AI

MDPI

Article

Who Needs External References?—Text Summarization Evaluation Using Original Documents

Abdullah Al Foysal * and Ronald Böck *

Research Division, Genie Enterprise, Donnersbergweg 1, 67059 Ludwigshafen, Germany
* Correspondence: afoysal@genie-enterprise.com (A.A.F.); rboeck@genie-enterprise.com (R.B.); Tel.: +49-621-12185290 (A.A.F.); +49-621-16639018 (R.B.)

Abstract: Nowadays, individuals can be overwhelmed by a huge number of documents being present in daily life. Capturing the necessary details is often a challenge. Therefore, it is rather important to summarize documents to obtain the main information quickly. There currently exist automatic approaches to this task, but their quality is often not properly assessed. State-of-the-art metrics rely on human-generated summaries as a reference for the evaluation. If no reference is given, the assessment will be challenging. Therefore, in the absence of human-generated reference summaries, we investigated an alternative approach to how machine-generated summaries can be evaluated. For this, we focus on the original text or document to retrieve a metric that allows a direct evaluation of automatically generated summaries. This approach is particularly helpful in cases where it is difficult or costly to find reference summaries. In this paper, we present a novel metric called Summary Score without Reference—SUSWIR—which is based on four factors already known in the text summarization community: Semantic Similarity, Redundancy, Relevance, and Bias Avoidance Analysis, overcoming drawbacks of common metrics. Therefore, we aim to close a gap in the current evaluation environment for machine-generated text summaries. The novel metric is introduced theoretically and tested on five datasets from their respective domains. The conducted experiments yielded noteworthy outcomes, employing the utilization of SUSWIR.

Keywords: automatic text summarization; latent semantic analysis; cosine similarity; jaccard similarity; named entity recognition

Citation: Foysal, A.A.; Böck, R. Who Needs External References?—Text Summarization Evaluation Using Original Documents. *AI* **2023**, *4*, 970–995. https://doi.org/10.3390/ai4040049

Academic Editor: Arslan Munir

Received: 21 September 2023
Revised: 28 October 2023
Accepted: 7 November 2023
Published: 15 November 2023

1. Introduction

Automatic Text Summarization (ATS) has recently gained importance in helping the capture of enormous amounts of information in multiple modalities, especially as the internet presents an (almost) endless source of data in terms of audio–visual information as well as textual samples. In this paper, we focus on text data, usually enriched with an abundance of details, offering information beyond our needs and sometimes increasing the effort required to identify the most relevant snippets [1]. Fortunately, (intelligent) search algorithms help to identify interesting data since millions of documents are generally available. Even specific search requests usually result in at least thousands of hits, nevertheless, longing to collect the particularly important aspects within the text. Further, a huge amount of text-based information is generated every day, making it difficult for individuals and organizations to manage, understand, and utilize these details. This discussion can be easily extended to collections of texts [2].

In general, not just considering the internet, where algorithms (somehow) assist the users [3], manual selection of important information from a large volume of data, perhaps spread across multiple sources, is very challenging for human beings. Consequently, text summarization has become an important component of every text analysis tool, assisting users to identify and navigate large pieces of content quickly by providing short and coherent summaries of key ideas. In this sense, ATS extracts or rewrites the most important

information from an authentic text document. The main goal of ATS is to minimize the size of the source text while maintaining all important aspects. The authors of [4] highlight four main reasons behind why ATS tools are beneficial:

- Summaries *minimize* reading *time and effort*.
- Summaries make document *selection easier* when researching.
- Automatic text summarization *enhances indexing* efficiency.
- Automatic text summarization techniques are *less biased* compared to human indexing.

An important element of ATS is the evaluation of the generated output. Initially, automatically produced summaries were evaluated manually by human judges, who were entrusted with determining the quality of the results. Although this procedure presents multiple challenges, there are at least two of them that should be kept in mind:

1. Human judges may have their own biases, preferences, and opinions that influence their evaluations [5]. These aspects can impact the assessment of summary quality and thus, make the results less objective.
2. Human evaluations can also often be time- and resource-consuming, particularly when it comes to difficult text summarization tasks [6].

Research, for instance in [5,7], already shows the effectiveness of human scoring as the primary metric for evaluating text summaries and discusses the reliability and challenges associated with human evaluation.

However, to reduce manual effort, increase objectivity, and allow scalability, multiple evaluation methodologies were already introduced, for instance, Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [8], Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance (MoverScore) [9], etc. Nevertheless, these state-of-the-art evaluation metrics also bear some drawbacks and limitations, which are described in Section 2. Therefore, ideally, an advanced summarization metric is needed to evaluate machine-generated summaries tackling the challenges additionally stated in Section 2.

This metric should be constructed in the following way: While evaluating and measuring the quality of summaries, this metric considers factors like coherence, informativeness, redundancy, and bias avoidance. Additionally, specific focus might be placed on tasks where a reference summary is not available or is being created with enormous effort. Thus, the creation of proper evaluation methods is important for guiding the development of efficient summarization models.

In this manuscript, we propose a novel metric that avoids the necessity for reference summaries. Instead, we use the original text as a reference, supposing it already represents all essential details. For this, the suggested metric (cf. Section 3) is based on four important factors, showing high relevance in the following domains, Semantic Similarity, Redundancy, Relevance, and Bias Avoidance Analysis. Details on the metric's composition are introduced and discussed in Section 3. Besides theoretical concepts, we also tested our approach on several data sets from multiple content domains (cf. Section 4.1). The respective results and discussions are presented in Section 4. Before we present the novel metric in more depth, we recapitulate the current state-of-the-art and derive respective research questions throughout the next sections.

2. Related Work and Motivation

Natural Language Processing (NLP) has made great progress in general, specifically in the domain of evaluating text summarization techniques. We already see a lot of effort to estimate the quality of (automatic) text generation approaches as well as related summaries, in order to fulfill the requirements for accurate and comprehensive assessments. These approaches allow us to analyze the summary quality on both a quantitative and qualitative level, supporting in a technical perspective the creation and improvement of summarization models. However, considering the assessment of automatically generated summaries, we still see no specific evaluation approaches, focusing only on this domain. Existing work is mainly transferred from other domains to estimate the summary's quality. The current

approaches will be sketched in the following, highlighting the particular use in the domain of automatic text summarization.

Regarding the current literature, we identified several automatic evaluation metrics, selecting the most prominent approaches: ROUGE [8], Bilingual Evaluation Understudy (BLEU) [10], and Metric for Evaluation of Translation with Explicit Ordering (METEOR) [11]. These metrics mainly focus on content overlap, n-gram matching, and linguistic features to gauge the fidelity of generated summaries. In the field of text summarization, ROUGE is by far the most popular and widely used metric. It offers a comprehensive evaluation of the effectiveness of summarization by evaluating several factors such as precision, recall, and F1-score, making it a possible option for summarization evaluation (cf. [8]). However, it faces limitations when it comes to capturing word order and semantic meaning. Also, it is quite sensitive to even small changes in reference summaries, and thus, often generates inconsistent results [8]. ROUGE typically involves the comparison of generated summaries to human-generated references. The authors applied Pearson's correlations to understand how well the ROUGE scores (automatic evaluation metric) align with human-assigned scores for content coverage. However, in our case, where human reference summaries are not available, but the original content is used as a reference, Pearson's correlation may not be applicable.

The BLEU algorithm was initially developed to evaluate machine translation tasks. However, an adaptation towards the evaluation of text summarization tasks could be achieved [10]. In this sense, BLEU measures the precision of n-grams in the machine-generated summary, concerning the human reference summaries, providing a clear and understandable score. However, since BLEU favors short summaries, being the metric's major drawback, it might not accurately reflect the quality of longer descriptions as BLEU is more precision-oriented. The longer sentences tend to have lower precision because there are more opportunities for mismatches. It also fails to identify the semantic meaning of a word [10].

METEOR is another metric that was originally designed for machine translation tasks. Both machine translation and summarization involve natural language generation from a source text. So, it is possible to use METEOR to evaluate the quality of text summaries in a similar manner as it evaluates machine translation. One of METEOR's key features is that it considers a variety of linguistic concepts, including stemming and synonym matching. It can therefore handle paraphrased texts better than BLEU and ROUGE [11]. However, METEOR can lead to inaccuracies when evaluating domain-specific or seldom-encountered words as it uses a generic synonym dictionary (e.g., WordNet) and a stemming algorithm (e.g., Porter Stemmer). Also, METEOR results can be changed based on the tunable parameters such as penalty values [11]. If these parameters are not appropriately set, METEOR may produce unreliable evaluation scores.

Character n-gram F-score (CHRF) [12] evaluates summaries at the character level, capturing content overlap and penalizing slight differences. This approach encounters difficulties where semantic understanding and fluency matter more than achieving an exact match. CHRF mitigates the sensitivity of n-gram metrics to synonym and paraphrase usage. Although CHRF is similar to ROUGE and BLEU (evaluate text by measuring the overlap of n-grams). However, the advantages of CHRF include its simplicity and resistance to changes in word order. Nevertheless, it has also a limited semantic understanding [12].

The aforementioned measurements offer useful insights into lexical alignment and fluency but may ignore details linked to coherence and contextual understanding. In response, additional techniques like MoverScore [9] and ROUGE with Word Embeddings (ROUGE-WE) [13] have been developed that use word embeddings and semantic metrics to capture more in-depth content preservation and semantic similarity. Their particular characteristics will be briefly discussed.

MoverScore uses word embeddings to measure semantic similarity between words in the human reference and machine-generated summaries. This creates robustness to capture semantic relationships, which is one of the key aspects of evaluating the quality

of summaries. However, MoverScore requires pre-trained word embeddings. Therefore, the metric's assessment is influenced by the particular models being used for word embedding [9].

The ROUGE-WE metric solved the problem of the traditional ROUGE approach. However, the performance of ROUGE-WE still depends on external resources like different word embedding models and human-generated references.

In [14], the authors sketched a metric, utilizing a pre-trained masked language model for evaluating summarization and data-to-text generation. However, they do not elaborate further details on an evaluation method for scenarios where human reference summaries are absent.

