New Evaluation Metric for Large Language Models in Summarization

D.K.W.S.Dilshan

2025

New Evaluation Metric for Large Language Models in Summarization

A dissertation submitted for the Degree of Master of Business Analytics

D.K.W.S.Dilshan
University of Colombo School of Computing
2025

Declaration

Name of the student: D.K.W.S.Dilshan
Registration number: 2022/BA/004
Name of the Degree Programme: Master of Business Analytics
Project/Thesis title: New Evaluation Metric for Large Language Models (LLMs) in Summarization

1. The project/thesis is my original work and has not been submitted previously for a degree at this or any other University/Institute. To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

2. I understand what plagiarism is, the various types of plagiarism, how to avoid it, what my resources are, who can help me if I am unsure about a research or plagiarism issue, as well as what the consequences are at University of Colombo School of Computing (UCSC) for plagiarism.

3. I understand that ignorance is not an excuse for plagiarism and that I am responsible for clarifying, asking questions and utilizing all available resources in order to educate myself and prevent myself from plagiarizing.

4. I am also aware of the dangers of using online plagiarism checkers and sites that offer essays for sale. I understand that if I use these resources, I am solely responsible for the consequences of my actions.

5. I assure that any work I submit with my name on it will reflect my own ideas and effort. I will properly cite all material that is not my own.

6. I understand that there is no acceptable excuse for committing plagiarism and that doing so is a violation of the Student Code of Conduct.

Signature of the Student	**Date (DD/MM/YYYY)**
	12/06/2025

Certified by Supervisor(s)

This is to certify that this project/thesis is based on the work of the above-mentioned student under my/our supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

	Supervisor 1	**Supervisor 2**	**Supervisor 3**
Name	Dr. Thilini Nadungodage		
Signature			
Date	12/06/2025		

I would like to dedicate this thesis to my supervisor and family.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to all those who provided support to make my research on “New Evaluation Metric for Large Language Models (LLMs) in Summarization” successful.

First, I would like to express my gratitude towards the project supervisor, Dr. Thilini Nadungodage, Senior Lecturer, Department of Computer Science, University of Colombo School of Computing. I am highly indebted to her for her guidance and constant supervision as well as for providing necessary information regarding the project and for her support in completing this project successfully. Her motivation and guidance have inspired me throughout this journey.

Finally, Last but not least, I should be sincerely thankful for my family and friends specially who have helped in various way to get this research completed successfully.

ABSTRACT

This thesis introduces and evaluates a new text summarization evaluation metric, SynSemScore, to fill the gap between the simple lexical-based metrics such as BLEU, ROUGE, and METEOR, and the complex embedding-based methods. While the previous methods are based mainly on surface level word matching or are costly in terms of computational complexity, SynSemScore lies in between by using syntactic matching together with semantic similarity measured using WordNet-like structures and a redundancy penalty to identify redundant content. The purpose is to offer a simple but effective way of measuring the quality of the machine produced summaries in terms of coherence, consistency, fluency and relevance.

To validate SynSemScore empirically, this research employs the SummEval dataset which offers a variety of human annotated summaries for CNN/Daily Mail articles. The data is cleaned up thoroughly to ensure data integrity and correlation analyses are made between SynSemScore and other metrics. The results of the experiments reveal that SynSemScore is more relevant to the human rating, outperforming traditional approaches in several aspects. Hence, the proposed metric can be considered as a feasible solution for assessing the quality of summarization models on a large scale without the need for expensive training of transformer-based models.

Besides the technical aspects of SynSemScore, the work describes the complete process of the research, including data collection and preparation, hyperparameter optimization, and correlation analysis of the metric. Finally, SynSemSemScore presents a flexible, easy-to-interpret, and semantic-aware method for assessing the performance of current summarization systems and can therefore help to enhance the quality of the generated text in both academic and business applications.

Keywords: Text Summarization, SynSemScore, Semantic Similarity, Redundancy Penalty, SummEval, Evaluation Metrics, Large Language Models.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

LLM	Large Language Models
AI	Artificial Intelligence
ML	Machine Learning
BLEU	Bilingual Evaluation Understudy
MT	Machine Translation
CNN	Convolutional Neural Network
SummaC	Summary-level Consistency Checker
CV	Cross-Validation
EDA	Exploratory Data Analysis
IDF	Inverse Document Frequency
NLI	Natural Language Inference
MAE	Mean Absolute Error
METEOR	Metric for Evaluation of Translation with Explicit ORdering
NLP	Natural Language Processing
RMSE	Root Mean Squared Error
BERT	Bidirectional Encoder Representations from Transformers
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
ROUGE-1	ROUGE Unigram
ROUGE-2	ROUGE Bigram
ROUGE-L	ROUGE Longest Common Subsequence
ACL	Association for Computational Linguistics
STS	Semantic Textual Similarity
SynSemScore	Syntactic-Semantic Score
TF	Term Frequency
WUP	Wu-Palmer Similarity
ATS	Abstractive text summarization
DUC	Document Understanding Conference
GloVe	Global Vectors for Word Representation

CHAPTER 1: INTRODUCTION

Large language models, or LLMs, are now a growing choice not only in the business sector but also in the academic field as the best-performing types of artificial intelligence (AI). Both researchers and major corporations have undoubtedly reached an impressive level of success in the past period (Bommasani et al., 2022). One of the groundbreaking advancements was the development of a highly sophisticated chatbot called ChatGPT, which went viral with millions of visitors across web forms and social media networks (Ray, 2023). Generally, the technological revolution that has taken place within the field of LLMs is so significant that it is very likely to completely change the way AI algorithms are constructed and applied. Instead of the previous models that were able to conduct separate commands solely, the LLMs have the ability to do a plethora of tasks and even more at the same time.

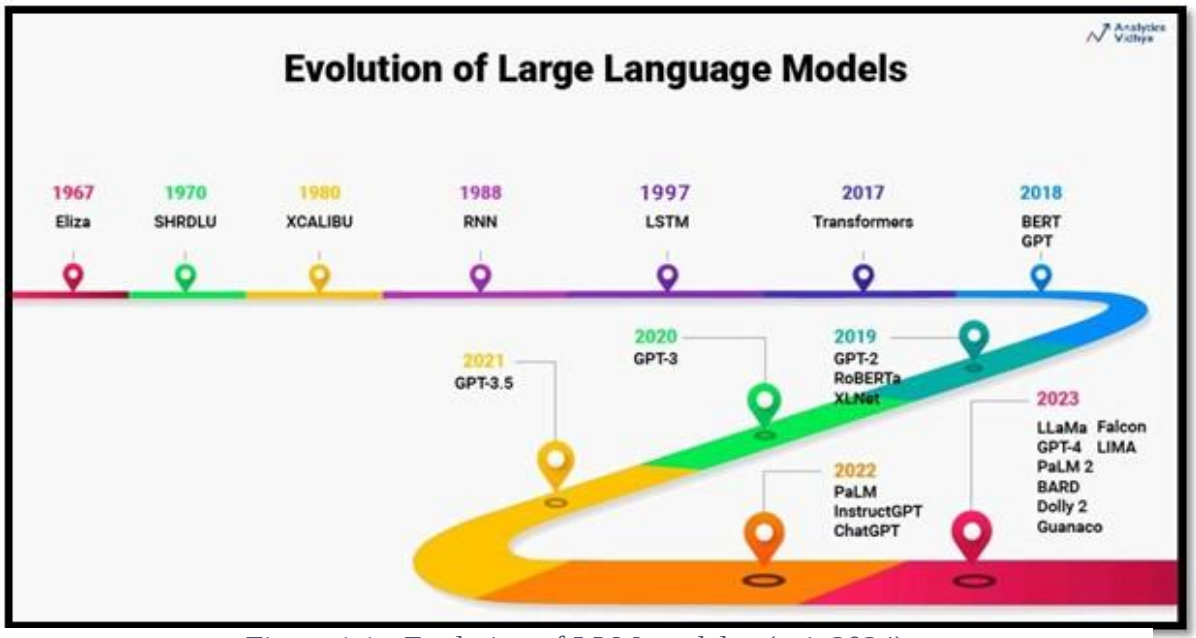


Figure 1.1 - Evolution of LLM models (pai, 2024)

In Figure 1.1 we can see the current evolution of the large language models. People with crucial information needs, including Patients and students are using LLMs with increasing frequency, due to their express proficiency with a range of applications, from general to domain-specific natural language tasks. However, there are several drawbacks of LLMs that hinder their optimal capability. These challenges include, but are not limited to, the lack of real-time knowledge updates, a lack of genuine emotion and thought, and verbose, protracted answers.

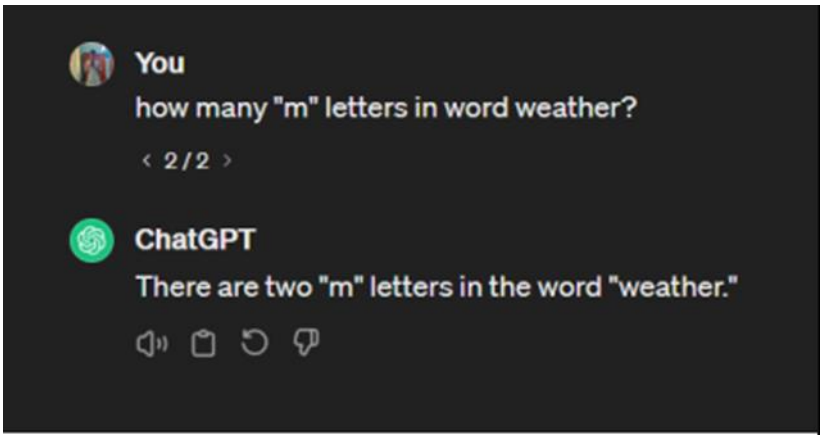


Figure 1.2 - hallucinations

Notably, one of the most significant flaws is that the generated text contains factual errors, which might lead to "hallucinations" (Ji et al., 2023) as depicted in Figure 1.2.

Hallucinations in LLMs refer to responses that contradict human knowledge and factual information. While humans normally make truthful claims (as in, they describe the world accurately), LLMs are free to create such fiction. More formally, a hallucination is when the model generates output that does not match the reality set. In other words, when an LLM produces an answer with facts that do not exist, it is called a hallucination. (Yao et al., 2024)

It's critical to understand the important factors and standards in LLM evaluation in order to address these problems. Since LLMs are still significant for research and everyday life, it is becoming more important to evaluate them at both the task and social levels in order to better understand the risk factors associated with them. Since the launch of ChatGPT, many studies are carried out in order to assess ChatGPT in comparison with other LLMs from various of topics including natural language tasks, reasoning, robustness, reliability, healthcare applications, legal and ethical issues. (Ray, 2023)

Empirical studies have shown that human judgment and reliable evaluation have been challenges that the existing automated evaluation metrics have not been able to solve. Furthermore, the methods that are ideally suited to the evaluation of the models are a subject of contention. Even though human judgment is typically viewed as the best method of evaluation (Guo et al., 2023), it usually has its downsides, such as difficulties in the reproducibility of the results. Nevertheless, evaluating human judgments is still one of the best tools at our disposal for determining the different biases and limitations associated with the automated evaluation metrics.

The automatic text summarization has risen to be an important area in the natural language processing (NLP) research domain due to the rapidly increasing amount of text data. These algorithms are capable of processing large volumes of data and extracting pieces of essential information apart from the textbook storage nature which reportedly carries further applications in research paper aggregation, and web content recommendation. Evaluation metrics are crucial in determining if machine-generated summaries display enough information and are coherent and readable. In many cases, the evaluations of the computer-generated summaries stress coherence, conciseness, readability, correct grammar, and content coverage as their main parameters.(Mani, 2001). However, human evaluations can become time-consuming, costly, and sometimes even unmanageable to stretch, which resulted in the urgent need for developing automated evaluation metrics not only in terms of efficiency but also in terms of consistency.

While the domain of Large Language Models (LLMs) is moving rapidly ahead of others, automatic summarization of texts has found its place as the primary task within NLP. These machines create brief and logical summaries from lengthy texts thus improving the efficiency and comprehension of the original materials. In general cleaning, the evaluation of the substance has been one of the most problematic things. Traditional measuring tools such as BLEU (Bilingual Evaluation Understudy)(Papineni et al., 2001), ROUGE (Recall-Oriented Understudy for Gisting Evaluation)(Lin, 2004) and METEOR (Metric for Evaluation of Translation with Explicit ORdering) (Banerjee and Lavie, 2005) , have frequently been used for analyzing the quality of summaries. Even though these metrics are effective, there are still certain limitations, especially when it comes to the semantics of the text, factual correctness, as well as the integrity of the text.

1.1 Motivation

There is a growing need for strong evaluation metrics that have closer alignment to human judgment, given the heavy use of automatic summarization techniques in many industries and research fields. Existing metrics are widely used, but fail to capture semantic meaning and structure, leading to discrepancies when compared to human rating. These limitations present a major obstacle towards building better-quality summarization systems. In addition, many of these practical tasks such as, generating news summary, medical reports, legal documents require high accuracy and coherence in generated summaries.

In contrast, using only approximated textual similarity as an evaluation metric can lead to misleading results and is bound to hamper future NLP-based research in the summarization

domain. Thus, the driving force behind this work is the design of SynSemScore, a new evaluation metric that promises to lie on the borderline between syntactic and semantic evaluation.

SynSemScore: an n-gram based summarization evaluation metric trained on redundancy penalties and WordNet based semantic similarity techniques. This study aims to advance the field of summarization evaluation by introducing new metrics that can help develop more effective summarization algorithms.

1.2 Statement of the problem

Despite new evaluation metrics for text summarization, so far, most of them are based on the lexical and syntactic matching metrics and have failed to ensure a semantic understanding (Dasgupta et al., 2024). For example, while BLEU and ROUGE rely largely on word and phrase overlaps, they don’t account for synonyms or contextual similarities at all. Because METEOR employs stemming and uses WordNet to match synonyms, it does identify some of these deeper matches, but it also doesn't have a way to do meaningful deep semantic alignment.

Moreover, since the evaluation metrics based on Transformer models take in high dimensional data, they are computationally expensive and require access to large amounts of memory and processing power, especially for large-scale models. Their high cost prevents them from being utilized as rapid tests or large-volume assessments (Nguyen et al., 2024). SynSemScore, on the other hand, is less computationally intensive, allowing for more efficient and scalable implementation.

In contrast, statistical-based metrics offer some benefits. They are lightweight and cheap to compute relative to model-based metrics, which often rely on deep neural networks. Since they only require simple, rudimentary components, they can be useful for fast testing, iterative building, and work in resource-constrained environments (Mastropaolo et al., 2023).In addition, the statistical metrics can be without deep learning or pre-trained embeddings frameworks which are demanding for intensive data and GPU computation, and therefore can be computed almost any environment (Liu et al., 2023).

The inconsistencies found in the traditional evaluation metrics as opposed to the assessments given by humans emphasize the necessity of a unique metric that efficiently connects these two. A novel evaluation metric should involve both syntactic correctness and semantic

consistency but should also assume that redundancy would not be the case. In this sense, this research proposes SynSemScore, a new summarization evaluation metric, which combines the syntax and semantic modules and aligns them better with human judgement.

1.3 Research Aims and Objectives

1.3.1 Aim

The objective of this research is to create and authenticate a novel metric for evaluation of text summarization that is built on a fusion of syntactic and semantic components, thus, achieving a higher correlation with human judgement. SynSemScore targets the shortcomings of current evaluation metrics (like BLEU, ROUGE, and METEOR) by giving a computationally inexpensive and, therefore, more generalizable approach for the assessment of machine generated texts.

1.3.2 Objectives

The main objectives of this research are:

1. To propose a new metric, SynSemScore, that combines syntactic and semantic analysis of the quality of the summaries generated by the machine.
2. Perform an in-depth analysis of current methods for evaluating LLMs, including their advantages, disadvantages, and potential areas of development.
3. To validate the effectiveness of SynSemScore by measuring its correlation with human judgment in evaluating text summaries.
4. To compare the performance of SynSemScore with traditional evaluation metrics, such as BLEU, ROUGE, and METEOR, using standard benchmark dataset.
5. Share the research results with the wider scientific community in order to improve LLM evaluation and increase knowledge of these robust models.

1.1 Scope

In order to provide an accurate and comprehensive study on text summarization metrics for assessment, it is essential that the scope of this research be clearly defined. This study focuses on the construction and validation of SynSemScore, a new metric specifically aimed at evaluating syntactic and semantic features of machine-generated summaries. By comparing SynSemScore with conventional evaluation measures such as BLEU, ROUGE, and METEOR, this study shows SynSemScore's significant alignment with human judgment improvement. It uses SynSemScore for assessing summaries generated by extractive and abstractive summarization models. To verify the performance of SynSemScore, this study further computes the correlations between the scores produced by SynSemScore and those from human evaluations. This paper thus limits its scope to such aspects, guaranteeing a relevant and in-depth discussion of the proposed evaluation metric and avoiding unnecessarily abstract and shallow explorations.

1.2 Structure of the Thesis

Table 1 provides an overview of what each chapter encompasses, the specific content included in each chapter, as well as the overview of the chapter-wise objectives containing hypotheses regarding the research. Such a systematic format facilitates better understanding of research problem, methodology and outcomes.

Chapter	Content	Description
Chapter 1	**Introduction**	Describes the motivation and background for the research, the problem statement, objectives, scope and significance of study.
Chapter 2	**Literature Review**	A review of existing work related to text summarization evaluation scores, and existing evaluation scores like BLEU, ROUGE and METEOR, and their need for better evaluation methodologies like SynSemScore.
Chapter	**Methodology**	Describes how SynSemScore was designed and implemented along with syntactic and semantic analysis

3		techniques, analyses for computational efficiency, and dataset preparation.
Chapter 4	**Evaluation and Results**	Details experimental setup and shows how the SynSemScore metric compares to existing metrics, as well as computational performance and correlation with position.
Chapter 5	**Conclusion and Future Work**	Provide an overview of key findings, highlight the importance of SynSemScore for summarization evaluation methods and major topics for future research.

Table 1.1 - Structure of the Thesis

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

As more industries rely on automatic text summarization, whether it be for news aggregation, legal text processing or medical report summarization, it is key to assess the quality of machine-generated summaries. In the field of Natural Language Processing (NLP), researchers commonly employ automatic scoring techniques to determine how closely these summaries match reference human-written summaries. Current accepting evaluation metrics such as prevalent BLEU (Papineni et al., 2001) ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005) assess only exact matching texts between generated sentences and referenced ones, thus it is error-prone to generate reasonable scale representations with those, missing a finer granularity as of deeper semantic and coherent relations. But the downside is that these methods can sometimes yield scores that do not entirely match human judgment, demonstrating the need for more trustworthy evaluation practices.

Recent progress in Large Language Models (LLMs) like GPT-4, BART, and Pegasus (Zhang et al., 2020a) has greatly improved abstractive summary generation quality. However, older n-gram-based evaluation metrics seem to be unable to evaluate such models properly since they offer no semantics, thus not taking reworded phrases, synonyms, or factual consistency into account. In response to this, researchers have investigated different evaluation approaches, including embedding-based metrics like BERTScore (Zhang et al., 2020a), MoverScore (Zhao et al., 2019), and SummaC (Laban et al., 2022). These approaches use contextualized word representations instead of exact word matches to determine meaning. Although they represent a step forward in comparison to classical approaches, these methods have a high computational cost and still struggle at detecting redundancy or incoherence in summary outputs.

To address the above gap in existing evaluation measures, this paper presents SynSemScore a measure that combines syntactic similarity (using ROUGE-L), semantic overlap (using matching via WordNet) and redundancy detection to derive a scoring that aligns more closely with humans scoring of text summaries. Unlike existing methods, SynSemScore′s algorithm is designed to prevent redundancy in the response that can plague automated summaries; moreover, it runs efficiently compared to transformer-based evaluation models. This allows for a more balanced and scalable summary quality measurement procedure that integrates both syntactic and semantic views. Next, we will look more closely at current evaluation

metrics their advantages and disadvantages. The conversation will also address human evaluation methods and the challenge of getting automated scoring to correspond with human judgments. Lastly, we will discuss the motivation for SynSemScore and the advantages it can offer relative to previous evaluation models.

2.2 Evaluation on LLMS through a focus on metrics

In the review by Hu and Zhou (2024), they discussed comprehensively the metrics which are used to evaluate large language models (LLMs). Its goal is to shed light on the mathematical expressions and statistical interpretations of existing metrics used for LLM evaluation with particular relevance to biomedical use cases. The authors aspired to clarify the mathematical definitions and statistical interpretations of the existing metrics especially to the biomedical contexts. The identified metrics are grouped into three main types: multi-classification (MC), token-similarity (TS), and question-answering (QA). Furthermore, they examine the application of these metrics in practice and provide a comparative analysis that can help researchers choose the best-fitted metrics for different tasks. Also, the study points out the available repositories for these metrics and their relevance to the new-developed biomedical LLMs(Hu and Zhou, 2024).

