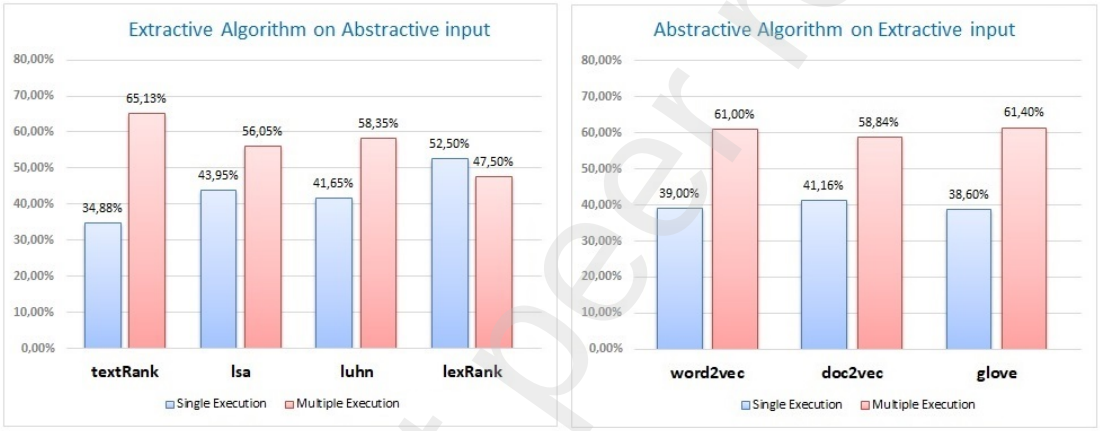

Figure 15: Results comparison for summaries between a single execution (blue) and a multiple execution (red) for each type of methodology.

Table 9: Comparison between a single and multiple execution of Extractive Bert (resp Abstractive) algorithm on the input of Abstractive (resp Extractive) algorithms.

	Performances		Mean	
	Single execution	**Multiple execution**	**Single execution**	**Multiple execution**
Bert Ext (on Abs)	69.75%	30.25%	0.2981	0.2471
Bert Abs (on Ext)	73.04%	26.96%	0.2908	0.2068

like what was expected, the single execution of Bert outperforms the multiple execution by almost 30% (see Fig. 16).

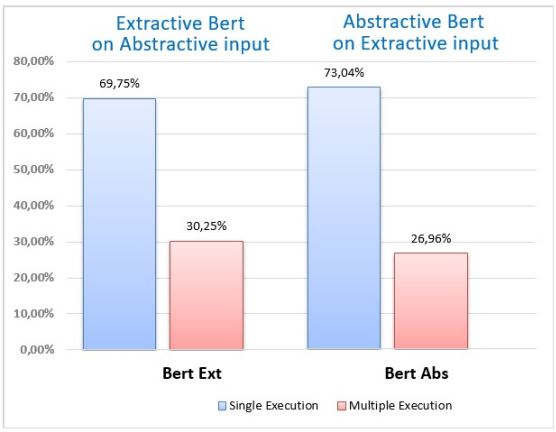

Figure 16: Results comparison for summaries between a single execution (blue) and a multiple execution (red) for Bert Extractive and Abstractive.

7. Validity Evaluation and Threats discussion

The results validity of an experiment can be compromised by various types of threats: the Conclusion, In-

ternal, Construct and External Validity. In this section, we see these threats for RQ1 (extendable to RQ2).

7.1. Conclusion Validity

The threat of having *low statistical power* was excluded. In fact, 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 well 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 that can include 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. In fact, all ROUGE metrics are computed using a standard Python library and being a well-defined algorithm, the results are also 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.

7.2. 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, the use of 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, also the developed software may have errors, especially in the implementation phase, and this 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 and the average performances of each algorithm were compared with the average standard quality provided by the developers.

Statistical regression, selection, and mortality threats, which are related to human behaviours, are not considered during the validity evaluation phase.