The authors of [15] discuss the use of Latent Dirichlet Allocation to improve the comprehension of scientific articles. Instead of reading the whole article, they concentrate on studying abstracts. The authors claim that this approach is quite helpful in understanding the text. This approach links to the evaluation of summaries because it extracts the main topics from different texts (like machine-generated and human-generated summaries), allowing a comparison to see how well a summary captures the main information. In our approach, we tackle this aspect using Latent Semantic Analysis (LSA) (cf. Section 3.1.1)

In our proposed method, we relied on LSA and Cosine Similarity (CS) to identify matches and relations within texts (cf. Section 3.1). This combination is a well-known setting for this task. However, the authors of [16] suggest another way to retrieve information from data sets and find relations using divided spaces. From this perspective, the method in [16] could be used in future work to access corresponding entities and relations faster.

In general, for a detailed explanation of each measure, we refer to the particular references. Nevertheless, Table 1 summarizes the main characteristics of each approach. All methods share one common aspect: each metric depends on (human-generated) reference summaries. Therefore, from our point of view, it is necessary to develop a novel approach that is able to evaluate the quality of a text summary in the absence of a reference.

Table 1. Coverage of characteristics of several summary evaluation metrics regarding different factors. Details on the respective metrics are given in Section 2.

Metric	Factors				
	n-Gram Matching	**Semantic Similarity**	**Stemming**	**Synonym**	**Resource Dependent**
ROUGE	✓	✗	✗	✗	✗
METEOR	✓	✓	✓	✓	✗
BLEU	✓	✗	✗	✗	✗
ROUGE-WE	✓	✓	✗	✓	✓
MoverScore	✓	✓	✗	✓	✓

In the following, we conclude the discussion from the introduction and related work to consolidate the motivation of the current manuscript.

As we already discussed in Section 1, the evaluation of results in tasks like text summarization can be very challenging. Especially for summarization tasks, multiple correct answers are usually possible, depending on several factors and circumstances, for instance, the purpose of the original text or summary, the length of the summary, and the person or system generating the summary. Currently, the community relies on reference summaries (generated by human professionals) and model summaries (generated by algorithms) in order to apply the existing evaluation methods (cf. Section 2), achieving an assessment of quality for the automatically created summarization. However, the collection of reference summaries sometimes might be difficult and require much effort.

Therefore, our goal is to establish a novel metric for the evaluation of text summarization, reducing or ideally avoiding current drawbacks (cf. related measures in Section 2 and

Table 1), working in a reference-free manner. Given this goal, we address the following research questions in the remaining manuscript:

RQ1: Is it possible to evaluate the quality of a machine-generated summary if we do not have a reference summary?
RQ2: Under which circumstances does the technique of using the original content as a point of reference yield the maximum level of effectiveness?
RQ3: Can the process of evaluating machine-generated summaries utilizing original input data be more trustworthy and accurate than conventional methods?
RQ4: What are important factors to rely on in reference-free metrics?

3. Proposed Approach and Methodology

Our main objective is to evaluate summaries produced by machines without requiring a (human-generated) reference summary. We use the original content as the reference text in the novel metric, called Summary Score without Reference (SUSWIR). In general, the approach is constructed as presented in Equation (1):

$$\begin{aligned} \text{SUSWIR}(\underline{X}, \underline{Y}) &= W_1(\text{SSF}(\underline{X}, \underline{Y})) + W_2(\text{RLF}(\underline{X}, \underline{Y})) + \\ & \quad W_3(\text{RDF}(\underline{Y})) + W_4(\text{BAA}(\underline{X}, \underline{Y})) \end{aligned} \tag{1}$$

In this sense, SUSWIR($\underline{X}, \underline{Y}$) is the weighted combination of four factors that assess the quality of the **summary** $\underline{Y}$ in relation to the **original document** $\underline{X}$, where $\underline{X}$ and $\underline{Y}$ represent raw texts, respectively. Every factor is given a weight, represented by $W_i \in [0, 1]$, $\sum_i W_i = 1$, which indicates its relative importance of each factor. In particular, we consider the following four factors, which will be further introduced and discussed separately in the respective sections:

- Semantic Similarity Factor (SSF): This factor evaluates how well the content of summaries matches the underlying semantic meaning of the original text. In order to maintain the overall concepts and ideas, a good summary must capture the core semantics of the original content. The high semantic similarity suggests that the summary accurately represents the original's content, enhancing the quality and informativeness of the summary.
- Relevance Factor (RLF): This factor quantifies how well the summary conveys the important details from the original content. It examines how closely the summary follows the key ideas of the original content or text. An effective summary highlights the important points and skips unnecessary details.
- Redundancy Factor (RDF): This factor measures the amount of redundant information in the summary. Overly redundant summaries normally represent less information, and thus, do not help readers. The evaluation of redundancy ensures that the summary conveys diverse and distinct information, resulting in a concise and useful document.
- Bias Avoidance Analysis (BAA): This factor checks for the introduction of subjective opinions or biases in the summary that were not in the original content.

These factors directly impact the effectiveness and comprehensibility of (automatically) generated summaries. A machine-generated summary is more accurate, concise, relevant, and fair when these factors appear together because they contribute to its overall quality.

The total score for SUSWIR is calculated by summing up these different factors. This process allows for a comprehensive assessment of the summary's quality, taking into account various dimensions. In this sense, SUSWIR has a range from 0 to 1, with higher scores suggesting that the summary covers key information from the original content. Although the proposed metric generates an overall score for the summary assessment, the individual factors are still important. In the evaluation process, each factor offers additional insights into the processed data, providing the option to focus on particular aspects of the assessment. In this sense, the weights W_i allow to influence each factor. For instance, low values in BAA indicate that the summarizer is likely to make biases. In fact, we highly

recommend observing the four factors to become aware (early) of unwanted effects in the automatic summarization. However, a fine-tuning of individual weights is suggested to adapt SUSWIR to the particular needs and characteristics of the task and dataset. In the current proof-of-concept, we do not apply any fine-tuning to the weights or factors (cf. Section 4.3).

Given these general observations, we now discuss in more detail the particular factors (cf. Sections 3.1–3.4), being combined to determine SUSWIR.

3.1. Semantic Similarity Factor

SSF aims to quantify the level of semantic relatedness or similarity between the original content and its summary. This evaluation provides a measure to determine how the summary captures the main ideas and meaning of the original content. This is important to assess the informativeness and quality of text summaries.

To calculate the SSF between the original content and its associated summary, LSA is used, followed by the calculation of CS. This strategy captures complex text relationships by understanding word meanings and tapping this for tasks like document similarity. This goes beyond basic word matching and reveals the context and ideas within the text. In particular, CS, described in Section 3.1.2, is the backbone of SSF. We used the LSA-transformed vector for SSF to improve our evaluation. The LSA concept is presented in Section 3.1.1. The decision to use LSA along with CS rather than using direct CS is supported by (at least) two arguments:

- When working with diverse text corpora, direct CS struggles to capture the relationships between words and documents. Additionally, it cannot properly handle synonyms or polysemous words and is sensitive to noisy data.
- LSA simplifies text analysis by finding hidden meanings and reducing the complexity of the data. It also helps to filter out noise from the original data, making similarity measures more effective and robust. In order to deal with synonyms, LSA also has the ability to group related words together [17].

We consider an example: Given two text documents that show following information:

T1: "The stock market experienced a significant plunge".
T2: "Share prices in the financial market plummeted sharply".

If we just use direct CS, there will be a high chance that almost no similarity is found between T1 and T2, because the words in the two documents are quite different. CS is not able to capture the semantic meaning of the text properly. In contrast, a combination of LSA and CS, will at first identify a semantic representation of the two statements and recognizes that "stock market" and "financial market" have almost the same meaning; in a second step, the similarity is calculated afterward. Therefore, LSA helps us to focus on what the words actually mean by simplifying things. For this, computing "similarity" in this manner, the sub-metric recognizes the shared meaning and provides a higher score.

In the following, we introduce the combined concepts of LSA and CS in a theoretical sense and thus, explain how SSF has been calculated.

Given the original content $\underline{X}$ and the respective summary $\underline{Y}$, at first, we created a list $\underline{D}$ which contains both, the original content and the summary, according to Equation (2). In a simplified way, $\underline{D}$ represents a list of documents.

$$\underline{D} = [\underline{X}, \underline{Y}] \tag{2}$$

Based on this listing, vectorization is applied. In our approach, we used the "TF-IDF" vectorizer [18] to create a document-term matrix (TF-IDF matrix) for documents in $\underline{D}$, resulting in the matrix (numerical representation of texts) given in Equation (3).

$$\mathbf{D}_{\text{tfidf}} = f_{\text{tfidf}}(\underline{D}) = \begin{bmatrix} \underline{X}_1 & \underline{X}_2 & \cdots & \underline{X}_m \\ \underline{Y}_1 & \underline{Y}_2 & \cdots & \underline{Y}_m \end{bmatrix}, \tag{3}$$

where

- $\mathbf{D}_{\text{tfidf}}$ is the TF-IDF matrix of the list of text $\underline{\text{D}}$,
- f_{tfidf} represents the function that generates the TF-IDF vectors,
- $\underline{\text{D}}$ is the input of the function f_{tfidf},
- $\underline{\text{X}}_{1...m}$ are the TF-IDF scores of each terms in original content $\underline{\text{X}}$ relative to the list of texts or documents in $\underline{\text{D}}$,
- $\underline{\text{Y}}_{1...m}$ are the TF-IDF scores of each terms in summary $\underline{\text{Y}}$ relative to the list of texts or documents in $\underline{\text{D}}$.

LSA is applied to the vectorized documents $\mathbf{D}_{\text{tfidf}}$, resulting in Equation (4):

$$\mathbf{D}_{\text{LSA}} = f_{\text{LSA}}(\mathbf{D}_{\text{tfidf}}) = \begin{bmatrix} \underline{\text{X}}_1 & \dots & \underline{\text{X}}_N \\ \underline{\text{Y}}_1 & \dots & \underline{\text{Y}}_N \end{bmatrix}, \tag{4}$$

where $\mathbf{D}_{\text{LSA}}$ gives us a $R \times N$ shaped matrix and f_{LSA} represents the function that generates the new matrix using the vectorized documents $\mathbf{D}_{\text{tfidf}}$. R (number of rows in the LSA matrix) represents the number of documents, in our case documents are the original content and the summary. Each document is represented as a vector in the TF-IDF matrix. N represents the number of latent topics or dimensions. LSA allows us to choose how many latent topics or dimensions we want to retain. If we choose to retain N dimensions, then N is the number of columns in the LSA matrix.