The metrics are classified into three primary types:

1. **Multi-Classification Metrics**: These metrics assess the performance of LLMs on tasks with non-binary labels, such as Accuracy, Recall, Precision, F1 Score, micro-F1, and macro-F1.

2. **Token-Similarity Metrics**: These metrics evaluate the similarity of generated text and reference text by applying measures, including Perplexity, BLEU, ROUGE, METEOR, and BertScore.

3. **Question-Answering Metrics**: These metrics measure specifically the performance in question-answering tasks usually including fuzzy matching metrics like Strict Accuracy, Lenient Accuracy, and Mean Reciprocal Rank.

The methodology for the evaluation of LLMs focuses on the actual performance of the operations taking into account such elements as the action space, application areas, and coverage in the domain of LLMs. Considering that the primary objective of LLMs (for example, GPT-3, BERT, T5, etc.) is to process long textual inputs so as to create answers that

resemble those of a human, assessing these models appears crucial for making sure that they are safe, fair, effective, and are aligned with the expectations of the humans. In order to reflect the current evaluation methods in regard to the measurement of LLM performance, the discussion would embark on some of the regularly convened techniques of evaluation.

BLEU (Bilingual Evaluation Understudy), is an automated metric for the evaluation of machine translation (MT) systems (Papineni et al. 2001). Its primary aim is to increase the speed, save costs, and become more consistent with the human evaluation, which is often contradictory, labor-consuming, and expensive. BLEU gives the score of translation of the machine based on the number of overlapping n-grams (units of text) with human references, which depends heavily on the human judge's insight into translation accuracy. Specifically, BLEU finds the modified n-gram precision score which reflects the percentage of candidate translation n-grams found in the reference translations. The importance of multiple reference translations is emphasized to display human variability and a brevity penalty is introduced to avoid excessively short translations that might achieve high scores merely on precision. The overall BLEU score is formed by the geometric mean of modified precision scores obtained across various n-grams and is combined with brevity penalty.

Papineni et al. (2001) proved that BLEU corresponds very well with the estimates by humans in many translation systems, effectively placing them in the same order as the ones produced by humans. The authors point out that BLEU is a strong performer in translation tasks as it can be applied across various languages and easily incorporated into MT development cycles. In spite of the fact that BLEU has certain shortcomings at the sentence level, it still provides trustworthy results across large corpora and under the right preprocessing conditions which makes it the best choice for large-scale comparative research of MT systems.

ROUGE, (Lin, 2004) which is short for Recall-Oriented Understudy for Gisting Evaluation, offers several scoring measures to evaluate the quality of summaries against a set of reference summaries produced by humans. The main ROUGE measures are ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S, which deal with n-gram frequency matching, longest common subsequence, weighted longest common subsequence and skip-bigram co-occurrence, respectively. Such metrics have been commonly used in large scale evaluations such as the Document Understanding Conference (DUC).

ROUGE metrics assess summaries by comparing their similarity to reference summaries using various techniques:

1. **ROUGE-N:** Calculates n-gram overlap between candidate and reference summary. It is recall oriented and also focused on how much of the reference summary is covered in the candidate summary.

2. **ROUGE-L:** This metric is based on the longest common subsequence (LCS) and measures the longest subsequence of words that appear in both the candidate and reference summaries in the same order. It highlights the structural commonalities between summaries.

3. **ROUGE-W:** A weighted version of ROUGE-L, this metric weights longer, consecutive matches more heavily, which encourages summaries that preserve the contiguous text in the reference.

4. **ROUGE-S:** Focuses on skip-bigram co-occurrence, allowing word pairs to match even if there are intervening words between them. This measure is particularly useful for capturing key phrases that might not appear contiguously.

To evaluate with multiple references, ROUGE employs a jackknifing method that prevents any one combination of reference summaries from dominating the score, with summary quality obtained by averaging scores across different combinations of reference summaries.

ROUGE was evaluated using data from DUC 2001, 2002, and 2003. The results demonstrate that ROUGE metrics, particularly ROUGE-2 (bigram overlap) and ROUGE-L (longest common subsequence), show a strong correlation with human judgments across various summarization tasks. The use of multiple references and exclusion of stop words generally improved the performance of ROUGE metrics. Besides these metrics like ROUGE performed well on the single-document summarization task and towards the very short summaries (headline-like) as well it showed some promise although correlation was a bit low for multi-document summarization tasks due to the number of samples.

Banerjee and Lavie (2005) have introduced METEOR, a metric aims to solve that and other drawbacks of existing automatic metrics (e.g. BLEU) by refining term for term matching to enhance the correlation with human judgments. The core idea is to consider generalized unigram matching, such as surface form, stemmed form and synonym matches for more specialized level of translation evaluation.

The methodology is centered around computing a score based on word-to-word alignments between the MT output and one or more reference translations. METEOR evaluates these

alignments using unigram precision, unigram recall, and a penalty for word order fragmentation. The score calculation incorporates a harmonic mean of precision and recall, weighted towards recall, and includes a penalty for fragmentation to better reflect translation quality.

METEOR was tested on the LDC TIDES 2003 Arabic-to-English and Chinese-to-English datasets. It showed improved correlation with human judgments compared to BLEU, particularly at the segment level, which is crucial for fine-grained translation quality assessment. The correlation values for METEOR were 0.347 for Arabic and 0.331 for Chinese, outperforming BLEU, which had correlations of 0.817 and 0.892 at the system level, respectively. Additionally, the study demonstrated the effectiveness of incorporating recall and penalizing fragmented matches in improving the metric's sensitivity to translation quality.

2.3 Embedding-Based Metrics and Neural Evaluation Approaches

Traditional evaluation of machine-generated summaries has based on lexical based metrics such as ROUGE (Lin, 2004), BLEU (Papineni et al., 2001) and METEOR (Banerjee and Lavie, 2005) Nevertheless, these methods have significant drawbacks most importantly, they fail to account for semantic similarity and paraphrased content. This is why more recent embedding-based metrics and neural evaluation approaches have been proposed to measure word sequence similarities that focus on the semantics of the generated summaries rather than surface overlap.

2.3.1 Embedding-Based Metrics

Embedding-based evaluation approaches are based on the idea that testing semantic similarity instead of word-for-word matches. Instead of counting overlapping words or phrases, these methods represent words and sentences in a multi-dimensional space and then measure how close the two texts lie in that space. This means that they are way better at spotting paraphrased content while the same time preserving the general idea.

Word Mover's Distance (WMD) (Kusner et al., 2015) is one of the early approaches of this kind It works by calculating the minimum effort required to transform one text into another using word embeddings. Building on this concept, MoverScore improves upon this by using pre-trained language models to derive word representations and discovers that its similarity measure is more fine-grained. In fact, research has shown that MoverScore is more in line with human judgments than ROUGE or BLEU metrics commonly used.

BERTScore, another popular metric, uses embeddings from the BERT model to determine token-by-token similarity. BERTScore defines words into a shared vector space, so it can pick up on subtle semantic relationships, making it very good for judging abstractive summaries. Conventional keyword-based techniques which are dependent on word similarity mainly focus on strict string matching while BERTScore employs the cosine similarity between the embeddings making it a more relaxed method to check for any changes in the text or to other attributes such as paraphrasing.

Embedding-based metrics, notwithstanding their benefits, do have a few drawbacks. One of the main difficulties is that they are not capable of ascertaining the truthfulness of statements. These methods give high scores to semantic similarity, so a well-written summary with misleading or inaccurate information may be a highly ranked summary. Due to this problem, researchers have investigated more sophisticated, neural-based approaches for evaluation that require reasoning and consistency to facts.(Zhang et al., 2020b)

2.3.2 Neural Evaluation Approaches

Neural evaluation methods are built on this foundation by the application of deep neural networks for the prediction of how a well a summary is matched to human judgment. Unlike these methods do not perform string matching or semantic similarity checking instead they are making use of pre-trained language models fine-tuned on the actual evaluation data. In this way, they are able to assess not only the degree to which a summary captures the meaning of the original text but also whether it stays true to the information in the original.

One representative instance is SummaC, which operates on Natural Language Inference (NLI) methods to detect if the summary drawn by a model pictures the same content and claims established in the text. Notably, this technique is highly efficient in exposing factual discrepancies which are frequently disregarded by traditional evaluation metrics, including embedding-based approaches. In the same vain, QAEval (Eyal et al., 2019) proceeds this sequence of evaluation by employing question-answering models that automatically confirm that the key facts from the source text are represented in the summary. This strategy is structured in a way as to ensure that the essential elements and the informational content are preserved.

Recent progress in evaluation metrics exploits deep learning techniques to apply modeling of human preferences with superior quality. Sellam et al. (2020) proposed BLEURT, a pre-trained BERT-based evaluation metric that is fine-tuned on human-rated text pairs for more

robustness. BLEURT generalizes across domains and tasks by using synthetic perturbations of the reference sentences. BLEURT beats all prior evaluation metrics in the WMT Metrics Shared Task and the WebNLG dataset, revealing a stronger correlation with human evaluations. In contrast to BERTScore, BLEURT is explicitly trained to predict human ratings and thus more accurately quantifies fluency, coherence, and factual consistency in LM outputs. BLEURT, on the other hand, relies on high-quality human ratings in tuning its model, making it less feasible in low-resource languages and specialized domains.

These neural-based methods have made huge strides in the ability to evaluate a summary compared to many previous metrics which addressing many of the shortcomings. Using "reasoning" and "factual validation", they provide a more robust measure of summary quality over emissive word-level matching methods.

The transition from using lexical-based metrics to embedding-based and learned metrics has led to substantial quality improvements across a range of studies. Although simple and easy-to-use, BLEU and ROUGE are not capable of measuring the profound semantic relationships. Unlike BERTScore, which is potent for this reason, BLEURT does this well by not only using contextual embeddings, but also, integrating human-rated data to provide a more robust evaluation. The findings imply that hybrid methods, which incorporate lexical, embedding, and learned evaluation metric approaches, are more accurate and scalable indicators of text generation performance. (Sellam et al., 2020)

Though embedding-based and neural evaluation metrics have their advantages, they also suffer from a number of challenges. Firstly, computational cost is still a substantial limiting factor, as deep learning models need much more computational power than traditional lexical measures. Second, generalizability can be an issue; many neural models will achieve good performance in certain domains but not transfer well to others. Another concern is interpretability, as neural networks typically function as black boxes, preventing intuitive understanding of the reasoning behind a specific summary receiving a certain score.(Sellam et al., 2020).To overcome these challenges, novel hybrid methods have been proposed that employ overall syntactic, semantic and factual consistency evaluations.

2.4 Redundancy and Coherence in Summarization Evaluation

Even though the summarization systems are the ones that bring it to the summaries of the texts, the systems often face two major problems: redundancy and coherence. Redundancy starts as the case when the summaries paraphrase or reiterate the same phrases again and

again, thus, quality of the summaries is reduced, and the absurdly added repetition is made. Absence of coherence makes it impossible for the reader to follow the logical chain as a result, the summaries become difficult for them to grasp. These factors negatively impact the quality of the produced summaries, restricting their availability.(Cardenas et al., 2024)

The recent works were mainly focused on dealing with these adverse effects through the development of novel evaluation techniques addressing redundancy and coherence issues. The detection of redundancy is commonly executed by lexical overlap examination to know the identical phrases or sentences. A more intelligent approach makes use of embedding-based analyses to detect paraphrased content and reused ideas(Rahman and Borah, 2021). The coherence assessment measures the logical relationship of the summary items. The use of entity-based coherence models and neural coherence metrics in gauging the logical connection between sentences are pivotal for assessing machine-generated summaries.(Goyal et al., 2022)

By effectively tackling of redundancy and coherence, researchers wish to advance summarization models that could create clear, coherent, logically arranged, and informative summaries that are easy to understand (Li et al., 2021).

2.4.1 Challenges of Redundancy

Redundancy remains one of the major issues that people face in text summarization, and is more so with extractive strategies, where sentences are opted directly from the original text without alteration. This often causes duplication, which in turn results in poor efficiency and overall declines in the quality of summaries (Cardenas et al., 2024). Still, when it comes to summarization, the presence of redundancy is extremely important for content that is concise although, on the other hand, the summarization researchers, are of the view that, aggressive redundancy elimination might lower the coherence and thus lead to the occurrence of disjointed or incomplete summaries.

Indeed, Cardenas, Gallé, and Cohen (2024) are who directly express the need for redundancies to be weighed against comprehension. They assert that the repetition of lucrative thoughts which, in effect, reduces redundancy, is essential but however, under no circumstances should this be achieved at the cost of logical reasoning in the summary. Their work shows two independent strategies to deal with redundancy: reward-oriented optimization and unsupervised schemes.

The reward guided approach uses RL to train redundancy reduction and sentence cohesion. This guarantees that the chosen sentences are preserving important information without unnecessary redundancy. On the other hand, the unsupervised strategy is based on some psycholinguistic theories like Micro-Macro Structure Theory of Text Comprehension propounded by Kintsch and Van Dijk in 1978. By minimizing excessive verbal redundancy through human cognitive processes, this method provides natural and efficient text coherence by avoiding any discrepancies between the parts and as such retaining full text comprehension.

The central finding of their investigation is that the summaries, of which the main focus is emphasizing cohesion, are largely more organized and easier to read. However, on the contrary, the rigorous redundancy decreases the informational content. Consequently, redundancy is also addressed as context-dependent; for instance, the level of redundancy appropriate for news articles differs markedly from that required in scientific papers, where strategic redundancy is deliberately employed to enhance readers' knowledge acquisition and comprehension.

The paper proposes hybrid methods that correlate redundancy control with coherence improvement. By these means the informativeness, readability, and logical consistency of the summaries, which without such means might have become fragmented, are successfully preserved. So, authors emphasize that as is the case with much of the knowledge we acquire managing redundancy cannot be entirely eliminated, and excessively reducing of it may disrupting the natural flow of information. Context-sensitive adjustments in redundancy, therefore, play a pivotal role in enhancing and dynamically advancing summarization evaluation processes.(Cardenas et al., 2024)

2.4.2 Coherence and Discourse-Based Evaluation

In summarization, coherence indeed has a key role as it creates the summary that flows logically and is structured, readable, and contextually connected. Adding sentences from the source is the way extractive summarization methods keep up a high level of sentence-level coherence as instead of describing the application of some new technology, they directly quote ideas from the source. As a consequence, such methods can occasionally overlook critical points or introduce abrupt transitions, leading to missing or inconsistent references, ultimately resulting in an extract that appears fragmented or disjointed.

To resolve such issues, Goyal, Li, and Durrett (2022) invented SNaC (Summary Narrative Coherence), a framework that is specially created to enable the study of coherence in long-form summaries with high precision. Unlike conventional coherence metrics, SNaC is more advanced. The traditional method gives just one overall score which could be based on similarity or linguistic features whereas SNaC goes further. It notices concrete coherence errors like abrupt transitions that shouldn't happen, missing references of important events, and unclear introductions of characters or terms.

The notable aspect of SNaC is its span-level annotations which highlight the spots where coherence deteriorates. This quality is exceptionally valuable in the automatic training of machine learning models which can be programmed to automatically detect and correct coherence problems. The research highlighted that even the most cutting-edge summarization models generate summaries that is more resemble disconnected listings of events than a cohesive narrative.

Through the addition of human-labeled coherence errors, the researchers managed to train an automatic classifier which recognizes narrative inconsistencies far better than previously available techniques. They found that other widely accepted evaluation methods, like ROUGE and BERTScore, are inadequate to address coherence problems because they do not penalize them thus not reflecting the actual reason why it was written the way it was.

Overall, the study argues that discourse-aware evaluation methods are very crucial. Instead of only being concerned about whether all the necessary information has been given in the summary coherence-focused tests would make the models feel obligated to read clearly, to have a good structure, and to develop the logical sequence first (Goyal et al., 2022). Implementing this is a must for the improvement of both the summarization process and the subsequent evaluation, providing that the produced text is not only informative but also coherent enough to understand easily.

2.5 Hybrid Approaches for Summarization Evaluation

The hybrid approach, which is a combination of lexical and semantic similarity, has become a popular and effective tool for measuring the quality of summaries. This part of the paper discusses two aspects of hybrid summarization evaluation: the fusion of lexical and semantic metrics and the relative performance of hybrid models.

2.5.1 Combining Lexical and Semantic Metrics

Although these classical measurement tools as ROUGE and BLEU have a good performance like a measurement of word overlap, they lack deeper meaning. As these techniques depend on n-gram matching, they do not detect paraphrasing, synonym use, and context changes. Hence, rewriting the same idea in different words can result in a lower score for the text, which creates a gap between evaluation by machines and humans.

To solve these problems, Sarwar et al. in 2024 proposed HybridEval, a novel evaluative metric that combines ROUGE scores with the cosine similarity scores produced from InferSent embeddings. This hybrid method enables the alignment of two measures depending on lexical similarity (word overlap) and, in this case, also on semantic similarity (conceptual alignment). Opposed to just counting the number of similar words, HybridEval measures the extent to which the meaning of the machine-generated summary is preserved according to the reference text.

A significant advantage of HybridEval is that it assigns sentences to one of six categories Extreme, Critical, Tricky, Negative, Average, and Perfect, which leads to a more comprehensive analysis of the degree of different summaries corresponding to human-written references. This detailed categorization not only makes the evaluation more understandable, but also helps researchers to single out the particular weaknesses in summarization models.

Empirical findings indicate that HybridEval outperforms traditional evaluation methods, improving accuracy by 10-15% compared to standalone lexical-based metrics. It also shows a stronger correlation with human judgment, reinforcing the idea that integrating semantic analysis into evaluation frameworks leads to more reliable assessments.

The study ultimately argues that hybrid evaluation approaches should become the new standard for summarization assessment. As modern summarization models move beyond simple extractive methods, relying solely on word overlap is no longer sufficient. Future research should continue to refine methods that balance syntactic and semantic evaluation, ensuring that summaries are judged on meaning, coherence, and readability, rather than just word-for-word similarity (Sarwar et al., 2024).

2.5.2 Comparative Performance of Hybrid Models

Hybrid evaluation metrics, for example, HybridEval, hold significant value with respect to determining the quality of summarization by being both lexical and semantic similarity based but their real effectiveness must be evaluated on different summarization models. The application of hybrid summarization systems, which integrate extractive and abstractive methods with the context-aware selection algorithm to improve summary coherence and informativeness, recently received attention from the researchers.

Kherwa et al. (2024) presented a hybrid model of text summarization that semantic embeddings and contextual modeling gives to the possibility of creating better summary structure. The findings of the study indicate that hybrid models are a way for abstracting texts that would be more coherent than current methods, which is a major issue in the extractive summarization process. The merging of semantic embeddings is a case in point that hybrid techniques provide the opportunity to the content that will be extracted to have the discourse of the connection making the summaries more readable and in line with the original context.

Best known for their flexibility, hybrid models can integrate both more relevant pieces of information and less redundancy. Though extractive summarization is an effective tool in preserving factual content, it often includes excessive detail or repeated information that can be overwhelming for the reader. At the same time, abstractive models can rephrase the content but sometimes they lose some crucial details, which make the summary incomplete. The hybrid blueprint offers a fair solution that saves the essential content by eliminating redundancy and unnecessary information.

Furthermore, studies that utilized HybridEval and supplementary semantic-based evaluation metrics showcase the hybrid models score higher than the models evaluated merely with ROUGE or BLEU. These findings demonstrate the inadequacy of lexical-based evaluation metrics in terms of quality assessment of summaries, thus justifying the need for the involvement of semantic analysis in evaluation frameworks.

This research has reached the conclusion that hybrid summarization models evaluated by means of hybrid evaluation techniques are the most valid reflectors of the quality of the summaries. As a result, the evaluation frameworks would synchronize both human judgment and the progress in machine learning. With the continuous evolution of summarization models, the traditional evaluation methods alone would not be sufficient in measuring the complexity of modern and contextually aware summarization systems (Kherwa et al., 2024).

The realization of these points highlights the necessity for the ongoing improvement of evaluation methodologies; thus, the development of future frameworks that would properly measure coherence, informativeness, and redundancy control in the high-quality summarization models will be ensured.

2.6 Advancements in Summarization Evaluation

The field of abstractive text summarization (ATS) has garnered significant interest due to its capability to generate concise, coherent summaries that are better than the traditional extractive summaries. ATS interprets and rephrases content, generating summaries that closely resemble human-authored texts. This capability has made ATS a critical area of research, particularly as demand for automated content generation increases across various domains.

Sunusi, Omar, and Zakaria in 2024 provides an extensive analysis of recent developments in ATS, covering model architectures, datasets, evaluation methods, and emerging challenges. Their study traces the evolution of ATS models, from early sequence-to-sequence (Seq2Seq) architectures, which relied on Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRUs), to more advanced Transformer-based models such as BERT, GPT, and BART.