7.3. 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 different meanings because ”better” is not well 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 solution could be the use of different types of measures, in order to have a cross-check between them. But, as we are evaluating a measure for the quality of the text summaries, it is not possible to use a second metric.

As regards *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, in order to have a reliable statistical meaning. Each algorithm corresponds to the construct levels. Finally, to compare the two measures, all algorithms are averaged to obtain

[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”

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 performed for each portion of the text to analyze.

Interaction of different treatments and between tests and treatments are threats closely related to human behaviour in the experiment. For this experiment tests, this is impossible. Other threats in this category are closely related to the subject behaviour in the experiment, so they have not been considered.

7.4. External Validity

External validity threats are conditions that 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. For this work, this type of threat is very important because the used dataset refers to a generic text to be summarized. If the goal of the experiment 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 also on 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 of having results on standard datasets in order to have literature-based data to compare.

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 used algorithms are not perfect, because the available computing power is not sufficient to perform the experiment with more complex algorithms. For this reason, different algorithms were used in order 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.

8. Conclusions and Future work

The primary objective of this study has been the evaluation of the ROUGE metric for TS algorithms and then to determine whether a single execution of an algorithm produces better results than a multiple one. We conclude from our research that ROUGE is inefficient for TS, and then that a multiple execution yields better results than a single execution (also when evaluated by ROUGE). In conclusion, a good ROUGE score does not imply good summary quality when readability and grammatical accuracy are also taken into account.

We plan to extend the analysis to additional algorithms in the future, even though they are less known. Our idea is to develop new methods for evaluating summary quality that do not rely on statistical metrics, possibly using NLP algorithms for text comprehension.

Another future development will regard the inspection of summaries on a single topic, using algorithms trained with data from a narrowed interest field, producing more engaging and accurate results.

ACKOWLEDGEMENTS

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

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).

References

[1] 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.

[2] M. Barbella., M. Risi., G. Tortora., A comparison of methods for the evaluation of text summarization techniques, in: Proceedings of the 10th International Conference on Data Science, Technology and Applications - DATA, SciTePress, 2021, pp. 200–207.

[3] 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.

[4] P. C. F. de Oliveira, How to evaluate the 'goodness' of summaries automatically, Ph.D. thesis, University of Surrey (2005).

[5] M. R. Keyvanpour, M. B. Shirzad, H. Rashidghalam, Elts: A brief review for extractive learning-based text summarizatoin algorithms, in: 5th Intl. Conf. on Web Research (ICWR), IEEE, 2019, pp. 234–239.

[6] 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.

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

[8] S. Verma, V. Nidhi, Extractive summarization using deep learning, arXiv preprint arXiv:1708.04439 (2017).

[9] 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.

[10] M. Yousefi-Azar, L. Hamey, Text summarization using unsupervised deep learning, Expert Systems with Applications 68 (2017) 93–105.
[11] 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.
[12] D. Patel, S. Shah, H. Chhinkaniwala, Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique, Expert Systems with Applications 134 (2019) 167–177.
[13] 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. doi:10.1109/ICCS45141.2019.9065722.
[14] 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/`
[15] X. Han, T. Lv, Z. Hu, X. Wang, C. Wang, Text summarization using framenet-based semantic graph model, Sci. Prog. 2016 (2016).
[16] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, D. Radev, Graph-based neural multi-document summarization, arXiv preprint arXiv:1706.06681 (2017).
[17] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, N. A. Smith, Toward abstractive summarization using semantic representations, arXiv preprint arXiv:1805.10399 (2018).
[18] 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, International Journal of Parallel Programming 46 (5) (2018) 992–1016.
[19] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al., Abstractive text summarization using sequence-to-sequence rnns and beyond, arXiv preprint arXiv:1602.06023 (2016).
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. in Neural Inf. Processing Systems 30 (2017) 5998–6008.
[21] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[22] 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.
[23] 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.
[24] 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.