For this, we can derive the CS components as follows:

$$\mathbf{x} = \mathbf{D}_{\text{LSA}}[0] = [\underline{\text{X}}_1 \ \dots \ \underline{\text{X}}_N] \tag{5}$$

$$\mathbf{y} = \mathbf{D}_{\text{LSA}}[1] = [\underline{\text{Y}}_1 \ \dots \ \underline{\text{Y}}_N], \tag{6}$$

where

- $\mathbf{x}$ or $\mathbf{D}_{\text{LSA}}[0]$ shows the first row of $\mathbf{D}_{\text{LSA}}$ matrix which represents the LSA-transformed vector corresponding to the original content. $\underline{\text{X}}_{1...N}$ indicates the importance score between the original content and each latent topic $(1 \dots N)$ based on LSA.
- $\mathbf{y}$ or $\mathbf{D}_{\text{LSA}}[1]$ shows the second row of $\mathbf{D}_{\text{LSA}}$ matrix which represents the LSA-transformed vector corresponding to the summary. $\underline{\text{Y}}_{1...N}$ indicates the importance score between the summary and each latent topic $(1 \dots N)$ based on LSA.

Finally, SSF (cf. Equation (7)) represents the similarity of $\underline{\text{X}}$ and $\underline{\text{Y}}$ based on those vectors generated in Equations (5) and (6).

$$\text{SSF}(\underline{\text{X}}, \underline{\text{Y}}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \tag{7}$$

3.1.1. Latent Semantic Analysis

LSA is a method that reduces the number of dimensions and finds hidden patterns in a group of text documents. It is often used to capture the semantic meaning and relationships between words and documents [19]. LSA can be generally divided into three different phases:

- **Document-word frequency matrix:** In this phase a document-word matrix of shape $m \times n$ is created, where m represents the number of documents and n represents the number of unique words having scores. There are many ways to create a document-word frequency matrix, including methods like Bag-of-Words and TF-IDF. In the current manuscript, we used the TF-IDF approach.
- **Singular Value Decomposition** (SVD): After calculating the document-word frequency matrix, the SVD technique is applied to reduce the dimensionality of this matrix [20]. The reason is that the initial matrix might be very sparse, noisy, and redundant across its many dimensions.

- **Topic encoded data:** After applying SVD to the document-word frequency matrix, topic-encoded data is produced by preserving the first few columns of the **U** (document-topic matrix) and **V** (term-topic matrix) matrices corresponding to the most important singular values. It means that in SVD, the singular values represent the importance of different patterns or dimensions in the original data. These values are ordered from most significant (dominant) to least significant. By preserving the columns associated with the dominant singular values, we effectively capture the most meaningful information and relationships in the data while reducing its dimensionality. This process helps to distill the essential semantic relationships between words and documents, making the data more manageable and informative.

Given these steps, a decomposition of the input matrix can be achieved according to Equation (8)

$$\text{LSA}(\mathbf{X}) = \mathbf{U}\mathbf{S}\mathbf{V}^{\text{T}} \tag{8}$$

Let **X** be the document-term matrix obtained from TF-IDF vectors of the original content and summary. The LSA process decomposes **X** into three matrices:

1. **U** (document-topic matrix): This represents information about the relationships between documents and topics.
2. **S** (singular values matrix): This is a diagonal matrix with the singular values denoting each topic's significance on it.
3. $\mathbf{V}^{\mathbf{T}}$ (term-topic matrix): This represents information about the relationships between words and topics.

3.1.2. Cosine Similarity

CS is a metric that measures the cosine of the angle between two vectors. It is commonly used in text analysis to calculate the similarity between two documents based on their word frequency distributions. It is established through a mathematical definition, involving the dot product of two vectors divided by their respective magnitudes [21]. Therefore, the similarity between vectors **a** and **b** is calculated as follows (cf. Equation (9)):

$$\text{CS}(\mathbf{a}, \mathbf{b}) = \cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|}, \tag{9}$$

where

- θ is the angle between the vectors; the greater the angle, the less similar they are.
- The dot product of **a** and **b** is denoted by $\mathbf{a} \cdot \mathbf{b}$.
- The vector's L2 norm or magnitude is represented by $\|\mathbf{a}\|$.

3.2. Relevance Factor

The main goal of the relevance factor is to measure how well the summary captures the key points and ideas present in the original content. It, therefore, evaluates the semantic alignment between both documents.

In this context, the METEOR method is selected to gauge the quality and relevance of a summary in relation to the original content. We skip using methods like BERT [22], SBERT [23] etc., for calculating relevance scores as well as for other linguistic aspects because those methods demand a lot of computing power, memory, and a massive amount of training data. To keep our approach less dependent on external resources (e.g., in terms of computational power; cf. Section 4.2), we opted for the METEOR method. We employed METEOR in the SUSWIR approach due to its linguistic features, such as stemming and synonym matching, making it particularly well-suited for calculating relevance scores. Both, machine translation (original domain of METEOR) and summarization, involve generating coherent natural language from source texts, forming a basis for METEOR's application. METEOR's "Chunk Penalty" is particularly valuable as it assesses the original order and structure of the content, aiding in evaluating summary fluency. Hence, we have

chosen METEOR to compute relevance scores for our proposed approach. While METEOR has known limitations (as discussed in Section 2), we have addressed these concerns by incorporating additional factors into the SUSWIR approach.

For a better understanding of the manuscript's approach, we briefly introduce the main concepts of METEOR and adapt them with respect to the assessment of summary quality. Details of the particular methods can be found in [11]. METEOR computes a score that represents the semantic alignment and similarity between the so-called reference (original content) and the candidate (summary) sentences. For this, it takes into account both, exact word matches and similar word changes, that preserve the same meaning. Therefore, the measure provides a detailed evaluation, which is beyond mere overlaps of (basic) words, allowing to check the relevance of the summary.

In the following, we address the steps to achieve a METEOR-based assessment, derived from [11]. The overall measure is based on the penalized version of the F-score. In this sense, recall R (Equation (10)) and precision P (Equation (11)) are introduced.

$$R = \frac{\text{Number of Matching Unigrams between the Reference and Original Content}}{\text{Total Number of Original content Unigrams}} \tag{10}$$

$$P = \frac{\text{Number of Matching Unigrams between the Reference and Original Content}}{\text{Total Number of System Unigrams}} \tag{11}$$

According to [11], the F-score F (cf. Equation (12)) is specifically defined as follows, combining both, precision and recall, into one value.

$$F = \frac{10PR}{R + 9P} \tag{12}$$

Additionally, METEOR uses a term called Chunk Penalty (CP, cf. Equation (13)), which is calculated based on the number of contiguous words that are matched between the original content and summary sentences [11]. This helps to assess the quality of machine-generated summaries by checking whether the summarized content retains its original order and structure. This is important because it ensures that the summary makes sense and flows well, much like how a human would summarize a document.

$$\text{CP} = \lambda \cdot \left(\frac{C}{U_m}\right)^3, \tag{13}$$

where

- CP represents the Chunk Penalty.
- C represents the number of chunks in the machine-generated summary.
- U_m represents the number of unigrams in a machine-generated summary.
- λ is a parameter that adjusts the impact of the chunk penalty; often set to a low number (e.g., 0.25).

Given the aforementioned statements and equations, finally the Relevance Factor (RLF; Equation (14)) can be calculated. It is worth noting that in the current implementation of SUSWIR, it is equivalent to the METEOR score.

$$\text{RLF}(\underline{X}, \underline{Y}) = F \cdot (1 - \text{CP}) \tag{14}$$

3.3. Redundancy

For the assessment of redundancy, we rely only on the information included in the summary itself. Redundancy frequently occurs when many phrases express the same idea, leading to needless repetition.

Figure 1 shows the visual representation of how we calculate redundancy in a summary. In the example, the summary contains m sentences, represented by S_m, which might provide redundant details.

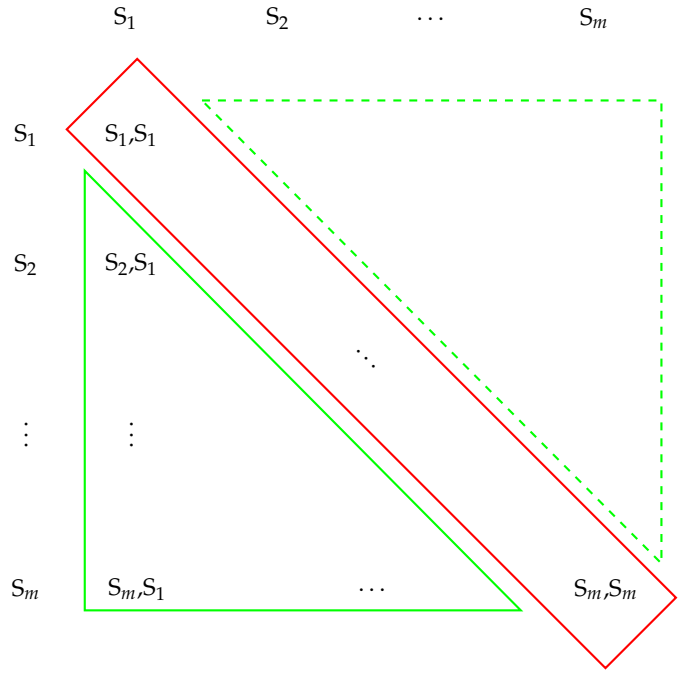

Figure 1. Visual representation of redundancy calculation. In this graphical representation, we consider m sentences ($S_1 ... S_m$). There is a total of m^2 unique relationships to calculate similarity scores, obtaining a redundancy assessment. Among these, m number of relationships involve exact duplicates such as (S_1, S_1), (S_2, S_2), (S_3, S_3),..., (S_m, S_m), highlighted in red, resulting in a score of 1.0. The remaining relationships will be included in the calculation. However, there are pairwise repetitions in this matrix-like structure (e.g., (S_1, S_2) or (S_2, S_1)), indicated as a dashed green triangle. To determine the redundancy score, we focus on the unique relationships, like (S_2, S_1), on the secondary diagonals. So, we can consider the lower triangular (thick green line) from this figure and avoid the upper triangular (dashed green line) as it represents the mirror image of the lower triangular.