The introduction of the attention mechanism not only improved the ATS models but also made them easier to interpret. This made it possible for the models to concentrate on the relevant parts of the input text while generating the summary. Furthermore, the introduction of pointer-generator networks shifted the balance to some extent, allowing the models to both copy critical information from the source and create new sentences, thus mitigating some of the earlier fluency and coherence issues. However, despite these advancements, training ATS models remains challenging, particularly for low-resource languages, where the scarcity of high-quality and diverse datasets hinders model performance.

One of the main challenges that ATS faces is evaluating generated summaries. The research paper notes that ROUGE is still the mainly used metric even though it has its limitations in evaluating semantic similarity, coherence, and factual correctness. This issue is especially concerning in domains like scientific, medical, and legal texts, where factual precision is important. The writers support the merging of semantic-based evaluation methods which test the preservation of meaning, not just the lexical similarity.

Apart from the evaluation problems, ATS is also confronted by some critical challenges. One of the common problems is hallucination where the models produce unreliable or misleading information not supported by the source text. This is of particular concern in the scientific, medical, and legal fields where the factual consistency is of paramount importance. The study also highlights the challenge of summary length control because the majority of the models often fail to find a balance between being concise and informative enough.

Another major issue is dataset bias. Most of the current summarization datasets are concentrated on newswire articles, which brings about the problem of the generalization of the models to other areas of interest. The writers mentioned the necessity of collecting more diverse and field-based datasets mainly for low-resource languages to extend the adaptability of the ATS models to datasets other than the ones extensively researched.

Through a roadmap provided by Sunusi and his colleagues (2024), the path to progress the abstractive text summarization and to continuous innovation in model architectures, evaluation frameworks, and rigorous dataset curation. Even though Transformer-based ATS models have the potential to increase the quality of summaries, the concerns that involve evaluation, the reliability of facts, and dataset bias are still there. Withal, in the short run, a comprehensive method that includes better model architectures, evaluation techniques tailored to the subject area, and proper dataset curation will be necessary to improve organizations' applicability of the abstractive summarization systems.

2.7 Summary of the literature review

Title	Reference	Research Gap / Area of Improvement	Findings & Results
BLEU: A Method for Automatic Evaluation of Machine Translation	Papineni et al. (2002)	The paraphrase and synonyms are penalized because BLEU does not have recall-based evaluation. In other words, BLEU depends on n-gram precision only. It also fails to evaluate the semantic similarity or grammatical coherence.	BLEU is a good system comparison tool because of its high correlation with human judgment on the large corpus. However, it does not correctly evaluate the quality of individual sentences. BLEU can be used as a general metric for the quality of translations but, on its own, it does not tell anything about the meaning and the structure of the sentences used in the original text.
ROUGE: A Package for Automatic Evaluation of Summaries	Lin (2004)	ROUGE is mainly based on the intersection of words while it fails to consider the semantic relationship between the words. It does not take into consideration the use of synonyms, paraphrases, or the organization of discourse.	In extractive summarization, ROUGE shows a strong association with human judgments. Some variants like ROUGE-L and ROUGE-SU can overcome basic problems of n-gram co-occurrence. ROUGE is widely used but should be combined with semantic-based metrics to enhance evaluation accuracy.

METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments	Banerjee & Lavie (2005)	By integrating recall, stemming, usages of synonyms, and penalties for word order METEOR attempts to surpass BLEU; however, the system is still devoid of any understanding that is quite deep.	METEOR surpasses BLEU in correlation with human judgment, especially at the sentence level, due to it taking into account both the recall and the paraphrasing. METEOR, with the help of more good features that fix the remaining issues such as factual consistency and deep semantics, provides a comprehensive assessment.
BERTSCORE: Evaluating Text Generation with BERT	Zhang et al. (2020)	Semantic meaning and paraphrasing are the shortcomings of the existing n-gram-based metrics.	BERTSCORE is more in accordance with the evaluations made by humans than the conventional metrics like BLEU and METEOR. Strengthening its position as a trustworthy and resilient assessment metric, it has been successful in both machine translation and image captioning tasks.
SUMMAC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization	Laban et al. (2022)	The previous trials with NLI models to detect inconsistencies were not successful as they suffered from the differences in input granularity.	The models, SUMMAC, especially SUMMACCONV, have exhibited better performance than the previous inconsistency detection strategies with a balanced accuracy of 74.4%. The research demonstrates that operating with sentence-level entailment in NLI models remarkably enhances their practical efficiency in summary inconsistency detection.

Accurate and Nuanced Open-QA Evaluation Through Textual Entailment	Yao and Barbosa (2024)	Present Open-QA evaluations are prone to vagueness and lack the ability to reflect semantic comprehension. The conventional metrics do not correlate well with human assessment.	The proposed entailment-based evaluation metric is more in line with human judgment, it provides an answer ranking that is more detailed and allows partial credit assignment. Alongside that, its accuracy is greater than that of the currently available Open-QA evaluation metrics and it demonstrates that it is the better and more reliable technique for assessing LLM-generated responses.
BLEURT: Learning Robust Metrics for Text Generation	Sellam et al. (2020)	Traditional text evaluation metrics like BLEU and ROUGE correlate poorly with human judgment. Existing methods struggle with semantic understanding.	With the help of BERT, learns to distinguish the human and machine-generated texts, which adds to its ability to improve the correlation with human assessments using synthetic pre-training data. Dominating the previous metrics, it yields superior results in the WMT Metrics Shared Task and WebNLG datasets, proving that it possesses robustness even when the data is scarce.

On the Trade-off between Redundancy and Local Coherence in Summarization	Cardenas et al. (2024)	Existing summarization models find it hard to obtain a triumvirate of redundancy, cohesion and informativeness. Especially, it is a big challenge in the case of highly redundant documents.	The study finds that, by managing cohesion, the authors are able to have better content organization in summaries without decreasing informativeness in a significant way. Although the cohesion of the unsupervised models that is high, they miss out on informativeness. The evaluations conducted by people are positive to the function of cohesion optimization in the quality of the summaries.
SNaC: Coherence Error Detection for Narrative Summarization	Goyal et al. (2022)	The current evaluation systems are not capable of identifying coherency faults in the long-text summarization. The availability of fine-grained coherence evaluation is insufficient.	The framework proposed by SNaC is the first to make coherence assessment possible, helping to spot the coherence mistakes in lengthy summaries. The study put forward the error taxonomy and annotation protocol that would be helpful for the automatic coherence modeling and evaluation.

HybridEval: An Improved Novel Hybrid Metric for Evaluation of Text Summarization	Sarwar et al. (2024)	The inadequacy of the semantic meaning and structure measurement in existing metrics like BLEU and ROUGE is the cause of the text summarization evaluation that is not accurately observed.	The HybridEval metric that combines cosine similarity with the BLEU and ROUGE scores, additional evaluation accuracy of 10-15% is achievable. It is a good way to address the issue of traditional n-gram-based evaluation methods as it correlates better with human judgment and more accurately reflects semantic similarity.
Exploring Abstractive Text Summarization: Methods, Dataset, Evaluation, and Emerging Challenges	Sunusi et al. (2024)	Currently the main mechanism of abstractive summarization is the encoder-decoder framework, however, due to factors such as lack of sufficient data, limited evaluation capacity and constraints of factuality, these models have difficulties.	The study emphasizes the shift to Transformer-centered models, the implementation of attention and pointer-generator mechanisms, and highlights the requirement for more reliable evaluation metrics rather than ROUGE. The main obstacles encountered in the process are problems with the quality of the dataset, dealing with model hallucination, and appropriate control of summary length.

Contextual Embedded Text Summarizer System: A Hybrid Approach	Kherwa et al., (2024)	The majority of summarization models face difficulties tackling both extractive and abstractive techniques efficiently. Several models cannot keep contextual connectedness and coherence in the summaries they generate. There arises a need for the hybrid models that would integrate deep learning to accomplish the improvement in accuracy and informativeness.	The proposed model is the hybrid contextual embedding with both extractive and abstractive summarization techniques. It is more coherent, fluent, and informative than the methods used to study. The model's correlation with human evaluations is stronger, and it surpasses such traditional metrics as ROUGE and BERTScore. The study gains a conclusion that hybrid summarization with contextual embeddings produces a significant betterment in the quality and coherence of the summary. The method ensures factual consistency and contextual relevance.

Table 2.1 - Summery

Table 2.1 show the summery of the literature review.

2.8 Chapter Summery

This chapter was summarized the existing research approaches used for the LLM evaluation. Gaps and the limitations of the discussed approaches were identified and discussed. Several of these issues will be further examined during this research in the upcoming chapters.

CHAPTER 3: METHODOLOGY

This chapter is an important chapter in a research type dissertation and the methodology shall be detailed well. Aspects relating to the proof-of-concept specification which includes process flow diagrams, design assumptions relating to the scope of the proof of concept, prototype architecture, algorithmic design details etc. shall be included. A key aspect to be detailed would be the algorithmic contribution made in a research type project.

3.1 Systematic Approach

The systematic approach of this research is described as shown in Figure 3.1. Each phase of this approach is discussed in more detail level in the following sections.

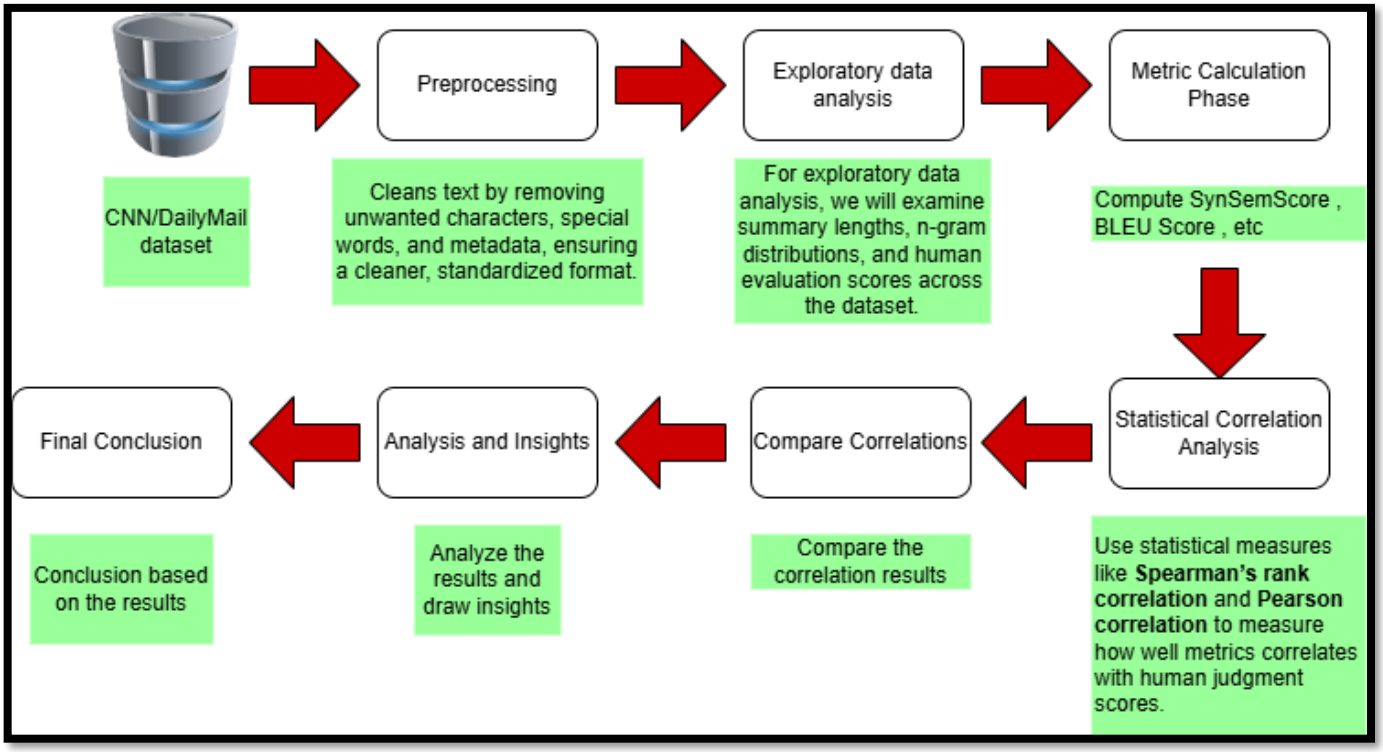


Figure 3.1 - Systematic Approach

3.2 Data Collection

For my project, I have utilized the SummEval dataset, which has been specifically developed to evaluating the quality of summarization and also has been used as a benchmark for automatic summarization systems. The dataset has been accessible via Hugging Face and has had a set of human-generated ratings about various attributes. This has been a thorough breakdown.

3.2.1 SummEval Dataset Overview

The CNN/Daily Mail dataset has been utilized in the SummEval project and has consisted of machine-generated summaries from different systems that have been rated by human annotators. The dataset has been structured to make it easy to compare different summarization evaluation metrics, like ROUGE and BLEU, assessing how consist of these metrics with human scores. Through the provision of a benchmark for the performance evaluation of the different summarization models, the dataset has not just been insightful for automatic summarization researchers, but it has also encouraged the development of improved evaluation techniques for the field. Additionally, it has been accessed at the SummEval repository on GitHub and Hugging Face.(https://github.com/Yale-LILY/SummEval)

3.2.2 Dataset Contents

The dataset provides valuable insights for extracting the human evaluations that have been carried out for a number of summarization models on the CNN/Daily Mail news articles dataset. The data is concentrated on four evaluation aspects that each have been rated by human evaluators:

1. Coherence: How The summary is logical and easy to follow.
2. Consistency: The summary is an accurate rather than misleading reflection of what is in the source article.
3. Fluency: The summary has proper grammar and is easy to read.
4. Relevance: How well the summary captures the most important points from the source text.

The four metrics are measured based on a Likert scale ranging from 1 to 5 for summaries by various models.

3.2.3 Structure of the Dataset

The dataset comprises of original articles obtained from CNN and Daily Mail, as well as automatically created summaries by machine models. Additionally, it comes with human evaluations that grade the summaries on four aspects: coherence, consistency, fluency, and relevance. Each summary is also annotated with the specific system that was used to generate it. The dataset is publicly available and can be utilized through Hugging Face's datasets library. Figure 3.2 show the structure of the dataset that we are using.

[17]:

	machine_summaries	human_summaries	relevance	coherence	fluency	consistency	text
0	[donald sterling , nba team last year . sterli...	[V. Stiviano must pay back $2.6 million in gif...	[1.6666666667, 1.6666666667, 2.3333333333, 3.3...	[1.3333333333000001, 3.0, 1.0, 2.6666666667000...	[1.0, 4.6666666667, 4.3333333333, 4.3333333333...	[1.0, 2.3333333333, 4.6666666667, 5.0, 5.0, 5....	(CNN)Donald Sterling's racist remarks cost him...
1	[north pacific gray whale has earned a spot in...	[The whale, Varvara, swam a round trip from Ru...	[2.3333333333, 4.6666666667, 3.666666666700000...	[1.3333333333000001, 4.6666666667, 3.666666666...	[1.0, 5.0, 4.6666666667, 3.6666666667000003, 5...	[1.3333333333000001, 5.0, 5.0, 4.3333333333, 5...	(CNN)A North Pacific gray whale has earned a s...
2	[russian fighter jet intercepted a u.s. reconn...	[The incident occurred on April 7 north of Pol...	[4.0, 4.0, 4.0, 3.3333333333, 3.3333333333, 4....	[3.3333333333, 4.3333333333, 1.6666666667, 2.6...	[3.6666666667000003, 4.3333333333, 5.0, 4.3333...	[5.0, 5.0, 4.6666666667, 5.0, 5.0, 5.0, 5.0, 5...	(CNN)After a Russian fighter jet intercepted a...
3	[michael barnett captured the fire on intersta...	[Country band Lady Antebellum's bus caught fir...	[2.0, 3.0, 2.6666666667000003, 3.3333333333, 3...	[2.0, 3.0, 2.6666666667000003, 3.3333333333, 3...	[2.6666666667000003, 5.0, 5.0, 5.0, 5.0, 5.0, ...	[2.3333333333, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5...	(CNN)Lady Antebellum singer Hillary Scott's to...
4	[deep reddish color caught seattle native tim ...	[Smoke from massive fires in Siberia created f...	[1.6666666667, 3.6666666667000003, 3.333333333...	[1.6666666667, 3.6666666667000003, 1.666666666...	[5.0, 5.0, 5.0, 5.0, 4.6666666667, 5.0, 5.0, 5...	[2.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, ...	(CNN)A fiery sunset greeted people in Washingt...
...	...	...	...	...	...	...	...

Figure 3.2 - Structure of the dataset

3.3 Data Pre-processing

In order to do the necessary preparations for the SummEval dataset so as to become viable for the analysis, certain preprocessing steps were taken that would ensure consistency, clean formatting, and a proper handling of both textual summaries and human evaluation scores

3.3.1 Loading and Parsing the Dataset

The dataset has been accessed using the Hugging Face library, and the summaries, source documents, and human evaluation scores have been retrieved. All components have been transformed into a structured format, with each record composed of the source text, machine-generated summary, and corresponding human ratings for coherence, consistency, fluency,

Figure 3.3 - Loading and parsing the dataset

and relevance. Figure 3.3 has shown how to load and parse the dataset into the Jupiter environment.

3.3.2 Missing Data Handling

Handling missing values primarily involves addressing incomplete human evaluation scores. Records with missing human ratings are either deleted in order not to pose biases during evaluation.

3.3.3 Normalization of Human Ratings

Human evaluation scores (ranging from 1 to 5) are normalized to make the scaling consistent for different annotators. This normalization is the reason of the reduction in the variability between the annotators, and also, it makes the scores much more comparable between the different systems. Figure 3.4 depicts how to normalize the data before preprocessing.

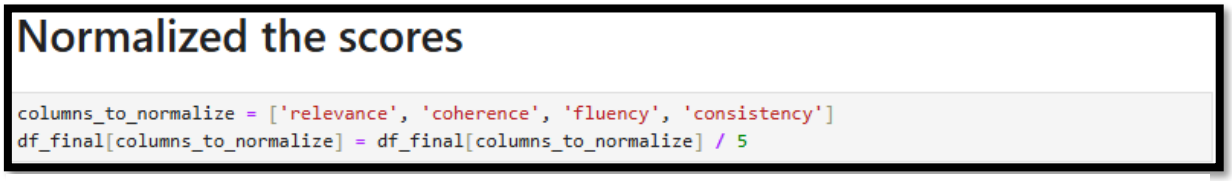


Figure 3.4 - Normalization

3.3.4 Text Preprocessing

The text data, including both source documents and generated summaries, is pre-processed by:

- Lowercasing: All text is converted into lowercase, in order to prevent any issues related to case-sensitivity.
- Removing Special Characters: Non-alphanumeric characters (except for essential punctuation) are deleted to ensure the uniformity of the text.
- Tokenization: Text is tokenized into words to prepare for further text-based evaluation or model input.
- Stopword Removal: The common stopwords that should not be included in the quality evaluations are eliminated to focus on the main content.

Data preprocessing

```
try:
    stop_words = set(stopwords.words('english'))
except LookupError:
    nltk.download('stopwords')
    stop_words = set(stopwords.words('english'))

try:
    nltk.data.find('tokenizers/punkt')
except LookupError:
    nltk.download('punkt')

def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stop_words]
    cleaned_text = ' '.join(tokens)

    return cleaned_text
```

Figure 3.5 - Preprocessing

All the preprocessing steps can be shown in the Figure 3.5.

3.4 Exploratory Data Analysis (EDA)

Our analysis focuses on comparing human-written and machine-generated summaries by examining their structure, linguistic patterns, and quality scores.

1) Summary Length Analysis

The contrast of the structural characteristics of machine-written, and human-written summaries is covered through the word count of each type. This gives a clear picture of their respective lengths and shows if the machine-generated ones are typically longer or shorter than the ones made by humans. Similarly, the data summarization of the word count such as the mean, median, standard deviation, minimum, and maximum help to check the differences in summary lengths. The use of such statistics provides the background to evaluate the trustworthiness and conciseness of AI-generated abstracts in relation to those written by humans.

2) Text Visualization (Word Clouds)

Word clouds are generated for summaries to see the topmost used words both in human and machine-generated texts. A word cloud is a visual display of words representing the frequency of the words, that is, more frequently used words are larger than the others. This analysis gives a clear picture that AI generated models have a different or additional vocabulary to an extent than human authors. Looking at the word clouds makes it evident how the words have been chosen, used more than once, and whether in some cases machine-generated summaries are biased.

3) Bigram Analysis (Common Two-Word Phrases)

To further analyse the linguistic patterns in the summaries, bigrams (two-word phrases) are extracted from both human and machine-generated texts. The most frequently occurring bigrams are counted, ranked, and plotted, highlighting the top 20 for each type of summary. Comparing the most common word pairs reveals patterns in phrase structure, repetition, and cohesion between human and machine-written summaries. This analysis helps in understanding how AI constructs sentences and whether it follows natural language patterns similar to human writers.

4) Quality Score Distributions

To evaluate the overall quality of machine-generated summaries, four key quality metrics are examined: relevance, coherence, fluency, and consistency. These metrics

are visualized using histograms to analyze their distributions. Relevance measures how well the summary captures the key content of the original text, coherence evaluates logical flow and readability, fluency assesses grammatical correctness and natural language usage, and consistency examines factual accuracy relative to the original content. These histograms allow for the detection of patterns or biases in machine-generated text quality and provide insights into areas where AI-generated summaries may need improvement.

5) Correlation Analysis Between Human Ratings

The correlation analysis between human ratings provides insights into how different evaluation metrics relevance, coherence, fluency, and consistency are interrelated in the assessment of human and machine-generated summaries. By computing Pearson correlation coefficients, the strength and direction of these relationships are identified. The heatmap visualization illustrates these correlations, with values ranging from -1 (strong negative correlation) to +1 (strong positive correlation). A high positive correlation between metrics (e.g., fluency and coherence) suggests that well-structured summaries tend to be both fluent and coherent. Conversely, weaker correlations indicate that some evaluation criteria may be independently assessed. This analysis helps in understanding the dependencies among different rating dimensions and informs future improvements in summarization models by focusing on key areas that impact overall summary quality.

```
# Compute correlation matrix
correlation_matrix = Preprocessed_dataset[['relevance', 'coherence', 'fluency', 'consistency']].corr()

# Plot heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title("Correlation Matrix of Human Ratings")
plt.show()
```

Figure 3.6 - Correlation between dimensions

Figure 3.6 depicts how to visualize the correlations between dimensions using heat maps.

3.5 Metric Calculation phrase

When it comes to CNN/Daily Mail dataset usage in measuring the metrics, several assessment techniques have frequently been applied to evaluate the overall quality of the autogenerated summaries concerning reference texts available in the specific dataset. The metrics

themselves have been used for the purpose of exploring how closely the summary contents relate to the original reference summaries in terms of language fluency, coherence, and fidelity. Below, the three most commonly used metrics BLEU, METEOR, and ROUGE, their nature, pros and cons of applying them on the CNN/Daily Mail dataset have been discussed. The reason I have compared just these three statistical metrics is that these have been the three most widely utilized evaluation metrics that are for traditional LLM summarization, and they also don't use any transformer-based architecture. Subsequently, we have mentioned a new evaluation metric SynSemScore, and have explained how it has proposed to increase the efficiency of the traditional methods.

3.5.1 BLEU (Bilingual Evaluation Understudy).

BLEU is among the very first and is the most popular metric up-to-date for generating texts that are judged against similar texts. For the CNN/Daily Mail dataset, this metric determines how the n-grams (i.e., sequences of contiguous words) in the generated summary match those in the reference summary.

3.5.1.1 Advantages:

- BLEU is a computationally cost-effective method because of enabling quick scoring of summaries throughout the whole dataset which is a great deal for CNN/Daily Mail.
- It is a basic precision-based metric, sticking to n-gram overlap gives a direct and easy understanded precision measure.
- BLEU is frequently utilized; thus, it presents a point of comparison that is beneficial for analyses of different summarization models that were trained with the CNN/Daily Mail dataset.

3.5.1.2 Disadvantages:

- BLEU gives attention to precision alone and does not take recall into account, which can lead to a situation where models generating summaries containing some specific reference content have extra penalties.
- Apart from that, focusing exclusively on exact n-gram matches makes BLEU missing word meaning and fluency, which in turn are problems for summarization tasks, such as CNN/Daily Mail dataset, which have frequent paraphrasing.

- BLEU can thus find a longer summary compared to the reference by over-penalizing, which is not a valid direction in the news summarization task when concise expression is essential.

3.5.1.3 Code Example for BLEU Calculation:

```python
import nltk
from nltk.translate.bleu_score import sentence_bleu

# Example
reference = "The government implemented new policies to curb inflation."
candidate = "The new policies were introduced by the government to control inflation."

# Tokenize sentences
reference_tokens = [nltk.word_tokenize(reference)]
candidate_tokens = nltk.word_tokenize(candidate)

# Calculate BLEU score
bleu_score = sentence_bleu(reference_tokens, candidate_tokens)
print(f"BLEU Score: {bleu_score:.4f}")
```

Figure 3.7 - BLEU calculation

Figure 3.7 show the python code example for the BLEU algorithm Calculation

3.5.1.4 Steps for BLEU Calculation:

1. N-gram Precision

    - The modified n-gram precision used by BLEU is a method whereby the candidate translation system checks from the reference translations as per the number of n-grams (uninterrupted word sequences) that occur in the candidate translation. The algorithm performs the counting of each n-gram as follows:

    a) The frequency of this n-gram in the candidate translation.

    b) The maximum times it can appear in any reference translation.

    If a word or n-gram occurs more times in the candidate translation than in the reference, the extra occurrences are clipped to the reference count.

2. Calculate Modified n-gram Precision

    - For each n-gram size (unigram, bigram, etc.), precision is calculated as:

$$P_n = \left(\frac{Clipped\ count\ of\ matching\ n-grams}{Total\ number\ of\ n-grams\ in\ the\ candidate\ translation}\right)$$ *Equation 3.1*

BLEU usually considers n-grams up to 4-grams (unigrams, bigrams, trigrams, and 4-grams).

3. Geometric Mean of Precisions

    - It is by using the geometric mean that BLEU integrates the modified precision scores of various n-grams sizes measured (which reflect fluency). The formula is:

$$P_n = exp\left(\frac{1}{N}\prod_{n=1}^{N} log\ p_n\right)$$ *Equation 3.2*

where P_n is the precision for each n-gram size, and N is the highest order of n-grams (usually 4).

4. Brevity Penalty (BP)
    - Translations that are too short are penalized by BLEU. The brevity penalty is determined based on the length of the candidate translation c and the length of the reference translation r. The brevity penalty has the following definition:

$$BP = \begin{cases} 1, & if\ c > r \\ e^{\left(1-\frac{r}{c}\right)} & if\ c \leq r \end{cases}$$ *Equation 3.3*

This ensures that very short translations are penalized.

5. Final BLEU Score
    - The BLEU score is calculated as the product of the geometric mean of n-gram precisions and the brevity penalty:

$$BLEU = BP \cdot P_n$$ *Equation 3.4*

The final score ranges from 0 to 1, where 1 means the candidate translation is a perfect match with the reference.

3.5.2 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE, specifically ROUGE-N (for n-gram overlap) and ROUGE-L (for longest common subsequence), is a recall-oriented metric that fits exceptionally well as the CNN/Daily Mail dataset. ROUGE is the most-used method of generating summary because it prioritizes the preserving of the main message, which is critical for news articles.

3.5.2.1 Advantages:

- ROUGE is recall-oriented, its main feature is the preservation of key content from the reference summary, which is essential, especially for the summarizing of news articles.
- ROUGE-L, with the longest common subsequence (LCS) approach, is very practical in gauging longer summaries as it gets to the point on the whole content overlap more than just the exact n-gram matches.
- Besides its widespread acceptance, ROUGE is a standard metric, making it an important tool for benchmarking.

3.5.2.2 Disadvantages:

- Similar to BLEU, ROUGE doesn't take into account how well the summary flows or if it's grammatically correct.
- ROUGE might exaggerate the degree of content overlap thus could penalize summaries that certainly left out some details but still grasped the main idea, which is a frequent necessity in concise news summarization.

3.5.2.3 Code Example for ROUGE Calculation:

```python
from rouge import Rouge

# Example
reference = "The government implemented new policies to curb inflation."
candidate = "The new policies were introduced by the government to control inflation."

# Calculate ROUGE scores
rouge = Rouge()
scores = rouge.get_scores(candidate, reference)
print(f"ROUGE Score: {scores}")
```

Figure 3.8 - ROUGE calculation

Figure 3.8 show the python code example for the ROUGE algorithm Calculation.

3.5.2.4 Steps for Calculating ROUGE-N

1. Tokenize the Candidate and Reference Summaries
    - Break the candidate and reference summaries into n-grams (continuous sequences of words of length n).
    - For ROUGE-1, we use unigrams (single words).
2. Count Matching N-grams
    - Locate the n-grams in the candidate summary that also appear in the reference summary.
    - For ROUGE-1, we consider how many unigrams from the candidate appear in the reference.
3. Calculate ROUGE-N Recall
    - ROUGE-N is a metric that evaluates recall primarily which is the proportion of the reference that is captured by the candidate. The formula for ROUGE-N is:

$$ROUGE{-}N = \frac{Number\ of\ matching\ n{-}grams}{Total\ number\ of\ n{-}grams\ in\ reference} \quad \textit{Equation 3.5}$$

4. (Optional) Calculate ROUGE-N Precision
    - We can also compute precision which tells you the total number of right answers in candidate output as stipulated by the reference. The formula is:

$$\text{Precision} = \frac{\text{Number of matching n}-\text{grams}}{\text{Total number of n}-\text{grams in candidate}} \quad \textit{Equation 3.6}$$

5. (Optional) Calculate ROUGE-N F-Score
    - If both recall and precision are calculated, you can compute the F-Score to balance the two:

$$F_{\beta=1} = \frac{2.Precision.Recall}{Precision+Recall} \quad \textit{Equation 3.7}$$

3.5.2.5 Steps for ROUGE-L (Longest Common Subsequence)

1. Identify the Longest Common Subsequence (LCS) that is between the candidate and reference.
    - LCS: Longest subsequence that appears in both summaries in the same order (but not necessarily consecutively).
2. Calculate ROUGE-L Recall and Precision is:

$$R_{LCS} = \frac{Length\ of\ LCS}{Total\ words\ in\ reference} \quad \textit{Equation 3.8}$$

$$P_{LCS} = \frac{Length\ of\ LCS}{Total\ words\ in\ candidate}$$ *Equation 3.9*

3. Compute the ROUGE-L F-Score (optional) is:

$$F_{LCS} = \frac{2.p_{LCS}.R_{LCS}}{P_{LCS}+R_{LCS}}$$ *Equation 3.10*

3.5.3 METEOR (Metric for Evaluation of Translation with Explicit ORdering)

METEOR overcomes some of the limitations of BLEU by including recall in the equation and enabling synonym matching, stemming, and paraphrasing. METEOR proves to be more effective in terms of summary expressions that are more generic and allow the use of different words or sentence structures compared to the reference summaries in the CNN/Daily Mail dataset.

3.5.3.1 Advantages:

- METEOR takes into account synonyms and stemming, which are useful in advertising tasks, like different words to express the same idea.
- By considering at both precision and recall, METEOR is more balanced than BLEU.
- METEOR frequently coincides with human evaluations, especially in summarization tasks where fluency and meaning retention are the most important factors.

3.5.3.2 Disadvantages:

- METEOR computationally expensive than BLEU because of its advanced features like alignment, stemming, and synonym matching. The alignment requires more resources as the words and phrases are mapped between the generated and reference text, while stemming reduces the words to their root forms, and the synonym matching widens the evaluation to include the semantically similar words.
- Although it's rapidly gaining in popularity, it is not as widely seen and is not as standardized compared to BLEU; thus, model comparisons are more challenging between different models. The non-popularity of the product means fewer benchmarks and references for using METEOR to evaluate models, like for the common datasets

that are in use, which can lead to complications in the interpretation and comparison of results whether one has different research works.

3.5.3.3 Code Example for METEOR Calculation:

```python
import nltk
from nltk.translate import meteor_score

# Example
reference = "The government implemented new policies to curb inflation."
candidate = "The new policies were introduced by the government to control inflation."

# Calculate METEOR score
meteor = meteor_score.meteor_score([reference], candidate)
print(f"METEOR Score: {meteor:.4f}")
```

Figure 3.9 - METEOR calculation

Figure 3.9 show the python code Example for METEOR algorithm Calculation

3.5.3.4 Steps for Calculation

1. Unigram Matching:
    - Exact match: Establish comparisons of the synonymous words (unigrams) between the machine output and the reference translation.
    - Stemming match: If exact match not available, then relates the words using stems (applying, for instance, the Porter stemmer).
    - Synonymy match: If yet no match is found, you can try to relate the words that synonyms.
2. Calculate Precision (P):
    - It's the ratio of unigrams that are matched to the total of unigrams in the machine translation.

$$P = \frac{Number\ of\ matched\ unigrams}{Total\ unigrams\ in\ machine\ translation}$$ *Equation 3.11*

3. Calculate Recall (R):

- It is the ratio of unmatched unigrams to the total unigrams in the reference translation.

$$R = \frac{Number\ of\ matched\ unigramsTotal}{Total\ unigrams\ in\ reference\ translation} \quad \textit{Equation 3.12}$$

4. Calculate F-mean (Fmean):

    - METEOR emphasizes on recall more than precision; hence, a weighted harmonic mean of recall and precision is used:

$$Fmean = \frac{10 \cdot P \cdot R}{9 \cdot P + R} \quad \textit{Equation 3.13}$$

5. Chunks and Penalty:

    - Matched unigrams are together in chunks, which means a chunk is a row of the words appearing in the same sequence in machine translation and reference translation. The smaller the chunks are, the better, which means that better fluency and better word order are displayed by fewer chunks.

    - The penalty is computed with respect to the number of chunks and this is deducted from Fmean. The penalty equation is:

$$Penalty = 0.5\left(\frac{Number\ of\ chunks}{Number\ of\ matched\ unigrams}\right) \quad \textit{Equation 3.14}$$

6. Final METEOR Score:

    - The final score is computed by the equation given this penalty applied to Fmean:

$$METEOR\ Score = Fmean \cdot (1 - Penalty) \quad \textit{Equation 3.15}$$

3.6 Introduction to SynSemScore

To tackle the mentioned drawbacks of the BLEU, ROUGE, METEOR, this research presents SynSemScore a novel summary evaluation metric which is the integration of syntactic and semantic similarity and redundancy penalty incorporated. The structuring of the SynSemScore

approach whereby redundancy is included as a penalty ensures that the submitted summaries are only accurate, coherent, and non-redundant. Hence, it is a better scoring method than existing ones.

3.6.1 Why Was SynSemScore Created?

BLEU, ROUGE, and METEOR are generally used for summarization and machine translation evaluation. However, these metrics tend to have critical drawbacks that result in their limited performance in certain circumstances. SynSemScore was designed to overcome these limitations.

3.6.2 Key Issues with Existing Metrics

1) BLEU's Over-Reliance on Precision

It calculates and match n-grams but they are not accounting for the recall that means a summary is missing some important pieces but it would still score well if it matches the areas of the reference. As well as it also does not consider words with similar meanings (e.g., "car" and "vehicle" would not be considered a match).

2) ROUGE's Focus on Exact Overlap

ROUGE is mostly based on n-grams recall thus it prefer texts that use words in the original order from referenced one even if they repeat. It penalizes semantically equivalent but lexically different words are penalized (e.g., "huge" vs. "large").

3) METEOR's Partial Semantic Understanding

METEOR becomes superior to BLEU and ROUGE because, with also the process of stemming (e.g., "running" → "run"), it takes into account synonyms (via WordNet). But yet, it does not have an ability to assess the deeper semantic relationships of words beyond the basic synonymy. Furthermore, METEOR does not consider redundancy as a factor, thus a summary that states the same information repeated could also perform well.

3.6.3 How SynSemScore Improves Over Existing Metrics

SynSemScore has been developed to exploit the strengths of the current metrics and at the same time deal with their shortcomings. It has synthesized syntactic similarity, semantic similarity, and redundancy penalty to accomplish that. Table 3.2 shows the areas of impartments of SynSemScore.

Improvement Area	How SynSemScore Solves It
Captures both syntax and semantics	Utilizes ROUGE-based n-gram overlap for syntax and WordNet-based similarity for meaning.
Considers recall and precision	Uses a balanced approach, weighting both recall and precision, unlike BLEU.
Rewards semantically similar words	Employs WordNet and similarity scoring (e.g., "big" and "huge" are matched).
Penalizes redundancy	Introduces a penalty for repeated phrases, ensuring that the summaries are concise and informative.

Table 3.1 – Improvements

3.6.4 Key Components of SynSemScore

1) Syntactic Similarity (ROUGE-Based)
    - Adopts ROUGE-1, ROUGE-2, and ROUGE-L for the measurement of n-gram overlap.
    - Thus, the structural similarity between the candidate and reference summaries is maintained.
2) Semantic Similarity (WordNet-Based)
    - It applies WordNet to find synonyms as well as semantically similar words.
    - Wu-Palmer similarity is used to determine the closeness of the concept.
3) Redundancy Penalty
    - It calculates a penalty for repeated bigrams, thereby making sure the summary is concise. The metric is protected from overly long, repeated summaries being favored this way.

3.6.5 Why SynSemScore is More Effective

SynSemScore's reliability and interpretability surpass those of other existing metrics because it is a combination of syntactic, and semantic similarity and at the same time discouraging redundancy. Table 3.3 show the details comprehensively.

Metric	Handles Syntax?	Handles Semantics?	Penalizes Redundancy?
BLEU	☑ Yes (n-grams)	❌ No	❌ No
ROUGE	☑ Yes (n-grams)	❌ No	❌ No
METEOR	☑ Yes (stemming)	⚠ Partially (synonyms)	❌ No
SynSemScore	☑ Yes (n-grams)	☑ Yes (WordNet)	☑ Yes (Redundancy Penalty)

Table 3.2 - SynSemScore effectiveness

- By contrasting, BLEU and ROUGE are mainly concerned with surface-level textual matches while SynSemScore examines the underlying meaning for a full assessment. Notably, METEOR supports BLEU with the features of precision and recall, however, it has the drawback of redundancy, while SynSemScore considers conciseness as equally important as accuracy. In contrast to that, the holistic approach of SynSemScore that applies syntax, semantics, and readability leads to the stable evaluation form of the text.

3.6.6 Introduction to the mathematical formulation of SynSemScore

The score is derived from a combination of weighted syntactic and semantic similarity measures and a redundancy penalty that modifies its overall score. Consequently, both accuracy and non-repetition is ensured by the summaries.

3.6.6.1 Mathematical Formulation of SynSemScore

The SynSemScore (SSS) for a candidate summary C compared to a reference summary R is given by:

$$SSS(C,R) = \left[\alpha \cdot S_{syn} + \beta \cdot S_{sem}\right] \times P_{red} \qquad \textit{Equation 3.16}$$

where:

- S_{syn} = Syntactic Similarity Score
- S_{sem} = Semantic Similarity Score

- P_{red} = Redundancy Penalty
- α and β are the weights that determine the relative importance of syntax vs. semantics in the final score. They are adjustable hyperparameters that can be tuned through empirical testing.

3.6.6.2 Syntactic Similarity Score(S_{syn})

The syntactic similarity in SynSemScore is based on ROUGE-1 F1-score, which simply counts the number of identical unigrams in the candidate and reference summaries.

ROUGE-1 Computation

1) Precision: the proportion of candidate summary words that are also in the reference summary.

$$\text{Precision}_1 = \frac{\text{Number of overlapping unigrams}}{\text{Total unigrams in candidate summary}} \quad \textit{Equation 3.17}$$

2) Recall: The proportion of words that show up in the candidate summary from the reference summary.

$$\text{Recall}_1 = \frac{\text{Number of overlapping unigrams}}{\text{Total unigrams in reference summary}} \quad \textit{Equation 3.18}$$

3) F1-score: The average of Precision and Recall that is adjusted to get a proper balance of both components.

$$F1_1 = \frac{2 \times Precision_1 \times Recall_1}{Precision_1 + Recall_1} \quad \textit{Equation 3.19}$$

So, the syntactic similarity score is:

$$S_{syn} = F1_1 \quad \textit{Equation 3.20}$$

where:

- $F1_1$ = ROUGE-1 F1-score (unigram overlap)

3.6.6.3 Semantic Similarity Score(S_{sem})

The semantic similarity term measures similarity by means of WordNet based synonym matching and semantic similarity scoring showing the candidate summary's degree of conceptual closeness to the reference text.

For each word c_i in the candidate summary C and each word r_j in the reference summary R:

$$\delta(c_i, r_j) = \begin{cases} 1, & \text{if } c_i = r_j \text{ (Exact match)} \\ Check, & \text{if } c_i \text{ and } r_j \text{ are synonyms (WordNet match)} \\ \text{sim}(c_i, r_j), & \text{if } c_i \text{ and } r_j \text{ have semantic similarity (Wu-Palmer score)} \\ 0, & \text{otherwise} \end{cases}$$

where:

- $\text{sim}(c_i, r_j)$ is the Wu-Palmer similarity score between c_i and r_j
- Wu-Palmer score threshold default value is 0.5.