First, we consider each sentence as an individual piece of information. We calculate the CS between each pair of sentences, neglecting self-comparisons (highlighted in red in Figure 1), in order to determine the degree of repetition. Self-comparisons are skipped as they always result in a similarity of one, and thus do not provide valuable input to redundancy. The lower and upper triangular in Figure 1 (highlighted in green) visualize unique pairs of sentences, appearing maybe in permuted order.

Second, we use a predetermined threshold (usually set at 0.5) to evaluate the CS scores. Pairs with scores below this threshold are marked as less redundant, simultaneously accumulating the number of pairs being marked as providing redundant information. We choose the threshold of 0.5 to measure sentence redundancy because it balances sensitivity and specificity. Sensitivity ensures that genuinely redundant sentences are detected, while specificity minimizes the risk of falsely categorizing unrelated sentences as redundant. In text summarization, it is common to expect some level of redundancy to improve the coherence and readability of the summary. A threshold of 0.5, therefore, allows moderate redundancy while ensuring that highly similar sentences are captured. Setting the threshold too low (e.g., below 0.2) might lead to the inclusion of dissimilar sentences, potentially affecting the summary's quality. Conversely, setting it too high (e.g., above 0.8) might result in the exclusion of genuinely redundant but slightly different sentences.

Finally, we obtain the redundancy score RDF by dividing the number of less redundant sentence pairs by the total number of unique sentence pairs (cf. Equation (16)). RDF has a scale from 0 to 1, with higher scores suggesting that the summary is less redundant.

We developed the algorithm (cf. Algorithm 1) that calculates the redundancy of a summary text.

Algorithm 1 Redundancy Analysis

Require: Summary
Ensure: Redundancy Score
Tokenize the summary into sentences
if number of sentences (n) = 1 **then**
  Set Redundancy Score (*redundancy_score*) to 1
else
  Apply TF-IDF vectorization to sentences
  Set similarity threshold to 0.5
  Initialize *less_redundant_count* to 0
  for $i = 1$ **to** number of sentences **do**
    for $j = i + 1$ **to** number of sentences **do**
      Calculate cosine similarity score *similarity_score* between sentence i and sentence j
      if *similarity_score* $<$ similarity threshold **then**
        Increment *less_redundant_count*
      end if
    end for
  end for
  Calculate total possible pairs: $total\_pairs = \frac{n(n-1)}{2}$
  Calculate Redundancy Score: $redundancy\_score = \frac{less\_redundant\_count}{total\_pairs}$
end if
return *redundancy_score*

The entire procedure comprises the following steps:

1. **Extracting Sentences:** The first step in the procedure is to break down the summary into individual sentences. Every sentence is considered as an independent piece of information.
2. **Calculating Cosine Similarity:** We use CS to obtain the semantic similarity of sentences, represented as vectorized embeddings.
3. **Identifying Redundant Pairs:** We determine the cosine similarity score for each pair of sentences in the summary. For this, we classify these sentences as less redundant, providing a score lower than a particular threshold (e.g., 0.5).
4. **Counting Less Redundant Pairs:** Based on CS, Algorithm 1 calculates how many pairs of sentences are marked as less redundant.
5. **Total Possible Pairs:** The number of total possible pairs of sentences is calculated, using Equation (15).

$$\text{TPP} = \frac{n(n-1)}{2}, \tag{15}$$

where n represents the total number of sentences in the summary. Note: In cases where the summary consists of just one sentence (n = 1), the redundancy score is set to 1.

6. **Calculating the Redundancy Score:** The number of less redundant sentence pairs is divided by the total number of sentence pairs to obtain the redundancy score. The RDF is presented in Equation (16).

$$\text{RDF}(\underline{Y}) = \frac{\text{Less Redundant Pairs}}{\text{Total Possible Pairs}} = \frac{\text{Number of Less Redundant Pairs}}{\frac{n(n-1)}{2}} \tag{16}$$

3.4. Bias Avoidance Analysis

BAA shows whether the summary introduces any subjective ideas, opinions, or partiality that are not yet present in the original content. It is important to keep a summary objective and neutral, refraining from introducing any bias. To estimate the possible influence of biases, we rely on Named Entity Recognition (NER) [24] and Jaccard Similarity (JS) [25] to evaluate the similarity of named entities between the original content and the summary.

For instance, let us consider an example where the original content comprises named entities such as "food", "sport", and "person". Now, in the summary text, if NER identifies additional entities like "food", "location", "person", and "company", it indicates that the summary has introduced some bias, for instance "location" and "company". This means that the summary has incorporated information or entities not found in the original content, potentially leading to a biased representation. BAA helps to ensure that summaries faithfully reflect the source material without undue influence or subjective elements.

In general, NER is used to extract named entities, for instance, names of people, organizations, locations, etc., from a given text. For an overview of recent techniques in the particular field, we refer interested readers to [26–28]. A brief introduction is given also in Section 3.4.1. In our approach, we utilized NER to extract named entities simultaneously from both, the original document and the summary. This provides us with a collection of entities (i.e., one list per text) being present in both texts, which needs to be compared to identify biases.

For this, JS is used, comparing the sets of named entities from the original material and the summary. This similarity score indicates how well the summary represents the named entities present in the original content. The main idea of the concept is introduced in Section 3.4.2.

Basically, the concept of JS is the backbone of the BAA assessment. Given this observation, the BAA score between the named entity sets $\mathcal{N}_{\underline{X}}$ and $\mathcal{N}_{\underline{Y}}$ can be calculated according to Equation (17)

$$\mathrm{BAA}(\underline{X}, \underline{Y}) = \frac{|\mathcal{N}_{\underline{X}} \cap \mathcal{N}_{\underline{Y}}|}{|\mathcal{N}_{\underline{X}} \cup \mathcal{N}_{\underline{Y}}|}, \tag{17}$$

where

- $\mathcal{N}_{\underline{X}}$ is the set of named entities taken from the original content,
- $\mathcal{N}_{\underline{Y}}$ is the set of named entities taken from the summary,
- $\cap$ represents the intersection of sets,
- $\cup$ represents the union of sets.

In the following, we consider two examples (with and without bias), discussing the characteristics of BAA.

Example 1 (with bias):

1. Original Content: "The new restaurant in town offers delicious Italian cuisine".
2. Summary: "The amazing new restaurant in town offers mouthwatering Italian cuisine that will leave you craving for more".

In this example, the summary introduces bias by using subjective terms like "amazing" and "mouthwatering", which were not present in the original content.

Example 2 (without bias):

1. Original Content: "The company reported its quarterly financial results".
2. Summary: "The firm disclosed its quarterly financial outcomes".

In the second example, the summary refrains from introducing any bias and maintains a neutral tone. The terms used in the summary are similar to those in the original content, and there are no subjective opinions or embellishments.

A high BAA score shows that the summary successfully avoids bias, which means that it correctly reflects the named entities found in the original content (cf. Example 2).

In other words, a higher score indicates a closer match between the specified entities in the summary and those in the original content. The BAA factor works better in extractive summarization than in abstractive summarization because abstractive summaries tend to contain more unique words. To overcome this limitation of the BAA factor, we can utilize additionally the SSF (cf. Section 3.1) as a kind of "counterpart".

3.4.1. Named Entity Recognition

NER is a NLP technique to process natural language that involves finding and categorizing important objects within a given text document, for instance, names of people, companies, places, and dates [24]. Understanding and extracting structured information from unstructured text data is a crucial function of NER. There are several applications that benefit from NER such as text summarization, information retrieval, and question-answering. Especially for the assessment of possible biases within a text, this method provides a powerful approach. To achieve the relations, NER involves the application of part-of-speech tagging and syntactic analysis, identifying and classifying sequences of words that correspond to specific named entity categories.

The process can be explained as follows: We consider a given text document D, containing a sequence of words,

$$\mathrm{D} = [\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_n], \tag{18}$$

where n is the total number of words. Based on D the following steps are processed:

- **Part-of-Speech Tagging** (POS): This method assigns a grammatical category to each word in a text, such as a noun, verb, or adjective. In order to figure out the grammatical role of a word within a sentence, POS is really important.
- **Syntactic Analysis:** Syntactic analysis is a method of analyzing sentence structure to find word connections and phrases. This helps in the identification of multi-word entities like "New York" (a place) and "John Smith" (a person).
- **Named Entity Classification:** NER classifies the recognized phrases into specified groups of named entities after POS tagging and syntactic analysis. These categories include a wide range of elements, including names of people and organizations, as well as places and dates.

3.4.2. Jaccard Similarity

In JS, the intersection of two nominal attributes is divided by their union to determine the similarity between them [29,30]. It is a widely used method to assess how similar two objects are, for instance, two text documents [25]. The basic equation is stated in Equation (19).

$$\mathrm{JS}(\mathcal{A}, \mathcal{B}) = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|} \tag{19}$$

where

- $\mathcal{A}$ and $\mathcal{B}$ represent the two sets being compared,
- $|\mathcal{A} \cap \mathcal{B}|$ represents the number of elements that are common to both sets (the intersection of sets),
- $|\mathcal{A} \cup \mathcal{B}|$ represents the total number of unique elements in both sets (the union of sets).

The JS score ranges from 0 to 1, with higher scores suggesting that the identified entities in the original content and the summary are more comparable.

4. Experiments and Results

This section mainly provides a detailed explanation and discussion of the experiments (in the sense of a proof-of-concept) using SUSWIR as well as the respective findings (cf. Sections 4.3 and 4.4). This is complemented by the description of the utilized data sets.

4.1. Datasets

For the manuscript's experiments, we used the following five datasets, which will be briefly introduced: CNN/Daily Mail Dataset [31], BBC Articles Dataset [32], SAMSum dataset [33], DialogSum dataset [34], and BillSum dataset [35]. Except for the BBC Articles Dataset, for all other corpora, the details of train, development, and test sets are given (cf. Table 2), if applicable, generally predefined by the dataset distributors. All datasets provide *human-generated summaries*, assumed to be at a high-level of quality, of the original documents, which will be used later for the assessment of SUSWIR.