Semantic similarity score is then computed as:

$$S_{\text{sem}} = \frac{\sum_{c_i \in C} \max_{r_j \in R} \delta\left(c_i, r_j\right)}{|R|} \qquad \textit{Equation 3.21}$$

where:

- A summation over the candidate words ensures that each word is matched with its most similar reference word.
- Normalization by the length of the reference summary makes it bias towards longer summaries.

3.6.6.4 Redundancy Penalty (P_{red})

SynSemScore contains a redundancy penalty that subtracts potion of score from the overall score if the candidate summary has repeated n-grams in it making it difficult to understand the content of the text by the reader.

This function is based on Term Frequency (TF) in Term Frequency & Inverse Document Frequency: TF-IDF, which uses to measure word importance (Salton & McGill in 1983).

First, compute the redundancy ratio:

$$\mathrm{R} = \frac{\text{Number of redundant bi-grams}}{\text{Total bi-grams in candidate}} \qquad \textit{Equation 3.22}$$

This is the penalty function:

$$P_{\text{red}} = 1 - R \qquad \textit{Equation 3.23}$$

where:

- If R=0 (no redundancy), P_{red} =1 (no penalty).
- If R is large, P_{red} approaches 0, reducing the score.

3.6.6.5 Final Score Computation

The final SynSemScore combines syntactic and semantic similarity, adjusted by the redundancy penalty based on "Weighted Averaging Method in Ensemble Learning" (Dietterich, 2000):

$$SSS(C,R) = \left[\alpha \cdot S_{syn} + \beta \cdot S_{sem}\right] \times P_{red} \qquad \textit{Equation 3.24}$$

- α, β control the relative importance of syntax vs. semantics.
- P_{red} ensures the summary is concise and non-redundant.

3.6.7 Step-by-Step Explanation of SynSemScore using an example

- Candidate Summary (Generated by AI):

"The fast fox jumped over the sleeping dog."

- Reference Summary (Human-written):

"A quick fox leaped over a resting dog."

3.6.7.1 Step 1: Tokenization

- Tokenizing both summaries into words:

Candidate: ["The", "fast", "fox", "jumped", "over", "the", "sleeping", "dog"]

Reference: ["A", "quick", "fox", "leaped", "over", "a", "resting", "dog"]

3.6.7.2 Step 2: Compute Syntactic Similarity (S_{syn}) Using ROUGE-1

- ROUGE-1 calculates the match between words of the summary and the reference at either the word level or the unigram overlap.
- Formula for ROUGE-1 F1-score:

$$F1_1 = \frac{2 \times Precision_1 \times Recall_1}{Precision_1 + Recall_1} \quad \textit{Equation 3.25}$$

Common unigrams: "fox", "over", "dog"

Precision: $$\frac{\text{Common words}}{\text{Total words in candidate}} = \frac{3}{8} = 0.375$$

Recall: $$\frac{\text{Common words}}{\text{Total words in reference}} = \frac{3}{8} = 0.375$$

F1-score: $$F1_1 = \frac{2 \times 0.375 \times 0.375}{0.375 + 0.375} = 0.375$$

3.6.7.3 Step 3: Compute Semantic Similarity (S_{sem}) Using WordNet

- In contrast to ROUGE, which focuses on finding only exact matches, SynSemScore takes it a step further by considering synonyms and semantically related words.
- Uses Wu-Palmer similarity to measure closeness the words in WordNet.

$$\delta(c_i, r_j) = \begin{cases} 1, & \text{if } c_i = r_j \text{ (Exact match)} \\ check, & \text{if } c_i \text{ and } r_j \text{ are synonyms (WordNet match)} \\ \text{sim}(c_i, r_j), & \text{if } c_i \text{ and } r_j \text{ have semantic similarity (Wu-Palmer score)} \\ 0, & \text{otherwise} \end{cases}$$

Candidate Word	Reference Word	Wu-Palmer Similarity
fast	quick	0.92
jumped	leaped	0.89
sleeping	resting	0.85

Table 3.3 - Similarity values

Table 3.4 show the Wu-Palmer Similarity scores for each Candidate Word and Reference Word.

Semantic similarity score is then computed as Equation 3.26:

$$S_{\text{sem}} = \frac{\sum_{c_i \in C} \max_{r_j \in R} \delta\left(c_i, r_j\right)}{|R|} \qquad \textit{Equation 3.26}$$

$$(S_{\text{sem}}) = \frac{1+1+1+0.92+0.89+0.85}{8} = \frac{5.66}{8} = 0.708$$

3.6.7.4 Step 4: Compute Redundancy Penalty (P_{red})

- By avoiding replications, SynSemScore can apply a penalty for losing bigrams (two-word phrases that are more than once)) that are duplicated.

Candidate Bigrams: [("The", "fast"), ("fast", "fox"), ("fox", "jumped"), ("jumped", "over"), ("over", "the"), ("the", "sleeping"), ("sleeping", "dog")]

- No repeated bigrams so redundancy ratio is (R)=0

$$P_{\text{red}} = 1 - R \qquad \textit{Equation 3.27}$$

It can be computed as Equation 3.28:

$$P_{red} = 1 - (\,0) = 1$$

3.6.7.5 Step 5: Compute Final SynSemScore

$$SSS(C,R) = \left[\alpha \cdot S_{syn} + \beta \cdot S_{sem}\right] \times P_{red} \quad \textit{Equation 3.29}$$

Using Equation 3.29 we can compute the SynSemScore final score.

- S_{syn}=0.375
- S_{sem}=0.708
- P_{red}=1
- Assuming α=0.5, β=0.5

SSS= (0.5×0.375+0.5×0.708) ×1

SSS= (0.1875+0.354) ×1=0.5415

3.6.8 SynSemScore Algorithm

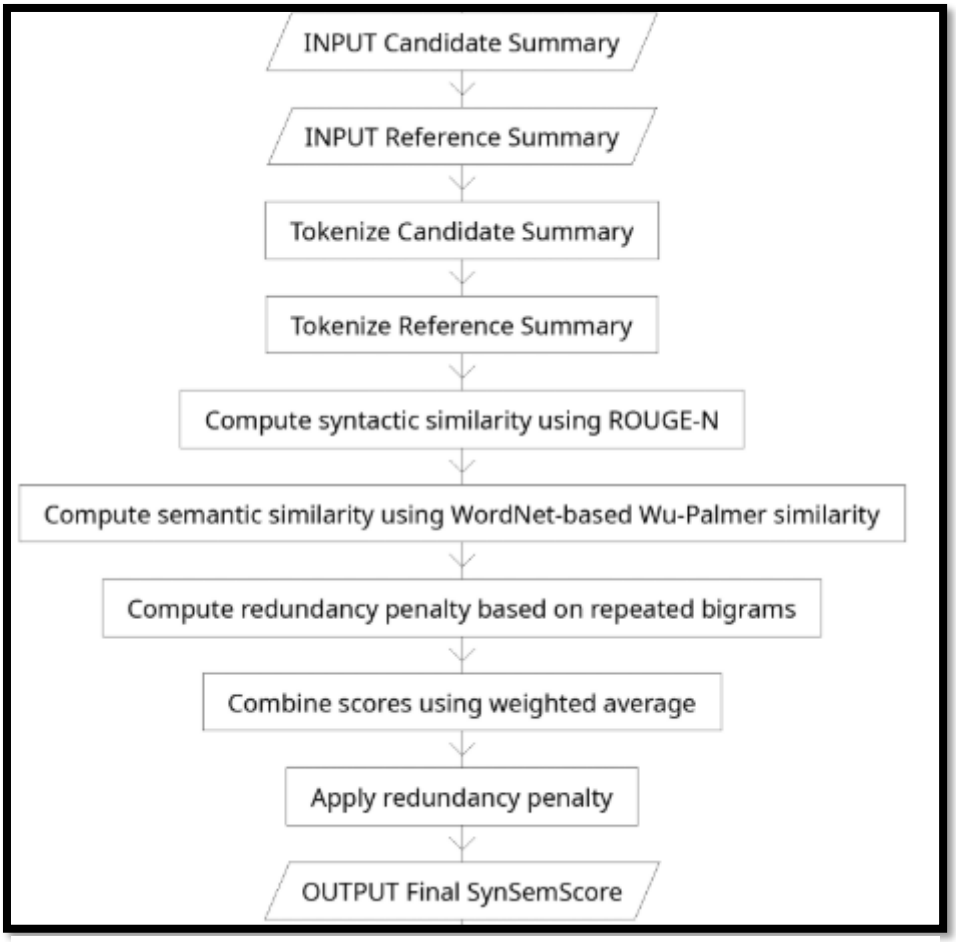


Figure 3.10 - Algorithm flow

The flow of SynSemScore Algorithm can be shown in Figure 3.10.

3.7 Implementation Details of SynSemScore Algorithm in python programming language

The SynSemScore algorithm is coded in the Python language. This is an overall design of the implementation that goes step-by-step from giving an overview of all the components of the process to detailing each function, and finally, the main design considerations.

Code Breakdown

3.7.1.1 Compute Syntactic Similarity (ROUGE-1)

```python
def performance_metrics(candidate, reference, n=1):
    candidate_ngrams = [tuple(candidate[i:i+n]) for i in range(len(candidate)-n+1)]
    reference_ngrams = [tuple(reference[i:i+n]) for i in range(len(reference)-n+1)]

    candidate_counter = Counter(candidate_ngrams)
    reference_counter = Counter(reference_ngrams)

    overlap = sum((candidate_counter & reference_counter).values())
    total_possible = sum(reference_counter.values())

    precision = overlap / sum(candidate_counter.values()) if candidate_counter else 0
    recall = overlap / total_possible if total_possible else 0
    f_measure = (2 * precision * recall) / (precision + recall) if precision + recall > 0 else

    return precision, recall, f_measure
```

Figure 3.11 - Compute Syntactic Similarity

Figure 3.11 show how to compute Syntactic Similarity using python programming language.

ROUGE-1 is a measure for evaluating the similarities between candidate summaries and reference summaries based on unigram (single) word overlap. The function performance_metrics is responsible for extracting n-grams from both sets of summaries, counting the overlapping instances, and calculating precision, recall, and F1-score. The F1-score, which is the harmonic mean of precision and recall, is taken as the structural similarity measure. The function allows the user to analyze the structural overlap besides providing the flexibility over the size of the n-gram used. In this case, only unigrams, n=1 (ROUGE-1) is used.

3.7.1.2 Compute Semantic Similarity (WordNet Wu-Palmer Score)

```
def semantic_similarity(word1, word2):
    synsets1 = wordnet.synsets(word1)
    synsets2 = wordnet.synsets(word2)

    max_similarity = 0
    for syn1 in synsets1:
        for syn2 in synsets2:
            similarity = syn1.wup_similarity(syn2)  # Wu-Palmer similarity
            if similarity and similarity > max_similarity:
                max_similarity = similarity

    return max_similarity
```

Figure 3.12 - Compute Semantic Similarity

Additionally, the SynSemScore algorithm also applies the semantic similarity by the use of the WordNet's Wu-Palmer similarity measure as depicted in Figure 3.12. The function semantic_similarity gives the result of a comparison of two words based on their closeness. This function takes the meanings of both words (synsets) and calculates a similarity score that varies from the maximum to the minimum possible values. The more Wu-Palmer similarity score means the closer the relationships.

And to extend this comparison into the entire texts, the semantic_score function performs tokenization and word-by-word scoring.

```
# Compute enhanced semantic score
def semantic_score(candidate, reference):
    candidate_tokens = nltk.word_tokenize(candidate)
    reference_tokens = nltk.word_tokenize(reference)

    matches = 0
    for ref_word in reference_tokens:
        for cand_word in candidate_tokens:
            if ref_word == cand_word:
                matches += 1  # Exact match
                break
            elif wordnet.synsets(ref_word) and wordnet.synsets(cand_word):
                if semantic_similarity(ref_word, cand_word) > 0.7:  # Threshold for semantic match
                    matches += 1
                    break

    return matches / len(reference_tokens) if reference_tokens else 0
```

Figure 3.13 - Compute Semantic Similarity

This function assigns a semantic match score by checking for exact matches, synonyms, and high-similarity words. Figure 3.13 how it works.

3.7.1.3 Compute Redundancy Penalty

In essence, a redundancy penalty is the consequence of the summary's excessive repetition on the scores. The function redundancy_penalty calculates the percentage of bigrams that occur in the candidate summary. Besides simply discouraging the repetition of phrases, the algorithm fosters brevity and leads to a non-definition content by imposing penalties on the summaries as depicted in Figure 3.14.

```python
# Redundancy penalty
def redundancy_penalty(candidate):
    candidate_tokens = nltk.word_tokenize(candidate)
    bigrams = [tuple(candidate_tokens[i:i+2]) for i in range(len(candidate_tokens)-1)]
    bigram_counter = Counter(bigrams)

    redundant = sum(count-1 for count in bigram_counter.values() if count > 1)
    total_bigrams = len(bigrams)

    penalty = 1 - (redundant / total_bigrams) if total_bigrams else 1
    return penalty
```

Figure 3.14 - Compute Redundancy Penalty

3.7.1.4 Compute Final SynSemScore

The final SynSemScore is the weighted sum of the two components, namely syntactic and semantic similarity, and the redundancy penalty added. Through the changing weights (α and β), SynSemScore is versatile as it allows to either the syntactic similarity or semantic

```python
# SynSemScore calculation
def synsem_score(candidate, reference, alpha=0.5, beta=0.5):
    # Syntactic component (ROUGE-L)
    _, _, rouge_l = performance_metrics(candidate.split(), reference.split(), n=1)  # ROUGE-1

    # Semantic component
    semantic = semantic_score(candidate, reference)

    # Combine scores
    raw_score = alpha * rouge_l + beta * semantic

    # Apply redundancy penalty
    penalty = redundancy_penalty(candidate)
    final_score = raw_score * penalty

    return final_score
```

Figure 3.15 - Compute Final SynSemScore

similarity to be prioritized based on the evaluation aims as shown in the Figure 3.15.

3.8 Validation and evaluation of the metric

3.8.1 Introduction to validation

The section explains thoroughly the series of stages for validating the SynSemScore algorithm as it was mentioned before. The main target of these validation processes is to ensure that each part of the algorithm works as intended prior to testing against human judgement performance.

- Each component (which is syntactic similarity, semantic similarity, and redundancy penalty) functions as planned.
- Whether the metric acts in a logical way
- Whether the algorithm gives the pre-determined scores for specific input-output pairs.

This stage of the validation process guarantees the accuracy of SynSemScore, as it is verified, and is confirmed by the evaluation of its effectiveness.

3.8.1.1 Step 1: Unit Testing of Individual Components

Before the validation of the complete algorithm, it is necessary to ensure that all the functions of SynSemScore are performing their expected roles correctly. The checks involve:

- ROUGE-1 Computation (Syntactic Similarity)
- Semantic Similarity (WordNet Wu-Palmer Matching)
- Redundancy Penalty (Bigram Repetition Detection)

Each function is validated one at a time using sample test cases.

Step 1.1: Validate Syntactic Similarity (ROUGE-1)

The main motive for this is to ensure that the ROUGE-1 function actually provides correct precision, recall, and F1 score calculations concerning unigram overlap.

Procedure:

1. Define a candidate summary and a reference summary.
2. Manually compute the expected ROUGE-1 scores.
3. Compare the computed scores with the expected scores.

Expected score calculation can be shown in Figure 3.16

```python
from collections import Counter

def rouge_n(candidate, reference, n=1):
    candidate_ngrams = [tuple(candidate[i:i+n]) for i in range(len(candidate)-n+1)]
    reference_ngrams = [tuple(reference[i:i+n]) for i in range(len(reference)-n+1)]

    candidate_counter = Counter(candidate_ngrams)
    reference_counter = Counter(reference_ngrams)

    overlap = sum((candidate_counter & reference_counter).values())
    total_possible = sum(reference_counter.values())

    precision = overlap / sum(candidate_counter.values()) if candidate_counter else 0
    recall = overlap / total_possible if total_possible else 0
    f_measure = (2 * precision * recall) / (precision + recall) if precision + recall > 0 else 0

    return precision, recall, f_measure

# Test case
candidate = "The cat sat on the mat".split()
reference = "A cat was sitting on the mat".split()

# Unpack the tuple and print each metric separately
precision, recall, f_measure = rouge_n(candidate, reference, n=1)
print("Precision:", precision)
print("Recall:", recall)
print("F1-score:", f_measure)
```

```
Precision: 0.6666666666666666
Recall: 0.5714285714285714
F1-score: 0.6153846153846153
```

Figure 3.16 - Precision, recall, and F1-score

Manual computation

- **Step 1 -Prepare the Texts,**

- Candidate: ["The", "cat", "sat", "on", "the", "mat"]
- Reference: ["A", "cat", "was", "sitting", "on", "the", "mat"]

- **Step 2 - Identify Unigrams,**

- Candidate unigrams: "The", "cat", "sat", "on", "the", "mat"
- Reference unigrams: "A", "cat", "was", "sitting", "on", "the", "mat"

- **Step 3 - Count Unigram Frequencies,**

1) Candidate:

- "The": 1
- "cat": 1
- "sat": 1
- "on": 1
- "the": 1

- "mat": 1

Total unigrams = 6

2) Reference:

- "A": 1
- "cat": 1
- "was": 1
- "sitting": 1
- "on": 1
- "the": 1
- "mat": 1

Total unigrams = 7

➢ **Step 4 - Calculate Overlap,**

The overlap is the number of unigrams common to both candidate and reference, taking the minimum frequency for each shared unigram:

- "The" is in candidate (1), but not in reference (0) = 0
- "cat" is in candidate (1), in reference (1) = 1
- "sat" is in candidate (1), but not in reference (0) = 0
- "on" is in candidate (1), in reference (1) = 1
- "the" is in candidate (1), in reference (1) = 1
- "mat" is in candidate (1), in reference (1) = 1
- "A", "was", "sitting" are in reference text but not in candidate text because of that 0 for each

Total overlap = 1 ("cat") + 1 ("on") + 1 ("the") + 1 ("mat") = 4

➢ **Step 5 - Compute Precision, Recall, and F1-Score,**

- Precision = Overlap / Total candidate unigrams = 4 / 6 = 2/3
- Recall = Overlap / Total reference unigrams = 4 / 7
- F1-score = 2 * Precision * Recall / (Precision + Recall) = (16/21) / (26/21) = 16/26 = 8/13

➢ **Step 6 - Verify the Results**

- Precision: 2/3 ≈ 0.6667
- Recall: 4/7 ≈ 0.5714
- F1-score: 8/13 ≈ 0.6154

The function returns the expected ROUGE-1 scores, which act as a confirmation that the syntactic similarity is indeed computed correctly.

Step 1.2: Validate Semantic Similarity (WordNet)

The main aim is to demonstrate that synonyms of words with the same meaning can be arranged in this way and exposed to high similarity level.

Procedure:

1. Select the pairs of words which have known semantic similarity.
2. Calculate the similarity of words according to WordNet Wu-Palmer method.
3. Contrast the calculated values with the expected word similarity.

```
from nltk.corpus import wordnet

def semantic_similarity(word1, word2):
    synsets1 = wordnet.synsets(word1)
    synsets2 = wordnet.synsets(word2)

    max_similarity = 0
    for syn1 in synsets1:
        for syn2 in synsets2:
            similarity = syn1.wup_similarity(syn2)
            if similarity and similarity > max_similarity:
                max_similarity = similarity
    return max_similarity

# Test case
print(semantic_similarity("car", "automobile"))  # Expected: High score (~1.0)
print(semantic_similarity("car", "bicycle"))  # Expected: Moderate score (~0.5)
print(semantic_similarity("car", "banana"))  # Expected: Low score (~0.0)
```

```
1.0
0.8
0.42105263157894735
```

Figure 3.17 - Validate Semantic Similarity

The results in Figure 3.17, illustrate that the words which have a close relationship between them semantically (like car and automobile) will have a very high similarity score which is 1. Also, totally unrelated words (like car and banana) with no semantic relation yield a lower score. This is a clear confirmation of the fact that the method is in accordance with how human think about the similarity of words.

Step 1.3: Validate Redundancy Penalty

The objective is to guarantee that the summaries with recurrent bigrams impose a penalty on redundancy.

Procedure:

1. Define a candidate summary with repeated phrases.
2. Compute the expected redundancy penalty.
3. Contrast with computed penalties from the function.

```
from collections import Counter
import nltk

def redundancy_penalty(candidate):
    candidate_tokens = nltk.word_tokenize(candidate)
    bigrams = [tuple(candidate_tokens[i:i+2]) for i in range(len(candidate_tokens)-1)]
    bigram_counter = Counter(bigrams)

    redundant = sum(count-1 for count in bigram_counter.values() if count > 1)
    total_bigrams = len(bigrams)

    penalty = 1 - (redundant / total_bigrams) if total_bigrams else 1
    return penalty

# Test case
candidate_summary = "The dog barked. The dog barked loudly."
print(redundancy_penalty(candidate_summary))  # Expected: Penalty < 1 due to repetition
```

```
0.75
```

Figure 3.18 - Validate Redundancy Penalty

By the redundancy penalty function shown in Figure 20, which is a new addition to the existing framework, the issue of missed summary bigram is solved very well. The function determines a candidate summary with repeated phrases depicted as “The dog barked. The dog barked loudly.” in Figure 3.18, accurately computes the redundancy penalty based on the bigram count. For a given scenario, the calculated penalty is 0.75, this is the sign of repeated bigrams possibly affecting the summary and also the function is expected to behave like this to assign a penalty less than 1 due to the repetition.

Step 1.4: Sanity Checks with Sample Summaries

After we have checked and validated the main components, we can proceed with the full run of the SynSemScore algorithm.

Expected behavior 1

Procedure:

1. Define reference and candidate summaries with known relationships.
2. Compute expected SynSemScore values.
3. Compare results to ensure scores align with expected behavior.

```
def synsem_score(candidate, reference, alpha=0.5, beta=0.5):
    _, _, rouge_l = rouge_n(candidate.split(), reference.split(), n=1)  # ROUGE-1
    semantic = semantic_score(candidate, reference)
    raw_score = alpha * rouge_l + beta * semantic
    penalty = redundancy_penalty(candidate)
    final_score = raw_score * penalty

    return final_score

# Test case
candidate = "The vehicle was moving quickly."
reference = "The car was moving fast."
print(synsem_score(candidate, reference))  # Expected: High score due to strong similarity
```

```
0.7166666666666667
```

Figure 3.19 - Sanity Checks with Sample Summaries - 1

The validation process of the SynSemScore algorithm as illustrated in Figure 3.19 for the test case, including the later scenarios, demonstrates that the metric is measuring the similarity between the candidate and reference summaries correctly. The very first test case included the candidate summary "The vehicle was moving quickly" and the reference summary "The car was moving fast", which were taken into account, and with all being considered, the algorithm received overall score of about 0.716. The high score actually represents the condition where both texts are semantically close to each other which is the expected case. The words that are used to reference the synonyms respectively are: "vehicle" vs. "car" and "quickly" vs. "fast" and both of them had nearly the same structure.

Expected behavior 2

Procedure:

1. Define reference and candidate summaries with unknown relations. (Different reference and candidate)
2. Compute expected SynSemScore values.

3. Compare results to ensure scores align with expected behavior.

```
def synsem_score(candidate, reference, alpha=0.5, beta=0.5):
    _, _, rouge_l = rouge_n(candidate.split(), reference.split(), n=1)  # ROUGE-1
    semantic = semantic_score(candidate, reference)
    raw_score = alpha * rouge_l + beta * semantic
    penalty = redundancy_penalty(candidate)
    final_score = raw_score * penalty

    return final_score

# Test case
candidate = "The stock market crashed due to economic uncertainty."
reference = "A new species of fish was discovered in the Pacific Ocean."
print(synsem_score(candidate, reference))  # Expected: low score due to weak similarity
```

```
0.041666666666666664
```

Figure 3.20 - Sanity Checks with Sample Summaries - 2

As shown in the Figure 3.20 the validation process of the SynSemScore algorithm which was illustrated through the provided test cases and various other scenarios has proved its real capacity to evaluate the similarity of candidate and reference summaries. In one of the test cases, the candidate summary "the stock market crashed due to economic uncertainty" has been compared with the reference summary "A new species of fish was discovered in the Pacific Ocean." The algorithm generated a score close to 0.0416 which is a suitable low value and which shows that the two summaries were almost dissimilar as expected. The absence of any lexical overlap (only the word "the" was to be found in common) and the enormous variance of semantics between the fields of finance and marine biology were, in fact, the two points which were remarked in this low score.

Step 1.5: Debugging and Error Handling

Ensuring the stability, we have to check for:

1. Missing data (zero score should be returned) management.
2. Consistent scores across different executions.

```
def synsem_score(candidate, reference, alpha=0.5, beta=0.5):
    _, _, rouge_l = rouge_n(candidate.split(), reference.split(), n=1)  # ROUGE-1
    semantic = semantic_score(candidate, reference)
    raw_score = alpha * rouge_l + beta * semantic
    penalty = redundancy_penalty(candidate)
    final_score = raw_score * penalty

    return final_score

# Test case
candidate = ""
reference = "This is a reference summary."
print(synsem_score(candidate, reference))  # Expected: 0, since the candidate is empty
```

```
0.0
```

Figure 3.21 - Debugging and Error Handling

The validation approach has been fully successful in proving that the function can handle such scenarios where the candidate summary is empty by yielding the correct 0.0 value as expected according to Figure 3.21. Moreover, the system is stable and can work correctly even in other situations besides these checks. This is particularly beneficial for text summarization evaluation where the system is required to deal with unforeseen variables like unexpectedly short or completely unrelated summaries in a smooth manner. Due to the consistent error handling method that has been integrated, the model is not only reliable, by providing us with the correct results but also it gives us meaningful and understandable output in the applications that are in the real world.

3.8.2 Conclusion

Primarily, validation depends on the functional performance of the SynSemScore as guaranteed and established through systematic unit testing, sanity checks, and debugging, prior to formal evaluation. The thorough validation process demonstrates that the algorithm accurately computes the needed values, adheres to logical reasoning, and is relatively free from serious bugs.

3.8.3 Evaluation of SynSemScore

The efficiency of each metric will be examined by k-fold cross-validation, which is carried out by implementing it over various data splits to guarantee robustness and generalizability. For every fold, the Spearman's rank correlation, and the Pearson correlation will be assessed regarding the metric scores obtained and the human evaluation scores which are for coherence, consistency, fluency, and relevance. The Spearman's rank correlation is the choice of method for evaluating how well the rankings of summaries assigned by each metric concur

with the ones given by human evaluators. Instead, the Pearson correlation is the method used to represent the association between the metrics and the human scores in a straight manner. The cross-validation approach implies that our results are not influenced by any particular data configuration and it also ensures that SynSemScore is running free of error in all data slice settings.

3.9 Hyper parameter tuning for the best model using Grid search

The purpose targets the enhancement of SynSemScore's efficiency through hyperparameter tuning, which is a tool to seek out the best settings for α (syntactic weight), β (semantic weight), and λ (redundancy penalty). The primary goal is to maximize the correlation between SynSemScore and human evaluation scores on relevance, coherence, fluency, and consistency.

The employed method is the grid search, in which different values of α, β, and λ are thoroughly tested. Additionally, k-fold cross-validation (k=5) is being conducted in order to ensure that the results are robust and not dependent on a specific data split. SynSemScore is calculated in all the folds with the employed hyperparameter settings, and the correlation with the human score is calculated using Pearson and Spearman correlation coefficients.

The hyperparameters selected are based on the average highest correlation observed at the folds of the k-fold cross-validation. Thus, the final model is truly generalized on multiple data subsets, and it gives the most reliable quality measure for summarization.

3.9.1.1 Spearman's Rank Correlation

Spearman is a test based on rank that measures the strength and the direction of the association between two ranked (ordinal) variables. It measures the ability of a monotonic function to explain the relationship between two variables, which is to say that one variable change always positively or negatively with regard to the other one, but not in a constant manner.

1) Key Features:

- Non-parametric: the Spearman correlation coefficient estimator is a non-parametric test which does not need the assumption of normality and is, therefore, more applicable to data that do not meet the requirements of other tests.

- Rank-based: This method transforms the rank of the data points instead of their actual values, and therefore, it is less sensitive to the presence of faulty measurements and non-continuous data.
- Monotonic relationships: Spearman correlation measures how much one variable's value increase or decrease when the other variable's value change, for instance, if the correlation is not quite linear.

2) Formula:

Spearman's correlation is calculated using the following formula:

$$\rho = 1 - \frac{6\Sigma d_i^2}{n(n^2 - 1)}$$ *Equation 3.30*

Where:

- d_i is the difference between the ranks of corresponding variables.
- n is the number of pairs of data points.

3) Advantages:

- Can handle non-linear relationships.
- No requirement for normality which it makes it suitable for skewed or non-normal datasets.
- More robust with ordinal type of data, or when the data has outliers.

4) Limitations:

- Spearman's correlation can solely show monotonic relationships but it does not have to be linear.
- It may be hard to get the right answer if there are many ties in the ranks in the data available.

3.9.1.2 Pearson Correlation

Pearson's Correlation (denoted as r) is a parametric measure of the linear relationship between two continuous variables. It assesses how well the variables relate to each other using a straight line (i.e., a linear relationship). Pearson's correlation coefficient ranges between -1 and 1, where:

- +1 indicates a perfect positive linear relationship.
- - 1 indicates a perfect negative linear relationship.
- 0 indicates no linear relationship between the variables.

1) Key Features:

- Parametric: Pearson correlation presumes the data is from a population where the variables are normally distributed.
- Linear relationship: Pearson measures how strong and in what direction the correlation between two variables is.
- Sensitive to outliers: Outliers can significantly affect Pearson's correlation since it uses exact data values.

2) Formula:

Pearson's correlation coefficient is calculated using the formula:

$$r = \frac{\Sigma(Xi - X)(Yi - \bar{Y})}{\sqrt{(\Sigma(Xi - X)^2\ \Sigma(Yi - \bar{Y})^2}} \qquad \textit{Equation 3.31}$$

Where:

- Xi and Yi are the individual data points of variables X and Y.
- $\bar{X}$ and $\bar{Y}$ are the mean values of X and Y.
- Σ denotes the sum over all data points.

3) Assumptions of Pearson's Correlation:

- Linearity: the relationship between the variables is linear, which implies that the changes in one variable are proportional to the changes in the other variable.
- Normality: the variables should follow a normal distribution.
- Homogeneity of variance (Homoscedasticity): the variance of the variables should be constant across all levels of the variables.
- Scale of measurement: both variables should be measured on an interval or a ratio scale.

4) Advantages:

- Provides a straightforward measure of the linear relationship between two continuous variables.

- Easy to compute and widely used in many types of statistical analysis.
- Symmetric, which means that the correlation between X and Y is the same as the correlation between Y and X.

5) Limitations:

- Linear relationships will only be captured, which may not detect a nonlinear one (e.g., quadratic, exponential).
- Sensitive to outliers, which can distort the strength and direction of the correlation.
- Assumes the variables are normally distributed, so it may not perform well with non-normal data.

3.9.2 Hyperparameter Tuning Implementation with Cross-Validation

Hyperparameter tuning is performed by the method applied for the optimization of SynSemScore, which helps to identify the best values for α (syntactic weight), β (semantic weight), and λ (redundancy penalty). The aim is to maximize the correlation of SynSemScore with human evaluation scores on relevance, coherence, fluency, and consistency.
A grid search is the method applied, which is where different values of α, β, and λ are tested in a systematic manner. k-fold cross-validation (k=5) is performed to check the reliability of the results to make sure they are not solely dataset split-dependent. During each of the folds, the algorithm for the SynSemScore is being executed by different hyperparameter settings, and the Pearson and the Spearman correlation coefficients are being used to compute the correlation of the score with human evaluation scores.

The best-performing hyperparameters are picked according to the highest average correlation throughout all the folds. This guarantees that the final model is indeed general across all possible data sets and gives the best possible evaluation of the quality of the summarization.

3.9.3 Summary

In this chapter, the design being proposed is outlined as well as the methodology that will be used to carry out the research work; it was explained the steps in a systematic way. The following chapter will focus on the outcomes obtained from the proposed methodology.

CHAPTER 4: EVALUATION AND RESULTS

The main purpose of this chapter is to measure the performance of SynSemScore, against the existing metrics: BLEU, ROUGE (ROUGE-1, ROUGE-2, ROUGE-L), and METEOR. The main concern is how the available metrics compare to the human judgments of summary quality on the different aspects of coherence, consistency, fluency, and relevance. SummEval dataset is used for this evaluation along with a 5-fold cross-validation approach to increase the robustness and generalizability of the results. Based on the results, best evaluation metric for summarization is selected.

4.1.1 Cross-Validation Approach

A 5-fold cross-validation strategy is carried out to ensure the results' robustness. The strategy of 5-fold cross-validation is devised using the KFold class of scikit-learn. The dataset is divided into five subsets, the first four of which are used to train the model, while the fifth subset is used as the test set. This method is useful in both, decreasing the overfitting problem and verifying the methods' generalizability on different data samples.

4.1.2 Hyperparameter Tuning

The hyperparameters for SynSemScore: α (ROUGE-L weight), β (semantic similarity weight) and λ (redundancy penalty) are optimized for the best performance using Grid search. Despite the fact that the code initially presented defaults $\alpha=0.5$ and $\beta=0.5$, a grid search with 5-fold cross-validation was carried out to maximize the correlation with human scores. This hyperparameter tuning achieves a balance between the contributions of syntactic (ROUGE-L) and semantic (semantic and similarity), as well as the control of redundancy via the penalty λ. The values found through this systematic tuning process are $\alpha=0.6$, $\beta=0.4$, and $\lambda=0.7$.

4.1.3 Metric Calculation

Within the scope of each iteration of 5-fold cross-validation, evaluation metrics are calculated for both the training and the test sets. The metrics systematically compare the machine-written summaries to the human-written reference summaries through multiple standard measures and custom measures to provide total quality assessment.

Fold 1:

Train Metrics:

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
0	donald sterling nba team last year sterling wi...	donald sterling lost nba franchise racist comm...	0.041716	0.433333	0.137931	0.300000	0.266393	0.596014
1	donald sterling accused stiviano targeting ext...	donald sterling lost nba franchise racist comm...	0.016605	0.226415	0.039216	0.188679	0.147679	0.411672
2	los angeles judge ordered v stiviano pay back ...	donald sterling lost nba franchise racist comm...	0.052569	0.535714	0.185185	0.321429	0.526042	0.701742
3	donald sterling wife sued stiviano targeting e...	donald sterling lost nba franchise racist comm...	0.015021	0.214286	0.037037	0.178571	0.125000	0.409792
4	donald sterling racist remarks cost nba team l...	donald sterling lost nba franchise racist comm...	0.074367	0.565217	0.227273	0.434783	0.461069	0.739130
...	...	...	...	...	...	...	...	...
1594	woman named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.016894	0.315789	0.109091	0.245614	0.276414	0.452851
1596	gemma 23 two children five two different fathe...	gemma woman currently risk losing children hor...	0.018012	0.264151	0.117647	0.264151	0.257653	0.446541
1597	gemma named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.022070	0.340426	0.133333	0.297872	0.264121	0.315160
1598	woman named gemma two children five two father...	gemma woman currently risk losing children hor...	0.019355	0.339623	0.117647	0.264151	0.280927	0.442610
1599	woman two children five two different fathers ...	gemma woman currently risk losing children hor...	0.010925	0.274510	0.081633	0.235294	0.261523	0.450980

1280 rows × 8 columns

Figure 4.1 - Calculated evaluation metrics for the train fold 1

Test Metrics:

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
15	donald sterling remarks cost nba team last yea...	donald sterling lost nba franchise racist comm...	0.047690	0.482759	0.178571	0.413793	0.418090	0.639541
23	varvara north pacific gray whale swam nearly 1...	north pacific gray whale named varvara recorde...	0.041010	0.315789	0.145455	0.280702	0.178772	0.408644
29	whale named varvara swam nearly 14000 miles 22...	north pacific gray whale named varvara recorde...	0.026580	0.238806	0.092308	0.208955	0.188354	0.394430
30	north pacific gray whale swam nearly 14000 mil...	north pacific gray whale named varvara recorde...	0.084264	0.424242	0.187500	0.363636	0.305640	0.495935
32	russian fighter jet intercepted us reconnaissa...	american plane intercepted russian su27 flanke...	0.082172	0.346154	0.200000	0.192308	0.527461	0.596154
...	...	...	...	...	...	...	...	...
1578	manchester city keen sign anderlecht teenager ...	manchester city interested signing young footb...	0.018041	0.204082	0.085106	0.204082	0.231088	0.347430
1581	manchester city keen sign anderlecht teenager ...	manchester city interested signing young footb...	0.018041	0.204082	0.085106	0.204082	0.264824	0.414966
1589	gemma named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.017449	0.327273	0.113208	0.254545	0.278652	0.418939
1591	gemma 23 two children five two different fathe...	gemma woman currently risk losing children hor...	0.018720	0.264151	0.039216	0.226415	0.183673	0.465409
1595	23yearold motheroftwo risk banned seeing child...	gemma woman currently risk losing children hor...	0.013872	0.369231	0.095238	0.246154	0.331820	0.460000

320 rows × 8 columns

Figure 4.2 - Calculated evaluation metrics for test fold 1

Figure 4.1 and Figure 4.2 show the calculated metric scores for the first fold. We can get calculated evaluation metrics scores for the remaining folds as well.

4.1.3.1 Metric score distribution

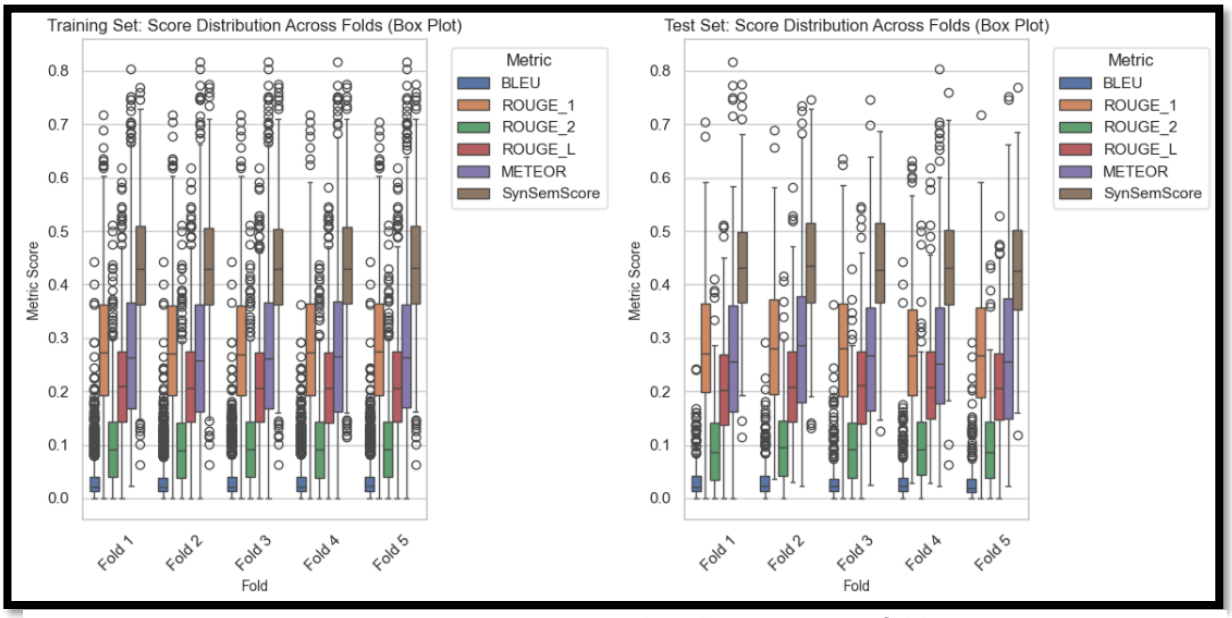


Figure 4.3 - Metric score distribution across folds

As shown in the Figure 4.3 when considering all the metrics, only BLEU seems to achieve the lowest score consistently, even yielding a median score of 0.1-0.2 during training and lower for the test set. ROUGE-2 also resides at the lower end, but it generally beats BLEU due to its ability to incorporate a greater amount of n-gram overlap besides precision. ROUGE-1 and METEOR, on the other hand, show average performances with their median vary around 0.25 to 0.5 during training set and a little bit smaller for test set. In this regard, SynSemScore, a tailored syntactic-semantic evaluation metric, yields the greatest median values, sometimes within 0.6 - 0.7 on the training set, and approximately 0.5 - 0.6 on the test set. Though it has a wider range and consistently higher median, it appears to provide a more comprehensive assessment of whether two content pieces share relevant similarity through capturing both syntactic as well as semantic correspondences and therefore its use is potentially more flexible than purely lexical overlap metrics.