Table 2. Collection of datasets used in the experiments. The table presents the respective number of samples per dataset, being split into training, development, and test sets. For BBC Articles, no predefined splitting is provided by the corpus distributor.

Dataset	Training	Development	Test
CNN/Daily Mail	287,226	13,368	11,490
BBC Articles	2225	–	–
BillSum	18,949	–	3269
SAMSum	14,732	818	819
DialogSum	12,460	500	1500

The CNN/Daily Mail dataset collects articles from both CNN and Daily Mail newspapers. Both newspaper publishers provide their news articles with bullet point summaries. These summaries contain usually multiple sentences. In our analysis, each news article was considered as original content, and its summary as the respective summary content. The combined dataset comprises articles from April 2007 for CNN and June 2010 for the Daily Mail, respectively, till the end of April 2015. Given this period, the validation data are from March 2015 and the test data are from April 2015. The distributor excluded articles with more than 2000 words as well as those articles where the summaries were not found. Therefore, the entire dataset contributes 287,226 training pairs, 13,368 validation pairs, and 11,490 testing pairs. We only used the test pairs in our experiments to evaluate the different methods (cf. Section 4.3).

BBC Articles Dataset has 2225 documents of BBC in the period of 2004 to 2005. This dataset contains news articles and summaries from five different categories including business, entertainment, politics, sport, and tech. We used the complete dataset for our experiment because the dataset provider did not furnish any predefined partitions.

BillSum is a summarization dataset that contains the texts of bills and human-written summaries of US Congressional and California state bills. The BillSum dataset is composed of three sections (names are taken from [35]): US training bills, US test bills, and California test bills. These bills were collected from the United States Government Publishing Office. This corpus includes bills from the 103rd to the 115th sessions of Congress, spanning the years 1993 to 2016. Due to the high variance in bill length, the distributors decided to focus on legislation of moderate length, ranging from 5000 to 20,000 characters. The choice to measure text length in characters, as opposed to words or sentences, is due to the complex structure of the texts. The dataset does not have short bills (less than 5000 characters). Intentionally, the length of summaries was restricted to 2000 characters [35]. Therefore, the dataset comprises 18,949 training pairs and 3269 test pairs. We considered only test pairs for our experiments.

SAMSum and DialogSum are two dialogue datasets consisting of conversations involving multiple participants. DialogSum dataset is a collection of conversations from four different sources: Dailydialog [36], DREAM [37], MuTual [38], and an English-speaking practice website. These conversations cover a wide range of everyday topics and typically involve interactions between friends, colleagues, as well as service providers and customers. The SAMSum dataset includes 16,000 online chat conversations and summaries. However,

it primarily focuses on short messenger app chats, which differ in language style and topics from typical spoken dialogues. In these datasets, the main text captures the dialogue itself, while the main summary provides a condensed version of that conversation. The average number of tokens or words per dialogue in the SAMSum dataset is 94, while the DialogSum dataset exhibits an average of 131 tokens or words per dialogue [34]. The SAMSum dataset contributes 14,732 data pairs for training, 819 data pairs for testing, and 818 data pairs for validation. In addition, the DialogSum dataset comprises 1500 test pairs. Again, we also only considered test pairs for our experiment.

4.2. Resources

In our work, we used specific tools and software to run the experiment smoothly. An overview is provided in Table 3.

All experiments were run on a personal computer using Windows 11. We chose Python, version 3.7.12 or higher, as our programming language, ensuring compatibility with the latest features. For machine learning tasks, we relied on `scikit-learn` [39], an easy-to-use library, and for NLP we used `nltk` (Natural Language Toolkit) [40].

Table 3. Overview of the hardware and software specifications used in the experiments.

Tool	**Version**
Operating System	Windows 11
CPU	Intel(R) Core(TM) i7-11800H @ 2.30 GHz
RAM	32 GB
Programming Language	Python $\geq$ 3.7.12
Libraries and Frameworks	1. scikit-learn version 1.0.2 2. nltk version 3.7

4.3. Experimental Settings

In our experiment, all the datasets, discussed in Section 4.1 and listed in Table 2, provide original contents along with human reference summaries, which is the typical setup for quality evaluation of machine-generated summaries. However, in our approach, we depart from this conventional evaluation paradigm. Instead, we assume a scenario where we do not have access to human reference summaries. For this, *we treat the human reference summaries* provided by the datasets *as if they were machine-generated summaries and consider the original content as our reference for comparison*. This procedure allows a fundamental assessment of our approach, by using high-quality summaries during the development phase. However, in the application, SUSWIR is then of course being used on automatically generated summaries. For this, as a second round of proof-of-concept, we also applied SUSWIR to automatically generated material, selecting only one corpus from Table 2. Details are presented in Section 4.3.4.

4.3.1. Data Pre-Processing

Regarding the utilized datasets (cf. Section 4.1), we see that each corpus still comprises details that are not relevant to the current experiments and thus, can be considered as noise from our perspective. Usually, text data contains different forms of "noise", for instance, punctuation, stop words, etc. Since we do not work with such kind of information, we decided to remove it from the raw data.

Another reason to clean the data is that we consider the original text (i.e., the articles in the datasets) as the reference text. In most cases, the original article is comparatively longer than the summary. So, basic data cleaning helped to remove unnecessary information and to reduce the overall length. We applied the following steps to clean the text before starting the experiments:

1. Case conversion (all lowercase).
2. Remove stop word.
3. Apply stemming.

4.3.2. Parameter Settings

As stated in Equation (1), SUSWIR is comprised of four factors, being individually weighted. In general, this allows a specific adaptation to assessment aspects (e.g., highlighting the semantic similarity of the two documents or redundancy-free summaries). The particular influence of the weights and factors was already discussed in Section 3 as well as throughout the specific factors introduction in Sections 3.1 to 3.4. In general, we recommend consistently taking into account individual factors and adapting the weights during the evaluation process. However, in the current manuscript's proof-of-concept, we do not opt for any adaptation and thus, decided to give equal weights ($W_i = 0.25$) to each of the four factors in the SUSWIR score, maintaining a total sum of weight of 1.

Since the proposed metric is newly developed, an appropriate comparison is necessary. For this, we compared our approach to well-known measures currently used in the community: standard ROUGE (focusing on F1-scores) and METEOR metric. To achieve comparable results, we relied also on known datasets (cf. Section 4.1), already used in multiple publications. Interestingly, such papers either refer to clean or raw datasets. In this sense, we, therefore, pre-processed the data accordingly as discussed in Section 4.3.1, resulting in two settings (cf. Table 4): (1) "CD" represents the clean data, being also used to calculate SUSWIR scores; (2) We used raw data (i.e., no pre-processing was applied) which is represented as "RD".

Since ROUGE computes n-grams or word sequence overlaps on a scale from 0 to 1 to determine the similarity between two texts, we tested three different variations. Rouge-1 (R1) calculates the number of overlaps of *uni-grams* between a reference text and its summary, Rouge-2 (R2) calculates the number of overlaps of *bi-grams*, and Rouge-L (RL) calculates the *longest common sequence* between a reference text and its summary [8].

We calculated ROUGE scores using the `rouge` Python package [41], and we further applied the `nltk` library [42] to estimate the METEOR score. To establish the ROUGE and METEOR metrics, we utilized the default parameter settings of the particular packages.

4.3.3. Statistical Significance

For further evaluation, we computed the Analysis of Variance (ANOVA) [43] to test the statistical significance of our results. ANOVA is a statistical tool, that compares means across several groups. It extends the t-test, which is used for two groups, to scenarios involving more than two groups. The main goal is to determine whether there are significant differences between the means of these groups by examining variations. ANOVA achieves this by comparing two independent estimates of the population variance. This helps to assess whether groups are significantly different from each other [43]. We set the significance level to $p < 0.05$. The p value in ANOVA is the probability of obtaining similar results to those actually observed, assuming no effect or no difference (null hypothesis). Therefore, we can state for our experiments, that if the p value is smaller than 0.05, there is a significant difference, and if p is larger than 0.05, no significant difference can be observed.

4.3.4. Automatically Generated Summaries

Automatic summaries or machine-generated summaries are generated by computer algorithms without human involvement. These summaries aim to capture the most important and relevant information from a (longer piece of) text, making it easier for readers to identify the key points quickly. In the first proof-of-concept for SUSWIR, we used material from different domains in five data sets (cf. Section 4.1 and Table 2), comprising human-generated summaries which were assessed against the original content. In the second round of conceptual proof, we selected the BBC Articles data set [32], which is the most convenient one in terms of computational time and human-based cross-check effort,

as the foundation for automatic summarization. To obtain a set of machine-generated summaries, we applied the T5 (Text-to-Text Transfer Transformer) model [44]. T5 is a transformer-based model, being designed to handle a wide range of NLP tasks like translation, summarization, question-answering, etc. with impressive accuracy. We used the hugging face library [45] to access a pre-trained T5 summarization model. The particular set of models comprises realizations in different sizes [46]: t5-small, t5-base, t5-large, t5-3b, and t5-11b. In our experiments, we used the t5-base because it offered faster inference, lower resource requirements, and decent performance. Smaller models like t5-small and t5-base have faster inference times, making them suitable for real-time or low-latency applications. They require less computational power, memory, and storage, making them more accessible to a broader range of users. Bigger models like t5-large, t5-3b, and t5-11b have high computational demands, requiring substantial computational resources, including GPUs or TPUs. In addition, their slower inference times can be impractical for real-time applications. As we tried to avoid any tuning effects (also not performed in the first proof-of-concept in Section 4.3.2), we applied default parameters, making only two changes: (1) minimum summary length was set to 80 words and (2) maximum length was set to 120 words. We did not apply any data pre-processing before generating summaries. This provided us with 2225 automatically generated summaries, respectively, corresponding to the original content.

However, we can assume that these summaries are of rather low quality compared to the corresponding human-generated version. This is in fact ideal since it allows an assessment of how the metrics score summaries of different quality levels originating from the same content (cf. Tables 4 and 5). Nevertheless, to ensure a level of quality in the automatically generated summaries, we manually compared a random selection of 100 samples to the corresponding high-quality human summaries. Subsequently, like the traditional approach, we also used ROUGE, METEOR, and SUSWIR to assess the quality of the machine-generated summaries by comparing them to human-generated summaries (cf. Table 6).

4.4. Results

In this section, we present the results of our experiments, mainly proofs-of-concepts, dividing the achievements in multiple aspects.

4.4.1. Human-Generated Summaries

Table 4 visualizes the achievements of our proof-of-concept utilizing the proposed metric SUSWIR. Regarding the respective values, we can first of all state that the proposed method generates higher score values compared to ROUGE and METEOR metrics, when checking the quality of the summary using CNN/Daily Mail, BillSum, SAMSum, and DialogSum datasets. However, this does not mean that SUSWIR is outperforming the other metrics; SUSWIR rather suggests an alternative metric, aiming for a reference-free assessment and a better interpretation of scores, using the entire range between 0 and 1. In this sense, a higher SUSWIR score represents a better alignment between the original text and the summary. As we pretend human-based reference summaries, provided by the datasets, as though they were machine-generated, we expect higher evaluation values from all three metrics. However, both ROUGE and METEOR metrics provide quite low evaluation values across the majority of the datasets. This results in a lower spectrum for interpretation, since we are not used to assessments being on the lower end of the 0 to 1 range. Further, currently, we are already using good quality human-generated summaries, leading to low scores in ROUGE and METEOR. However, we see similar low scores in the literature for summaries of less quality or being automatically generated. So, from our point of view, there is a mismatch of score values and quality, which was tackled by SUSWIR. Considering Table 4, SUSWIR handles this situation better, aiming to deliver more meaningful evaluation values.

Table 4. Experimental results for ROUGE, METEOR, and the proposed approach SUSWIR scores on CNN/Daily Mail test set, BBC Articles dataset, BillSum, SAMSum, and DialogSum test set. R1, R2, and RL indicate the length of the considered sequence (cf. Section 4.3.2). RD indicates raw data used in the experiments, and CD highlights pre-processed data. The numbers in bold font show the best score from the experiment across various metrics.

Dataset	ROUGE (RD)			ROUGE (CD)			METEOR		SUSWIR
	R1	R2	RL	R1	R2	RL	RD	CD	CD
CNN/Daily Mail	0.208	0.083	0.201	0.225	0.085	0.218	0.060	0.068	**0.427**
BBC Articles	**0.663**	0.593	**0.663**	0.639	0.563	0.639	0.346	0.369	0.651
BillSum	0.293	0.168	0.278	0.385	0.176	0.371	0.097	0.106	**0.482**
SAMSum	0.211	0.068	0.189	0.348	0.071	0.306	0.125	0.122	**0.510**
DialogSum	0.166	0.046	0.157	0.289	0.125	0.268	0.092	0.106	**0.491**

4.4.2. Machine-Generated Summaries

As already stated in Section 4.3.4, we ran a second round of proof-of-concept, considering automatically or machine-generated summaries on BBC Articles [32]. Applying the T5 text summarization model [46], we obtained summaries scored by the three aforementioned methods. Again, we used the original document as a reference for the assessment. Table 5 shows the achievements. We also employed the human summary as a reference (as being currently the standard approach) to evaluate the quality of the machine-generated summary (cf. Table 6).

Table 5. Comparing the outcomes using ROUGE, METEOR, and SUSWIR metrics for machine-generated summaries (using T5 summarization model) on the BBC Articles Dataset. The quality of machine-generated summaries is compared against the **original** contents. For the sake of easy comparison, we repeated the respective row from Table 4, showing the result for comparison of human-generated summaries against original contents. The numbers in bold font show the best score from the experiment across various metrics.

Machine Summaries (T5) Compared with	ROUGE (RD)			ROUGE (CD)			METEOR		SUSWIR
	R1	R2	RL	R1	R2	RL	RD	CD	CD
Original Documents	0.324	0.208	0.319	0.348	0.214	0.344	0.144	0.155	**0.460**
BBC Articles (Table 4)	**0.663**	0.593	**0.663**	0.639	0.563	0.639	0.346	0.369	0.651

We chose the BBC Articles dataset for this particular experiment because it provided higher scores compared to other datasets (cf. Table 4) in all three metrics. Table 5 shows that when we compared the machine-generated (T5 model generated) summaries with the original content, SUSWIR gives a relatively higher score (0.460) than ROUGE and METEOR. Comparing the results between Tables 4 and 5 on the BBC Articles dataset, we observe a decrease in all three metrics when using machine-generated summaries. This decrease results mainly from the fact that the learning model is not able to produce a high-quality summary in comparison to a human (cf. also Section 4.3.4). Table 5 also shows that both ROUGE and METEOR metrics yield lower scores (both clean and raw data) compared to SUSWIR.

For this, the following needs to be highlighted: The decrease does not necessarily indicate that the machine-generated summaries are poor or that ROUGE and METEOR provide bad results. The reason for this difference is related to the characteristics of the machine-generated summaries as well as ROUGE and METEOR metrics. Machine-generated summaries tend to have fewer words (depending on the parameter) in common with the original content and include more novel and diverse terms (T5 generates ab-

stractive summaries). Summary length also plays a role in the evaluation (in this case maximum summary length is 120 words). Shorter summaries naturally have fewer overlapping words, especially when compared to the reference text (as shown by the ROUGE-1 results). Conversely, longer summaries with more overlapping words can yield higher scores with ROUGE and METEOR metrics, *not necessarily indicating a high quality of summaries* (cf. Section 4.4.3). In contrast, SUSWIR handles these variations more effectively and provides more consistent results. In this sense, we can state that SUSWIR is not influence (future realizations of relevance estimation) or at least on a reduced level (cf. Section 3.2 and Equation (1)) on these aspects (compared to ROUGE or METEOR). We have illustrated this in a scenario in Section 4.4.3, showing the different results in Table 7.

Table 6. Comparing the outcomes using ROUGE, METEOR, and SUSWIR metrics for machine-generated summaries (using T5 summarization model) on the BBC Articles Dataset. The quality of machine-generated summaries is compared against the **human-generated** summaries (from the BBC Articles dataset). For the sake of easy comparison, we repeated the respective row from Table 4, showing the result for comparison of human-generated summaries against original contents. The numbers in bold font show the best score from the experiment across various metrics.

Machine Summaries (T5) Compared with	**ROUGE (RD)**			**ROUGE (CD)**			**METEOR**		**SUSWIR**
	R1	**R2**	**RL**	**R1**	**R2**	**RL**	**RD**	**CD**	**CD**
Human Summaries	0.375	0.219	0.363	0.377	0.210	0.367	0.235	0.226	**0.472**
BBC Articles (Table 4)	**0.663**	0.593	**0.663**	0.639	0.563	0.639	0.346	0.369	0.651

Additionally, we compared the machine-generated summaries with high-quality human-generated summaries (cf. Table 6), following the traditional approach of summary evaluation. In this case, SUSWIR also yielded a better evaluation score (0.472) compared to the other two metrics. To further maintain and assess the quality of automatically generated summaries, we manually evaluated 100 samples in comparison to the respective human-generated samples, and our observations indicated that the machine-generated summaries partially captured the main topic but not as effectively as human-generated summaries. However, the machine-generated summaries contain of course some information found in the human-generated summaries. In the majority of cases in our experiment, as observed, machine-generated summaries often employ novel, yet semantically equivalent words, to express the same information found in human-generated summaries. Consequently, this results in ROUGE metrics yielding lower evaluation scores compared to SUSWIR (cf. Table 6). In detail, both, human- and machine-generated summaries, do not provide the exact same text but convey similar ideas to varying degrees. One reason is that we (currently) did not adapt the summarization model to the BBC Articles. In this sense, our hypothesis might be that training the summarization model with the BBC Articles dataset will result in higher-quality machine-generated summaries, being closer to human-generated summaries in quality. This will be a matter of future research. Additionally, variations in summary lengths also contribute to differences in the summaries. Given this proof-of-concept, we apply our method to further data sets as well as fine-tuned summarization models to investigate SUSWIR characteristics in comparison to quality assessments using ROUGE and METEOR.

Exhaustive experiments in these cases will be a matter for future research, since in automatic text summarization as well as respective evaluation metrics multiple parameters influence the performance and the scoring results. This extends beyond the scope of a proof-of-concept.

4.4.3. Bias Issue

Besides the quality aspect of summaries, there is also the issue of biases in summaries (cf. Section 3.4). In this sense, in cases of biased machine-generated summaries (an abstrac-

tive approach or abstractive summary) both, ROUGE and METEOR, metrics face challenges to provide correct evaluation scores (cf. also discussion of limitations in Section 2). The experimental results may contain indications for this issue since we cannot guarantee bias-free summaries in the datasets.

For this, we further elaborate on this aspect, examining the performance of these three metrics using specific examples. Given the original content, we consider two possible scenarios: (1) a highly abstractive summary, introducing new words while still capturing the main idea from the original content; (2) an extractive summary that contains overlapping words, yet the overall idea did not align with the original content. In this sense, we based the example on a well-known phrase as follows:

- **Original Content:** "There is a brown fox. It is really quick. It jumped over the lazy dog. The dog is small in size".
- **Summary 1:** "A nimble, wood-coloured fox swiftly leaped above the lethargic and petite dog".
- **Summary 2:** "The quick brown fox jumped over the lazy fox".

From the above example, we can see that Summary 1 generates new words which also effectively captures the main idea of the original content. In contrast, Summary 2 contains overlapping words from the original content but fails to convey the central theme, lacking a clear representation of the main idea. Using ROUGE, METEOR, and SUSWIR to assess both summaries with the original document, we achieved the results presented in Table 7.

Table 7. Comparing the outcomes using ROUGE, METEOR, and SUSWIR metrics for Summary 1 and Summary 2 in relation to the original content from the example. Summary 1 reflects an abstractive summary, capturing the core idea of the original content, while Summary 2 serves as an extractive summary, lacking representation of the main idea. The numbers in bold font show the best score from the experiment across various metrics.