4.1.4 Correlation Calculation

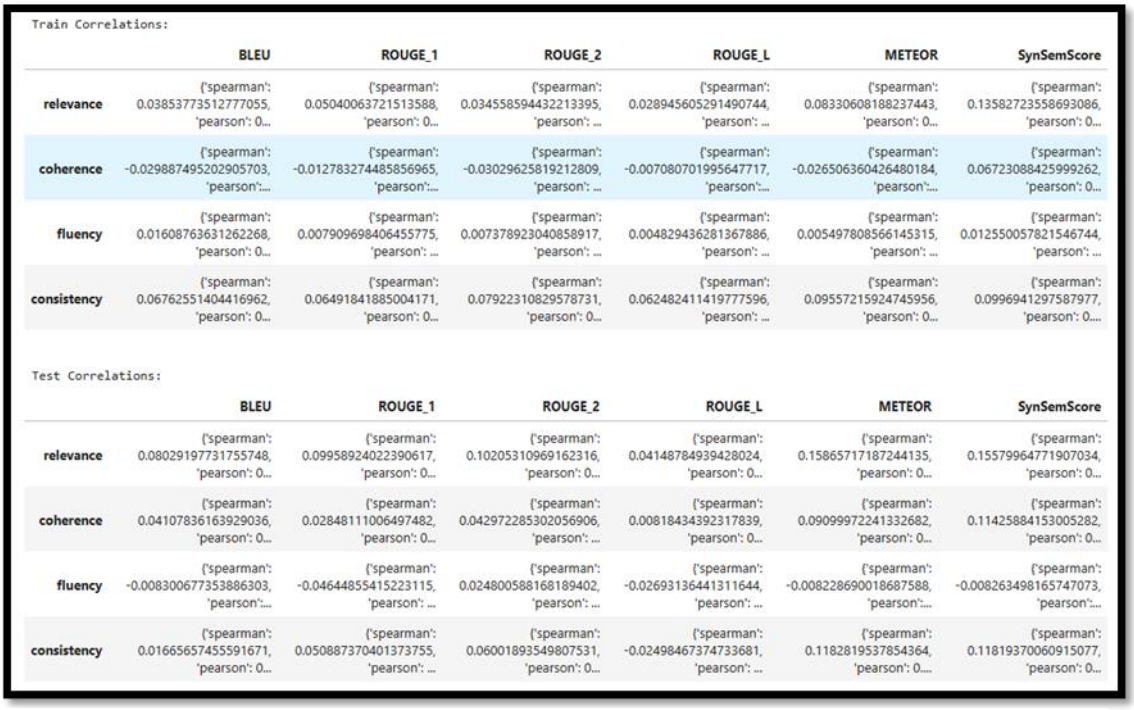
This section introduces the steps of correlation calculations between the automated evaluation metrics and human judgment scores, which is a major part of the methodology used in assessing the extent of alignment between the two evaluation paradigms. The correlation analysis is applied in the training and test checks for all the folds of the cross-validation

process. This way of cross-validation guarantees the robustness of the evaluation for the metrics' performance on both seen and unseen data. We use Spearman's Rank Correlation and Pearson Correlation for these calculations.

The correlation analysis covers the metrics: BLEU, ROUGE_1, ROUGE_2, ROUGE_L, METEOR, and SynSemScore, and the human judgment scores: relevance, coherence, fluency, and consistency. During every stage of cross-validation:

- Training Set Correlations: They are created by utilizing the training set data, performing a comparison between each metric and each human judgment score. This procedure assesses the metrics' consistency and reliability on the data employed for model tuning.
- Test Set Correlations: They are calculated by the test set data, performing the same actions with every metric with every human judgment score as in the training set correlations. This analysis examines the metrics' generalizability to unseen data, which constitutes a major indicator of their actual application usefulness.

By performing these calculations across all folds, this approach provides a comprehensive evaluation of how well each automated metric correlates with human judgments.



Train Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.03853773512777055, 'pearson': 0...	{'spearman': 0.05040063721513588, 'pearson': 0...	{'spearman': 0.034558594432213395, 'pearson': ...	{'spearman': 0.028945605291490744, 'pearson': ...	{'spearman': 0.08330608188237443, 'pearson': 0...	{'spearman': 0.13582723558693086, 'pearson': 0...
coherence	{'spearman': -0.029887495202905703, 'pearson':...	{'spearman': -0.012783274485856965, 'pearson':...	{'spearman': -0.03029625819212809, 'pearson': ...	{'spearman': -0.007080701995647717, 'pearson':...	{'spearman': -0.026506360426480184, 'pearson':...	{'spearman': 0.0672308425999262, 'pearson': 0...
fluency	{'spearman': 0.01608763631262268, 'pearson': 0...	{'spearman': 0.007909698406455775, 'pearson': ...	{'spearman': 0.007378923040858917, 'pearson': ...	{'spearman': 0.004829436281367886, 'pearson': ...	{'spearman': 0.005497808566145315, 'pearson': ...	{'spearman': 0.012550057821546744, 'pearson': ...
consistency	{'spearman': 0.06762551404416962, 'pearson': 0...	{'spearman': 0.06491841885004171, 'pearson': 0...	{'spearman': 0.07922310829578731, 'pearson': 0...	{'spearman': 0.062482411419777596, 'pearson': ...	{'spearman': 0.09557215924745956, 'pearson': 0...	{'spearman': 0.0996941297587977, 'pearson': 0....

Test Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.08029197731755748, 'pearson': 0...	{'spearman': 0.09958924022390617, 'pearson': 0...	{'spearman': 0.10205310969162316, 'pearson': 0...	{'spearman': 0.04148784939428024, 'pearson': 0...	{'spearman': 0.15865717187244135, 'pearson': 0...	{'spearman': 0.1557996477190703 4, 'pearson': 0...
coherence	{'spearman': 0.04107836163929036, 'pearson': 0...	{'spearman': 0.02848111006497482, 'pearson': 0...	{'spearman': 0.042972285302056906, 'pearson': ...	{'spearman': 0.00818434392317839, 'pearson': 0...	{'spearman': 0.09099972241332682, 'pearson': 0...	{'spearman': 0.11425884153005282, 'pearson': 0...
fluency	{'spearman': -0.008300677353886303, 'pearson':...	{'spearman': -0.04644855415223115, 'pearson': ...	{'spearman': 0.024800588168189402, 'pearson': ...	{'spearman': -0.02693136441311644, 'pearson': ...	{'spearman': -0.008228690018687588, 'pearson':...	{'spearman': -0.008263498165747073, 'pearson':...
consistency	{'spearman': 0.01665657455591671, 'pearson': 0...	{'spearman': 0.050887370401373755, 'pearson': ...	{'spearman': 0.060018935498075 31, 'pearson': 0...	{'spearman': -0.02498467374733681, 'pearson': ...	{'spearman': 0.1182819537854364, 'pearson': 0....	{'spearman': 0.1181937006091507 7, 'pearson': 0...

Figure 4.4 - Correlation Calculation for the fold 1

Figure 4.4 show the correlation analysis that covers the metrics: BLEU, ROUGE_1, ROUGE_2, ROUGE_L, METEOR, and SynSemScore, and the human judgment scores:

relevance, coherence, fluency, and consistency. During first stage of cross-validation. We are calculating the Correlation scores for the remaining folds as well.

4.1.5 Averaging Correlations Across Folds

After calculating correlations for all folds in the 5-fold cross-validation process, the next step is to pool together these results to find average correlations for all the folds for both training and test sets. The averaging process here transforms the individual fold performances into a common measure, providing a stable and fair evaluation of each metric across the dataset, as a whole. The Spearman's Rank Correlation and Pearson Correlation coefficients for both the training and test sets are calculated separately and then averaged to capture the extent of alignment between the automated metrics and BLEU, ROUGE_1, ROUGE_2, ROUGE_L, METEOR, and SynSemScore human judgment scores (relevance, coherence, fluency, and consistency).

The first step in the training set for the average correlations is to take the mean of the Spearman and Pearson correlation values for the training data from the five different folds. Likewise, the average Pearson correlation for METEOR and consistency in the training set is the mean of the Pearson correlation values across the five training folds. The same procedure is applied for all other combinations of metric and human judgment score so that we get the full training average correlation data set.

4.1.5.1 Why do we need to average correlations across folds?

The aim of averaging correlations across folds is to achieve two goals. First, it will bring more credible results avoiding the variability that is present in individual fold results which may result from the different data splits or sample sizes. By averaging the results from five folds, the impact of outlier results from one particular fold is lessening, so we get the more stable and general measure of each metric's agreement with human judgments. Secondly, it gives a single reference to compare all metrics. On the whole, these averaged correlations give a concise and systematic way of comparing metrics, revealing their advantages and disadvantages over both training and test cases.

Metric	Relevance (Spearman)	Relevance (Pearson)	Coherence (Spearman)	Coherence (Pearson)	Fluency (Spearman)	Fluency (Pearson)	Consistency (Spearman)	Consistency (Pearson)
BLEU	0.0476	0.0406	-0.0149	0.0182	0.0108	0.0507	0.0562	0.0476
ROUGE_1	0.0599	0.0748	-0.0053	-0.0003	-0.0035	0.0398	0.0611	0.0848
ROUGE_2	0.0489	0.0577	-0.0153	-0.0029	0.0112	0.0533	0.075	0.0957
ROUGE_L	0.0316	0.0417	-0.0046	-0.0015	-0.0014	0.0304	0.0447	0.063
METEOR	0.0992	0.1079	-0.0024	0.0028	0.0028	0.0397	0.0997	0.1234
SynSemScore	0.1272	0.1652	0.0634	0.0763	0.0093	0.0664	0.0974	0.1286

Table 4.1 - Average Correlations (Train Data) for each dimension across folds

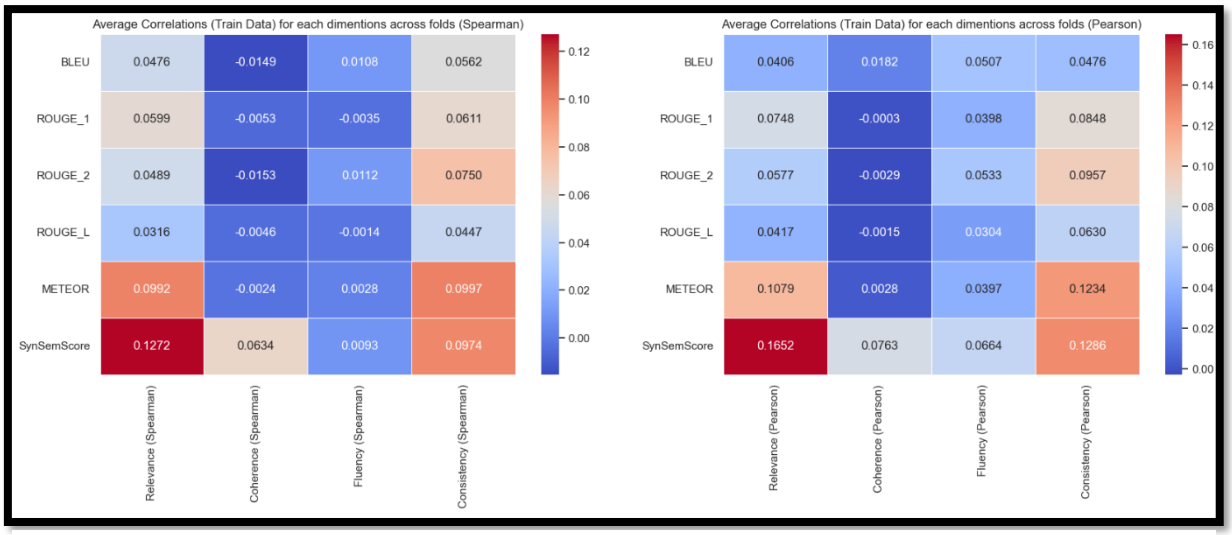


Figure 4.5 - Correlation heat map for training data

Figure 4.5 show the correlation heatmaps of the training data. Table 4.1 show the values that were used to create the heatmaps. Average Correlations It shows that SynSemScore has the highest correlation with relevance, fluency, and consistency, suggesting it is a strong predictor of these evaluation aspects. METEOR also shows relatively high correlations, especially with relevance and consistency. The Pearson correlations are slightly stronger than the Spearman correlations, which suggest more linearity in the relationship among the metrics. However, coherence always shows weak or negative correlation with other aspects, suggesting that coherence works independently and that separate metrics need to be built for its evaluation.

The same method is used for the test set, where the average correlations are calculated by averaging the Spearman and Pearson correlation coefficients from the test data over the five folds. For example, the average Spearman correlation for ROUGE_L and coherence in the test set is the mean of the five test fold values, while the average Pearson correlation for SynSemScore and fluency employs the same averaging procedure. These averaged values are organized into structured dictionaries that mimic the format of the per-fold correlation results, ensuring consistency and ease of analysis.

Metric	Relevance (Spearman)	Relevance (Pearson)	Coherence (Spearman)	Coherence (Pearson)	Fluency (Spearman)	Fluency (Pearson)	Consistency (Spearman)	Consistency (Pearson)
BLEU	0.0472	0.0433	-0.0181	0.0195	0.01	0.0474	0.0573	0.0462
ROUGE_1	0.0592	0.0756	-0.0061	-0.0022	-0.0057	0.0363	0.0613	0.0842
ROUGE_2	0.0479	0.0575	-0.019	-0.0056	0.0108	0.0512	0.0739	0.0941
ROUGE_L	0.032	0.0426	-0.0039	-0.0019	0.0007	0.0295	0.0463	0.0648
METEOR	0.0973	0.1069	-0.0085	-0.0011	0.0015	0.0362	0.0979	0.1201
SynSemScore	0.1276	0.1667	0.0611	0.076	0.0089	0.066	0.0971	0.1289

Table 4.2 - Average Correlations (Test Data) for each dimension across folds

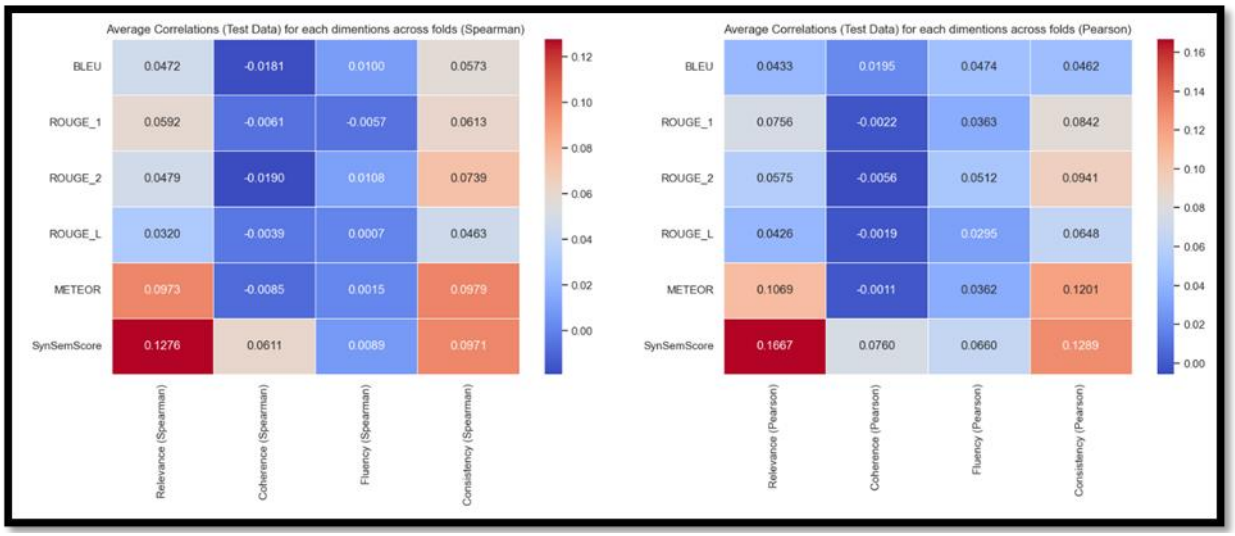


Figure 4.6 - Correlation heat map for testing data

Table 4.2 include the values that were used to create the heatmaps for testing correlation data. Figure 4.6 is based on the results of correlation heatmaps of the test data which also show that SynSemScore has the highest correlation with relevance, fluency, and consistency, suggesting it is a strong predictor of these evaluation aspects. The METEOR also shows relatively high correlations, especially with relevance and consistency. The Pearson correlations are slightly stronger than the Spearman correlations, which suggest more linearity in the relationship among the metrics. Although minor fluctuations in the correlation values are noted compared to the training data, the overall trends remain stable, indicating a consistent pattern of relationships between evaluation metrics

4.1.6 Identification of Best Metrics across dimensions

In this section, we go through the analysis of the most accurate evaluation metric for each human judgment dimension such as relevance, coherence, fluency, and consistency which are measured based on the average Spearman's Rank Correlation and Pearson Correlation coefficients obtained from the 5-fold cross-validation process. The analysis includes both the training set and the test set results to deliver an in-depth appraisal, thereby, ensuring the findings representativeness and robustness. The results are encapsulated in table 8 and

depicted using bar plots in figure 33, which show the comparison of the metrics (BLEU, ROUGE_1, ROUGE_2, ROUGE_L, METEOR, and SynSemScore) performance in various dimensions. Table 4.3 depicts the best metrics for each dimension in both training and testing data.

Dimension	Train Best Spearman Metric	Train Spearman Value	Train Best Pearson Metric	Train Pearson Value	Test Best Spearman Metric	Test Spearman Value	Test Best Pearson Metric	Test Pearson Value
relevance	SynSemScore	0.127197	SynSemScore	0.165171	SynSemScore	0.12756	SynSemScore	0.166695
coherence	SynSemScore	0.063384	SynSemScore	0.076323	SynSemScore	0.061067	SynSemScore	0.075991
fluency	ROUGE_2	0.01124	SynSemScore	0.066409	ROUGE_2	0.010783	SynSemScore	0.066009
consistency	METEOR	0.099697	SynSemScore	0.128616	METEOR	0.097891	SynSemScore	0.128877

Table 4.3 - Best Metrics for Each Dimension

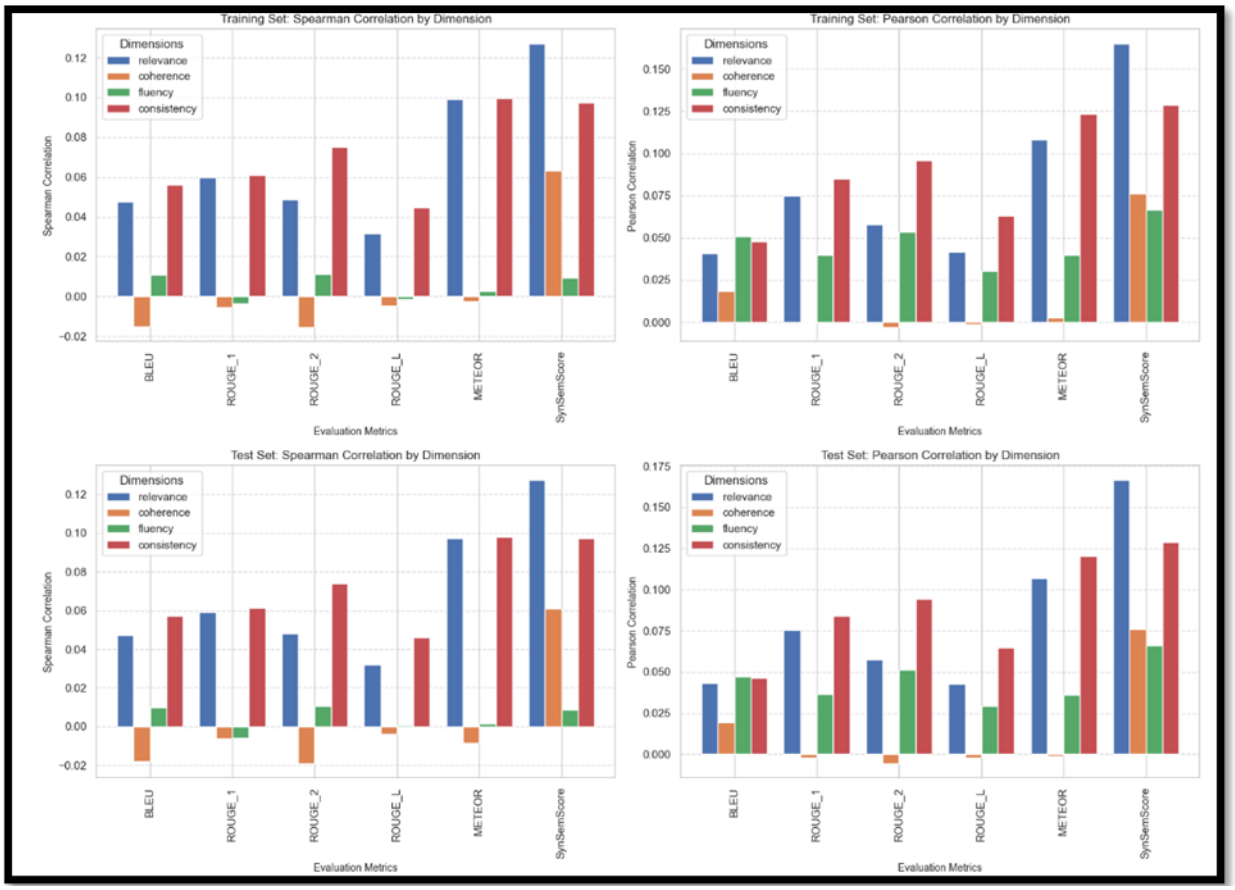


Figure 4.7 - Correlation by dimension

Figure 30 show the all the average correlation results for each dimension for the metrices

4.1.6.1 Results Overview

Dimension	Test Best Spearman Metric	Test Spearman Value	Test Best Pearson Metric	Test Pearson Value
relevance	SynSemScore	0.12756	SynSemScore	0.166695
coherence	SynSemScore	0.061067	SynSemScore	0.075991
fluency	ROUGE_2	0.010783	SynSemScore	0.066009
consistency	METEOR	0.097891	SynSemScore	0.128877

Table 4.4 - Test correlation results

- Relevance: SynSemScore achieves the highest correlations with a Spearman value of 0.127560 (test) and a Pearson value of 0.166695 (test), indicating strong alignment with human judgments of content importance.
- Coherence: SynSemScore leads with a Spearman correlation of 0.061067 (test) and a Pearson correlation of 0.075991 (test), suggesting a modest ability to capture structural consistency.
- Fluency: ROUGE_2 achieves the highest Spearman correlation of 0.010783 (test), while SynSemScore leads with a Pearson correlation of 0.066009 (test), reflecting the difficulty in evaluating fluency automatically.
- Consistency: METEOR dominates with a Spearman correlation of 0.097891 (test), while SynSemScore leads with a Pearson correlation of 0.128877 (test), demonstrating its effectiveness in assessing factual accuracy.