Metrics	Summary 1	Summary 2
ROUGE (RL)	0.206	0.639
METEOR	0.105	**0.248**
SUSWIR	**0.318**	0.411

Table 7 shows that all metrics yield lower results for Summary 1 (abstractive summary) compared to Summary 2 (extractive summary). This aspect is related to the fact that extractive summaries shows a higher degree of overlap with the reference text since (parts of) sentences are directly selected from the source. However, abstractive summaries, characterized by paraphrasing and restructuring, often result in more evaluation mismatches. The interpretation from Table 7, also indicates that when the summary is abstractive and effectively represents the main idea (Summary 1) from the original content, SUSWIR reflects this in a better (and interpretable) way compared to ROUGE and METEOR. Specifically, for Summary 1, SUSWIR yields a score of 0.318, while ROUGE suggests 0.206 and METEOR 0.105. This indicates that SUSWIR is more sensitive to the nuances of abstractive summaries and their alignments with the original content.

Conversely, when the summary involves word overlapping (Summary 2) but fails to convey the main idea of the original content, this is a challenging situation for the metrics. Given the overlapping character of this scenario, we see relatively "good" scores, although they should be rather low. While ROUGE and METEOR scores for Summary 2 are 0.639 and 0.248, respectively, SUSWIR delivered a comparatively lower score of 0.411. However, METEOR provided the lowest score compared to SUSWIR for Summary 2. From our perspective, there are three reasons for this aspect:

1. The METEOR score is influenced by the chunk penalty setting (cf. Equation (13)). Changing the chunk penalty can result in either a higher or lower METEOR score. This means that the choice of chunk penalty can substantially affect the final evaluation

outcome, highlighting its significance in the METEOR metric. In the experiments, we used the default chunk penalty provided by the respective library [42].
2. It is also important to consider that METEOR scores are often compressed within the range. This issue was already discussed in comparison to results in Table 4.
3. In the SUSWIR approach, we maintained equal weights for all four factors in the current experiments. Fine-tuning these weights according to the specific features of the summary, whether it is a one-sentence or a multi-line/multi-sentence summary, can result in enhanced performance and efficiency.

Given these considerations as well as the lowered score value, SUSWIR is capturing a nuanced relationship in this scenario. It further shows SUSWIR's ability to discern cases where a summary deviates from representing the central theme, reacting (just slightly) to overlapping words.

4.4.4. General Observations

Regarding the existing metrics results from Table 4, we showed the achievements for both ROUGE and METEOR metrics on clean and raw data conditions. Interestingly, applying commonly used tools for the calculation, we obtained quite low score values on the datasets, which are nevertheless, also seen in other papers. This is, from our perspective, an indicator that the current metrics are not covering the task characteristics suitably. In contrast, our approach maps the underlying intentions for a summary assessment quite well to a score value (cf. Table 4).

Comparing SUSWIR results to the conventional standards, we state that we obtained much better results, reflecting the underlying characteristics of the summaries. For the majority of datasets, SUSWIR scores are by far higher than ROUGE or METEOR values. Regarding Table 4 we see no huge differences in the scores with respect to the pre-processing conditions (clean vs. raw). For this, we directly compare our achievements to both settings on common metrics. An additional advantage of the novel metric is that a higher variety of quality levels can be represented (i.e., for the other metrics the scores are often "compressed" to values close to zero).

In contrast to most datasets, in our experiments, we achieved slightly better results on ROUGE-1 and ROUGE-L metrics (0.663) while using BBC Articles raw data compared to the SUSWIR approach (0.651). A particular reason could be that the summaries in the BBC Articles contain fewer unique words compared to other datasets. The ROUGE-1 score reinforces this observation since it specifically focuses on uni-gram matching and does not capture semantic nuances. In contrast, the SUSWIR approach can handle synonyms and words with similar meanings. Therefore, it provides almost similar results as ROUGE for the BBC Articles dataset. SUSWIR benefits from the underlying flexibility when it comes to creative or complex summaries, where different ways of expressing ideas matter.

However, to assess the findings on all datasets in further detail, we ran ANOVA tests to achieve statements on statistical significance. For this, we compared the SUSWIR and ROUGE-1 scores for clean data (CD) across all datasets. The null hypothesis assumed no significant difference between these scores. The obtained p-value ($p = 0.130$) is greater than the significance level ($p < 0.05$), resulting in no statistical significance. Hence, we can state the SUSWIR score can be seen as an additional (valid) score for all datasets. There was a statistically significant difference between SUSWIR and ROUGE-2 scores in clean data across the datasets ($p = 0.0143$), indicating that they provide differing assessments of text summarization quality. However, SUSWIR is again in favor since it provides a higher score value which can be used for interpretation argumentation. Finally, the ANOVA test found no statistically significant difference between SUSWIR and ROUGE-L scores ($p = 0.104$).

When comparing the score of SUSWIR and various ROUGE metrics, particularly in the context of BBC Articles, both for clean and raw data, the ANOVA test did not reveal any statistically significant differences. Specifically, for clean data, the p-value obtained was $p = 0.537$, and for raw data, it was $p = 0.830$. Again, we could argue that in such

cases the trimmed novel SUSWIR metric is to be selected, covering the characteristics of the intended task.

Additionally, a statistically significant difference was observed between SUSWIR and METEOR scores in clean data ($p = 0.0006$), suggesting that these metrics offer distinct perspectives on summarization quality. For METEOR, smaller score values were achieved.

4.5. Discussion

In our experiment, we evaluated the performance of two widely used automatic evaluation metrics for text summarization: ROUGE and METEOR and compared them to our approach SUSWIR. We conducted evaluation experiments across five distinct datasets (cf.S̃ection 4.1), known in the community, considering human-provided summaries (cf. respective statements in Section 4.3) as the machine-generated summaries and the original contents as reference texts. This setup simulated scenarios where only machine-generated summaries are available, without reference summaries for evaluation. Across the five datasets, we observed that ROUGE and METEOR metrics showed varying results. Notably, ROUGE (especially ROUGE-1 and ROUGE-L) demonstrated a strong performance on the BBC Articles dataset, achieving F1-scores of 0.663. We also examined the impact of using clean data (CD) versus raw data (RD) on the evaluation metrics. Overall, we found that using clean data consistently resulted in slightly higher ROUGE and METEOR scores compared to raw data across all datasets. This results from the reduced noise in the clean setting, where the data was pre-processed. In addition to ROUGE and METEOR, we introduced a novel evaluation metric, SUSWIR. SUSWIR showed highly promising results, outperforming both ROUGE and METEOR on most datasets, particularly on CNN/Daily Mail, BillSum, SAMSum, and DialogSum datasets. However, regarding BBC Articles, we found no statistical significance that ROUGE or METEOR score should be favored. Additionally, to assess performance, we compared SUSWIR, ROUGE, and METEOR metrics in various scenarios, including abstraction and extractive summaries (cf. Table 7). SUSWIR consistently delivered more reliable results, demonstrating its ability to grasp the semantic meaning and capture the core ideas of the content better than the other metrics. The same argument also holds true for the second proof-of-concept where automatically generated summaries were assessed. Again, SUSWIR provides reliable results (cf. Tables 5 and 6), allowing better and easier interpretation of the achieved scores.

Our main objective is to employ the original content as the basis for evaluating summaries. For this, the challenge of establishing references per document can be avoided. By analyzing the outcomes showcased in Table 4, we answer the research questions introduced in Section 2.

- **RQ1 Answer:** Our novel approach SUSWIR as well as the experimental results suggest that it is indeed possible to assess the quality of machine-generated summaries even in the absence of reference summaries. The utilization of the original content as a reference point provides a valuable alternative for evaluation. This finding is particularly significant in scenarios where reference summaries are unavailable.
- **RQ2 Answer:** Given the manuscript's observations, we can state that the method of using the original content as a point of reference works best in cases where the automatically generated summary closely matches the content's core ideas and keeps its semantic meaning. Therefore, the approach shows benefits, especially for the evaluation summaries of large or complicated materials. In these cases, the creation of human-crafted reference summaries is usually impractical or costly.
- **RQ3 Answer:** Given the experimental results (cf. Table 4), the proposed metric produces reliable and remarkable scores. Therefore, it provides a starting point for comparison, ensuring that summaries remain true and closely related to the original text.
- **RQ4 Answer:** Considering the current literature and our investigations, we state that a feasible metric for the assessment of summaries in a reference-free condition

needs to be based on four factors, namely Semantic Similarity, Redundancy Reduction, Relevance Assessment, and Bias Avoidance (cf. Section 3).

5. Conclusions

In this manuscript, we demonstrated an approach to evaluate the quality of machine-generated summaries in scenarios where reference summaries are unavailable. We used the original content as the reference text to evaluate summaries as it contains more information. Our proposed SUSWIR approach (introduced in Section 3) effectively addresses the limitations inherent in widely used summarization metrics such as ROUGE or METEOR. This approach employs four factors to check the quality of the summary. To evaluate our approach, we compared it experimentally to ROUGE and METEOR on five datasets (cf. Section 4.1). The empirical results of our study reveal the more reliable and superior performance of the SUSWIR approach in comparison to initial metrics (cf. Table 4 and Section 4.4), when human reference summaries are unavailable. Across the analysis of results from five datasets (cf. Table 4), the SUSWIR approach consistently surpasses both raw (RD) and pre-processed clean data (CD) in other evaluation metrics. In our experimental findings (see Table 4), the BBC Articles dataset stands out with impressive evaluations across all metrics, notably yielding a top-tier ROUGE (especially ROUGE-1 and ROUGE-L) score of 0.663. SUSWIR closely aligns with the performance of ROUGE on this dataset. Considering other datasets, SUSWIR consistently achieves notable results, including 0.482 on the BillSum dataset, 0.491 on the DialogSum dataset, 0.510 on the SAMSum dataset, and 0.427 on the CNN/DailyMail dataset. In contrast, the other two metrics struggle to generate reliable scores in these instances. SUSWIR also gives dependable results (0.460) when evaluating real machine-generated summaries created with the T5 summarization model [46], unlike ROUGE and METEOR metrics (cf. Table 5). These findings suggest that SUSWIR provides a better approach to analyze and assess machine-generated summaries, consistently providing reliable scores compared to ROUGE and METEOR. Similar results can be seen in exemplary testing of automatically generated summaries (cf. Table 5). This innovative method not only offers a valuable alternative for evaluating machine-generated summaries but also highlights the importance of using varied and informative reference texts when assessing summaries.

In future research, we extend our work to consider SUSWIR in raw data settings, where no data pre-processing is applied, and run respective adaptations, if necessary. This increases the flexibility and spectrum of usage of the developed metric.

Author Contributions: Conceptualization, A.A.F. and R.B.; methodology, A.A.F. and R.B.; software, A.A.F.; validation, A.A.F. and R.B.; writing—original draft preparation, A.A.F. and R.B.; writing—review and editing, R.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: CNN/Daily Mail Dataset: https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail; BBC Articles Dataset: https://www.kaggle.com/datasets/pariza/bbc-news-summary; BillSum dataset: https://www.kaggle.com/datasets/akornilo/billsum/; DialogSum dataset: https://github.com/cylnlp/DialogSum; SamSum Dataset: https://metatext.io/datasets/samsum.

Acknowledgments: We acknowledge support by Genie Enterprise and we also thank Siddarth Venkateswaran for valuable discussions on this research work.

Conflicts of Interest: Author Abdullah Al Foysal was employed by the company Genie Enterprise. The author Ronald Böck was employed by the company Genie Enterprise. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

ATS	Automatic Text Summarization
NLP	Natural Language Processing
TF-IDF	Term Frequency - Inverse Document Frequency
JS	Jaccard Similarity
CS	Cosine Similarity
NER	Named-entity recognition
LSA	Latent Semantic Analysis
SSF	Semantic Similarity Factor
RLF	Relevance Factor
RDF	Redundancy Factor
BAA	Bias Avoidance Analysis
BLEU	Bilingual Evaluation Understudy
METEOR	Metric for Evaluation of Translation with Explicit ORdering
CHRF	Character n-gram F-score
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
SUSWIR	Summary Score without Reference
ANOVA	Analysis of Variance
ROUGE-WE	ROUGE with Word Embeddings
MoverScore	Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance
T5	Text-to-Text Transfer Transformer

Nomenclature

$\underline{X}$ . . . represents raw text from original content
$\underline{Y}$ . . . represents raw text from summary
$\mathbf{x}$ . . . vector
$\mathbf{X}$ . . . matrix
x . . . scalar value
$\mathcal{A}$ . . . represents a set of elements

References

1. Saziyabegum, S.; Sajja, P.S. Literature Review on Extractive Text Summarization Approaches. *Int. J. Comput. Appl.* **2016**, *156*, 28–36. [CrossRef]
2. Nenkova, A.; McKeown, K. Others Automatic summarization. *Found. Trends® Inf. Retr.* **2011**, *5*, 103–233. [CrossRef]
3. Brin, S.; Page, L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.* **1998**, *30*, 107–117. [CrossRef]
4. Torres-Moreno, J. *Automatic Text Summarization*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
5. Iskender, N.; Polzehl, T.; Möller, S. Reliability of human evaluation for text summarization: Lessons learned and challenges ahead. In Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval), Kyiv, Ukraine, 19 April 2021; pp. 86–96.
6. Lloret, E.; Plaza, L.; Aker, A. The challenging task of summary evaluation: An overview. *Lang. Resour. Eval.* **2018**, *52*, 101–148. [CrossRef]
7. Vasilyev, O.; Bohannon, J. Is Human Scoring the Best Criteria for Summary Evaluation? *Find. Assoc. Comput. Linguist.* **2018**, *8*, 2184–2191.
8. Lin, C. Rouge: A package for automatic evaluation of summaries. In Proceedings of the Workshop on Text Summarization Branches Out, Barcelona, Spain, 25–26 July 2004; pp. 74–81.
9. Zhao, W.; Peyrard, M.; Liu, F.; Gao, Y.; Meyer, C.; Eger, S. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; Volume 11, pp. 563–578.
10. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
11. Banerjee, S.; Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; pp. 65–72.
12. Popović, M. chrF: Character n-gram F-score for automatic MT evaluation. In Proceedings of the Tenth Workshop on Statistical Machine Translation, Lisbon, Portugal, 17–18 September 2015; pp. 392–395.
13. Ng, J.; Abrecht, V. Better Summarization Evaluation with Word Embeddings for ROUGE. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1925–1930.
14. Colombo, P.; Clavel, C.; Piantanida, P. Infolm: A new metric to evaluate summarization & data2text generation. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington DC, USA, 7–14 February 2022; Volume 36, pp. 10554–10562.

15. Horn, N.; Gampfer, F.; Buchkremer, R. Latent Dirichlet allocation and t-distributed stochastic neighbor embedding enhance scientific reading comprehension of articles related to enterprise architecture. *AI* **2021**, *2*, 179–194. [CrossRef]
16. Al-Badarneh, A.; Al-Alaj, A.; Mahafzah, B. Multi Small Index (MSI): A spatial indexing structure. *J. Inf. Sci.* **2013**, *39*, 643–660. [CrossRef]
17. Foltz, P. Latent semantic analysis for text-based research. *Behav. Res. Methods Instrum. Comput*. **1996**, *28*, 197–202. [CrossRef]
18. Lavin, M. *Analyzing Documents with TF-IDF*; Programming Historian; University of Sussex: Brighton, UK, 2019.
19. Günther, F.; Dudschig, C.; Kaup, B. Latent semantic analysis cosines as a cognitive similarity measure: Evidence from priming studies. *Q. J. Exp. Psychol.* **2016**, *96*, 626–653. [CrossRef] [PubMed]
20. Uzhga-Rebrov, O.; Kuleshova, G. Using Singular Value Decomposition to Reduce Dimensionality of Initial Data Set. In Proceedings of the 2020 61st International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), Piscataway, NJ, USA, 15–16 October 2020; pp. 1–4.
21. Zahrotun, L. Comparison jaccard similarity, cosine similarity and combined both of the data clustering with shared nearest neighbor method. *Comput. Eng. Appl. J*. **2016**, *5*, 11–18. [CrossRef]
22. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
23. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3982–3992.
24. Roy, A. Recent trends in named entity recognition (ner). *arXiv* **2021**, arXiv:2101.11420.
25. Niwattanakul, S.; Singthongchai, J.; Naenudorn, E.; Wanapu, S. Using of Jaccard coefficient for keywords similarity. In Proceedings of the International Multiconference of Engineers and Computer Scientists, Hong Kong, China, 13–15 March 2013; Volume 1, pp. 380–384.
26. Jehangir, B.; Radhakrishnan, S.; Agarwal, R. A survey on Named Entity Recognition—datasets, tools, and methodologies. *Nat. Lang. Process. J*. **2023**, *3*, 100017. [CrossRef]
27. Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 50–70. [CrossRef]
28. Bose, P.; Srinivasan, S.; Sleeman, W.C., IV; Palta, J.; Kapoor, R.; Ghosh, P. A survey on recent named entity recognition and relationship extraction techniques on clinical texts. *Appl. Sci.* **2021**, *11*, 8319. [CrossRef]
29. Jaccard, P. Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bull. Soc. Vaudoise Sci. Nat*. **1901**, *34*, 241–272.
30. Bouchard, M.; Jousselme, A.; Doré, P. A proof for the positive definiteness of the Jaccard index matrix. *Int. J. Approx. Reason.* **2013**, *54*, 615–626. [CrossRef]
31. Hermann, K.; Kočiský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; Blunsom, P. Teaching Machines to Read and Comprehend. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 1, pp. 1693–1701.
32. SHARIF, P. BBC News Summary. Available online: https://www.kaggle.com/datasets/pariza/bbc-news-summary (accessed on 13 September 2023).
33. Gliwa, B.; Mochol, I.; Biesek, M.; Wawer, A. SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. In Proceedings of the 2nd Workshop on New Frontiers in Summarization,Hong Kong, China, 4 November 2019; pp. 70–79.
34. Chen, Y.; Liu, Y.; Chen, L.; Zhang, Y. DialogSum: A Real-Life Scenario Dialogue Summarization Dataset. In Proceedings of the Findings of The Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 5062–5074.
35. Kornilova, A.; Eidelman, V. BillSum: A Corpus for Automatic Summarization of US Legislation. In Proceedings of the 2nd Workshop on New Frontiers in Summarization,Hong Kong, China, 4 November 2019; pp. 48–56.
36. Li, Y.; Su, H.; Shen, X.; Li, W.; Cao, Z.; Niu, S. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers),Taipei, Taiwan, 27 November–1 December 2017; pp. 986–995.
37. Sun, K.; Yu, D.; Chen, J.; Yu, D.; Choi, Y.; Cardie, C. Dream: A challenge data set and models for dialogue-based reading comprehension. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 217–231. [CrossRef]
38. Cui, L.; Wu, Y.; Liu, S.; Zhang, Y.; Zhou, M. MuTual: A Dataset for Multi-Turn Dialogue Reasoning. In Proceedings of the 58th Annual Meeting of The Association for Computational Linguistics,Online, 5–10 July 2020; pp. 1406–1416.
39. Sckit-Learn Documentation. Available online: https://scikit-learn.org/stable/ (accessed on 19 October 2023).
40. NLTK Team. NLTK Documentation. Available online: https://www.nltk.org/ (accessed on 19 October 2023).
41. Pltrdy. A Full Python Library for the ROUGE Metric. Available online: https://pypi.org/project/rouge/ (accessed on 20 September 2023).
42. NLTK Team. NLTK-METEOR Documentation. Available online: https://www.nltk.org/api/nltk.translate.meteor_score.html (accessed on 20 September 2023).
43. Ostertagová, E.; Ostertag, O. Methodology and Application of One-way ANOVA. *Am. J. Mech. Eng.* **2013**, *1*, 256–261.

44. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv* **2019**, arXiv:1910.10683.
45. Hugging Face. Available online: https://huggingface.co/ (accessed on 26 October 2023).
46. Hugging Face, T5 Model. Available online: https://huggingface.co/docs/transformers/model_doc/t5#training (accessed on 26 October 2023).