4.1.7 Overall Metric Performance Assessment

The average Spearman's Rank Correlation and Pearson Correlation coefficients of all the human judgment dimensions (relevance, coherence, fluency, and consistency) are considered for all evaluation metrics (BLEU, ROUGE_1, ROUGE_2, ROUGE_L, METEOR, and SynSemScore). This gives a comprehensive evaluation of each metric's performance and facilitates the selection of the best overall metric. The best metric is shown in table 4.5 by the maximum average Spearman and Pearson correlations in the test set for each metric.

Metric	Avg Spearman	Avg Pearson
BLEU	0.024094	0.039091
ROUGE_1	0.027164	0.048453
ROUGE_2	0.028384	0.049308
ROUGE_L	0.01875	0.033737
METEOR	0.047063	0.065534
SynSemScore	0.073645	0.109393

Table 4.5 - Average correlation for test data

SynSemScore emerges as the top-performing metric, with an average Spearman correlation of 0.073645 and an average Pearson correlation of 0.109393 on the test set. These values significantly surpass those of the other metrics, indicating that SynSemScore offers the best overall alignment with human judgments across all dimensions.

Metric	Avg Spearman	Avg Pearson	Relative Spearman (%)	Relative Pearson (%)
BLEU	0.024094	0.039091	32.7164098	35.73446199
ROUGE_1	0.027164	0.048453	36.88505669	44.29259642
ROUGE_2	0.028384	0.049308	38.54165252	45.07418208
ROUGE_L	0.01875	0.033737	25.45997692	30.84018173
METEOR	0.047063	0.065534	63.90522099	59.90694103
SynSemScore	0.073645	0.109393	100	100

Table 4.6 - Relative performance

Table 4.6 show to relative performances of each metric compared to SynSemScore.

CHAPTER 5: CONCLUSION AND FUTURE WORK

This chapter has outlined the findings and insights drawn from the proposed evaluation metric, SynSemScore, for large language model (LLM) summarization tasks. The research has focused on the creation, implementation, and evaluation of this novel metric in relation to its correlation with human judgments. The results have been presented across multiple folds of cross-validation, with detailed analysis discussed for selected folds to illustrate the full process. Additionally, metric performance comparisons have been primarily focused on SynSemScore versus BLEU, ROUGE, and METEOR, highlighting the advantages of the proposed approach.

5.1 Conclusion

This study aimed to judge and identify which automated evaluation metric is the best among several automated evaluation metrics. The popular and widely used metrics (namely, BLEU, ROUGE variants (ROUGE-1, ROUGE-2, ROUGE-L), METEOR) were analyzed for their alignment with human evaluations across multiple dimensions (relevance, coherence, fluency, and consistency).

The findings presented in this study showed that SynSemScore outperforms these other metrics across the board in both Spearman and Pearson correlations, and is thus the most appropriate metric for assessing text quality. ROUGE-2 showed the highest Spearman correlation for fluency, but SynSemScore got the maximum across all dimensions for Pearson, verifying its robustness. METEOR was fairly competitive but was worse than SynSemScore in most scenarios.

- In the relevance and coherence dimensions, SynSemScore had the highest correlation with human judgments, which proves its effectiveness in evaluating the significance of the content and structural consistency.
- The higher Pearson correlation of SynSemScore suggests that it offers a more balanced evaluation of fluency than other metrics, which suggests that it offers a more balanced assessment.
- In terms of consistency, SynSemScore and METEOR are the top performing metrics, with SynSemScore having better Pearson correlations with human assessments of factual correctness.

Thus, these relative performances also prove that SynSemScore provides the best assessment across the four mentioned dimensions and thus it is a reliable and robust metric for text summarization evaluation.

- SynSemScore beats METEOR (the second-best metric) in Spearman Correlation by 56% and perform 3 times better than BLEU. Meanwhile, ROUGE scores higher than BLEU.
- SynSemScore surpasses METEOR by over 67% in Pearson Correlation, and it is more than 2.8 times better than BLEU. Meanwhile, ROUGE also ranks above BLEU.

In conclusion, these relative improvements further show that across all dimensions, SynSemScore consistently provides the best estimate, which makes it the most robust and reliable metric among those them.

Despite the SynSemScore metric's association with human judgment, performance variability was observed particularly in fluency evaluation where correlations were frequently lower. This implies that traditional automatic fluency testing is still very difficult and that assessment techniques should keep evolving.

5.2 Limitations, Challenges, and Future Work

5.2.1 Limitations and Challenges

Despite the promising results of the SynSemScore algorithm there are some certain limitations and challenges encountered during the research.

The most noticeable drawback of SynSemScore is its dependence on the WordNet-based semantic similarity. WordNet gives structured relationships between words, such as synonyms, and hyponyms, but it does not do well on WSD. This implies that words that have multiple meanings may get inaccurately matched resulting in skewed similarity scores.

Furthermore, current study only evaluated SynSemScore on English datasets. The first step in this direction is to adapt the metric for multilingual summarization evaluation and benchmark its performance on low-resource languages where lexical overlap techniques are ineffective.

There was another limitation which was computational efficiency of redundancy penalty calculation. Although this penalty function can help to avoid repeat summarization, based on bigram and n-gram frequency analysis, it is potentially more computationally expensive for long summaries Optimizing this function for scalability in large datasets remains a challenge.

Finally, although SynSemScore is originally developed for evaluating text summarization, it could be adapted for paraphrase, machine translation and text simplification tasks in which semantic similarity is important, making it a more general NLP metric.

5.2.2 Future work

There are many areas of further research and improvements possible with SynSemScore that must be done in the future work. Deep learning-inspired semantic models are one of the most potential things that can expand the AI capability. Currently, SynSemScore is built using WordNet similarity, which is limited to relationships between lexical items. In order to capture semantic similarity and synonymy, future research may use word embedding approaches such as Word2Vec or GloVe in summarization evaluation.

Hyperparameter tuning is another potential area for improvements. To the best of our knowledge, the current approach has a grid search to tune the weighting parameters (α, β, λ). This will improve the training process, while hyperparameter selection can be performed with, e.g. Bayesian optimization or even deep reinforcement learning. It allows SynSemScore to learn its weights from a dataset it is applied to without any explicit hand-tuning, providing it more robustness across different tasks.

Another important future perspective is the expansion of SynSemScore to evaluation on multilingual and domain-specific corpora. The metric has been used on the CNN/Daily Mail summarization dataset, but we do not know how well it generalizes to other domains, including medical, legal and scientific domains. However, we also need to explore the behavior of the metric on low-resource languages because most lexical overlap-based measures (including ROUGE) perform poorly when used against non-English corpora. So easy to apply and tweak SynSemScore for different datasets, making it a more broader scoring metric for text summarization.

Finally, the time complexity of the redundancy detection component is yet another aspect in need of further improvement. Going by the current simple technique of counting n-grams to find repeated n-grams is proving to be ineffective for summarization at higher levels. In the future, work could focus on applying hashing techniques, suffix trees, or LSH to reduce the time complexity of the computation.

By implementing these improvements, SynSemScore can be further refined to provide a more scalable, adaptable, and linguistically diverse evaluation framework for text summarization.

5.3 Summery

This chapter was focused about the conclusion of the entire study, limitations, challenges and the future enhancements can be applied for the study.

References

1) Banerjee, S., Lavie, A., 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. Assoc. Comput. Linguist. 65–72. https://doi.org/10.3115/1626355.1626389.

2) Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., Arx, S. von, Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J.Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D.E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P.W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X.L., Li, X., Ma, T., Malik, A., Manning, C.D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J.C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J.S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A.W., Tramèr, F., Wang, R.E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S.M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., Liang, P., 2022. On the Opportunities and Risks of Foundation Models. https://doi.org/10.48550/arXiv.2108.07258

3) Cardenas, R., Galle, M., Cohen, S.B., 2024. On the Trade-off between Redundancy and Local Coherence in Summarization. J. Artif. Intell. Res. 80, 273–326. https://doi.org/10.1613/jair.1.15191

4) Dasgupta, S., Chander, A., Borad, P., Motiyani, I., Chakraborty, T., 2024. PerSEval: Assessing Personalization in Text Summarizers. https://doi.org/10.48550/arXiv.2407.00453

5) Dietterich, T.G., 2000. Ensemble Methods in Machine Learning, in: Multiple Classifier Systems, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–15. https://doi.org/10.1007/3-540-45014-9_1

6) Eyal, M., Baumel, T., Elhadad, M., 2019. Question Answering as an Automatic Evaluation Metric for News Article Summarization. https://doi.org/10.48550/arXiv.1906.00318

7) Goyal, T., Li, J.J., Durrett, G., 2022. SNaC: Coherence Error Detection for Narrative Summarization. https://doi.org/10.48550/arXiv.2205.09641

8) Guo, Z., Jin, R., Liu, C., Huang, Y., Shi, D., Supryadi, Yu, L., Liu, Y., Li, J., Xiong, B., Xiong, D., 2023. Evaluating Large Language Models: A Comprehensive Survey. https://doi.org/10.48550/arXiv.2310.19736

9) Hu, T., Zhou, X.-H., 2024. Unveiling LLM Evaluation Focused on Metrics: Challenges and Solutions. https://doi.org/10.48550/arXiv.2404.09135

10) Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Chen, D., Dai, W., Chan, H.S., Madotto, A., Fung, P., 2023. Survey of Hallucination in Natural Language Generation. ACM Comput. Surv. 55, 1–38. https://doi.org/10.1145/3571730

11) Kherwa, P., Arora, J., Sharma, T., Gupta, D., Juneja, S., Muhammad, G., Nauman, A., 2024. Contextual embedded text summarizer system: A hybrid approach. Expert Syst. 42, e13733. https://doi.org/10.1111/exsy.13733

12) Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q., 2015. From Word Embeddings To Document Distances. 06 July 2015 37.

13) Laban, P., Schnabel, T., Bennett, P.N., Hearst, M.A., 2022. SummaC : Re-Visiting NLI-based Models for Inconsistency Detection in Summarization. Trans. Assoc. Comput. Linguist. 10, 163–177. https://doi.org/10.1162/tacl_a_00453

14) Li, S., Su, W., Liu, J., 2021. LenC: A Redundancy-Aware Length Control Framework for Extractive Summarization, in: 2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI). Presented at the 2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), IEEE, Yibin, China, pp. 1–7. https://doi.org/10.1109/PRAI53619.2021.9550801

15) Lin, C.-Y., 2004. ROUGE: A Package for Automatic Evaluation of Summaries. Assoc. Comput. Linguist. Text Summarization Branches Out, 74–81.

16) Liu, Y., Fabbri, A.R., Zhao, Y., Liu, P., Joty, S., Wu, C.-S., Xiong, C., Radev, D., 2023. Towards Interpretable and Efficient Automatic Reference-Based Summarization Evaluation. https://doi.org/10.48550/arXiv.2303.03608

17) Mani, I., 2001. Recent developments in text summarization, in: Proceedings of the Tenth International Conference on Information and Knowledge Management. Presented at the CIKM01: International Conference on Information and Knowledge Management, ACM, Atlanta Georgia USA, pp. 529–531. https://doi.org/10.1145/502585.502677

18) Mastropaolo, A., Ciniselli, M., Penta, M.D., Bavota, G., 2023. Evaluating Code Summarization Techniques: A New Metric and an Empirical Characterization. https://doi.org/10.48550/arXiv.2312.15475

19) Nguyen, H., Chen, H., Pobbathi, L., Ding, J., 2024. A Comparative Study of Quality Evaluation Methods for Text Summarization. https://doi.org/10.48550/arXiv.2407.00747

20) pai, A., 2024. Build Large Language Models from Scratc.

21) Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2001. BLEU: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02. Presented at the the 40th Annual Meeting, Association for Computational Linguistics, Philadelphia, Pennsylvania, p. 311. https://doi.org/10.3115/1073083.1073135

22) Rahman, N., Borah, B., 2021. Redundancy Removal Method for Multi-Document Query-Based Text Summarization, in: 2021 International Symposium on Electrical, Electronics and Information Engineering. Presented at the ISEEIE 2021: 2021

International Symposium on Electrical, Electronics and Information Engineering, ACM, Seoul Republic of Korea, pp. 568–574. https://doi.org/10.1145/3459104.3459197

23) Ray, P.P., 2023. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Internet Things Cyber-Phys. Syst. 3, 121–154. https://doi.org/10.1016/j.iotcps.2023.04.003

24) Sarwar, R., Ahmad, B., Teh, P.S., Tuarob, S., Thaipisutikul, T., Zaman, F., Aljohani, N.R., Zhu, J., Hassan, S.-U., Nawaz, R., Ansari, A.R., Fayyaz, M.A.B., 2024. HybridEval: An Improved Novel Hybrid Metric for Evaluation of Text Summarization. J. Inform. Web Eng. 3, 233–255. https://doi.org/10.33093/jiwe.2024.3.3.15

25) Sellam, T., Das, D., Parikh, A.P., 2020. BLEURT: Learning Robust Metrics for Text Generation. https://doi.org/10.48550/arXiv.2004.04696

26) Yao, J.-Y., Ning, K.-P., Liu, Z.-H., Ning, M.-N., Liu, Y.-Y., Yuan, L., 2024. LLM Lies: Hallucinations are not Bugs, but Features as Adversarial Examples. https://doi.org/10.48550/arXiv.2310.01469

27) Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y., 2020a. BERTScore: Evaluating Text Generation with BERT. https://doi.org/10.48550/arXiv.1904.09675

28) Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y., 2020b. BERTScore: Evaluating Text Generation with BERT. https://doi.org/10.48550/arXiv.1904.09675

29) Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C.M., Eger, S., 2019. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. https://doi.org/10.48550/arXiv.1909.02622

APPENDICES

Source codes and results generated for the evaluation of SynSemScore, including statistical techniques applied for summarization assessment, correlation analysis with human evaluation scores

Loading dataset from hugging face

```
df = pd.read_parquet("hf://datasets/mteb/summeval/data/test-00000-of-00001-35901af5f6649399.parquet")
df.head()
```

	machine_summaries	human_summaries	relevance	coherence	fluency	consistency	text	
0	[donald sterling , nba team last year . sterli...	[V. Stiviano must pay back $2.6 million in gif...	[1.6666666666666667, 1.6666666666666667, 2.333...	[1.3333333333333333, 3.0, 1.0, 2.6666666666666...	[1.0, 4.666666666666667, 4.333333333333333, 4....	[1.0, 2.3333333333333335, 4.666666666666667, 5...	(CNN)Donald Sterling's racist remarks cost him...	404f859482d47c12786896
1	[north pacific gray whale has earned a spot in...	[The whale, Varvara, swam a round trip from Ru...	[2.3333333333333335, 4.666666666666667, 3.6666...	[1.3333333333333333, 4.666666666666667, 3.6666...	[1.0, 5.0, 4.666666666666667, 3.66666666666666...	[1.3333333333333333, 5.0, 5.0, 4.3333333333333...	(CNN)A North Pacific gray whale has earned a s...	4761dc6d8bdf56b9ada971
2	[russian fighter jet intercepted a u.s. reconn...	[The incident occurred on April 7 north of Pol...	[4.0, 4.0, 4.0, 3.3333333333333335, 3.33333333...	[3.3333333333333335, 4.333333333333333, 1.6666...	[3.6666666666666665, 4.333333333333333, 5.0, 4...	[5.0, 5.0, 4.666666666666667, 5.0, 5.0, 5.0, 5...	(CNN)After a Russian fighter jet intercepted a...	5139ccfabee55ddb83e793
3	[michael barnett captured the fire on intersta...	[Country band Lady Antebellum's bus caught fir...	[2.0, 3.0, 2.6666666666666665, 3.3333333333333...	[2.0, 3.0, 2.6666666666666665, 3.3333333333333...	[2.6666666666666665, 5.0, 5.0, 5.0, 5.0, 5.0, ...	[2.3333333333333335, 5.0, 5.0, 5.0, 5.0, 5.0, ...	(CNN)Lady Antebellum singer Hillary Scott's to...	88c2481234e763c9bbc68d
4	[deep reddish color caught seattle native tim ...	[Smoke from massive fires in Siberia created f...	[1.6666666666666667, 3.6666666666666665, 3.333...	[1.6666666666666667, 3.6666666666666665, 1.666...	[5.0, 5.0, 5.0, 5.0, 4.666666666666667, 5.0, 5...	[2.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, ...	(CNN)A fiery sunset greeted people in Washingt...	a02e362c5b8f049848ce71

Appendix 1 - Loading dataset

Expand human summeries

```
df_expanded_human_summaries = df.explode('human_summaries')
df_expanded_human_summaries = df_expanded_human_summaries.reset_index(drop=True)
df_expanded_human_summaries['human_summaries'].head()
```

```
0    V. Stiviano must pay back $2.6 million in gift...
1    A player for a team has been under scrunity fo...
2    Donald Sterling lost his NBA team amid racist ...
3    Donald Sterling cheated with a mistress, Stivi...
4    Donald Sterling's mistress will have to pay St...
Name: human_summaries, dtype: object
```

Appendix 2 - Expand human summaries

Select one human summery for each machine summries

```
df_sampled = df_expanded_human_summaries.iloc[6::11].reset_index(drop=True)
df_sampled.head()
```

	machine_summaries	human_summaries	relevance	coherence	fluency	consistency	text	
0	[donald sterling , nba team last year . sterli...	Donald Sterling lost a NBA franchise for his r...	[1.6666666666666667, 1.6666666666666667, 2.333...	[1.3333333333333333, 3.0, 1.0, 2.6666666666666...	[1.0, 4.666666666666667, 4.333333333333333, 4....	[1.0, 2.3333333333333335, 4.666666666666667, 5...	(CNN)Donald Sterling's racist remarks cost him...	404f859482d47c12786896
1	[north pacific gray whale has earned a spot in...	A north pacific gray whale named varvara just ...	[2.3333333333333335, 4.666666666666667, 3.6666...	[1.3333333333333333, 4.666666666666667, 3.6666...	[1.0, 5.0, 4.666666666666667, 3.66666666666666...	[1.3333333333333333, 5.0, 5.0, 4.3333333333333...	(CNN)A North Pacific gray whale has earned a s...	4761dc6d8bdf56b9ada971
2	[russian fighter jet intercepted a u.s. reconn...	An American plane was intercepted by a Russian...	[4.0, 4.0, 4.0, 3.3333333333333335, 3.33333333...	[3.3333333333333335, 4.333333333333333, 1.6666...	[3.6666666666666665, 4.333333333333333, 5.0, 4...	[5.0, 5.0, 4.666666666666667, 5.0, 5.0, 5.0, 5...	(CNN)After a Russian fighter jet intercepted a...	5139ccfabee55ddb83e793
3	[michael barnett captured the fire on intersta...	Hillary Scott of Lady Antebellum's tour bus ca...	[2.0, 3.0, 2.6666666666666665, 3.3333333333333...	[2.0, 3.0, 2.6666666666666665, 3.3333333333333...	[2.6666666666666665, 5.0, 5.0, 5.0, 5.0, 5.0, ...	[2.3333333333333335, 5.0, 5.0, 5.0, 5.0, 5.0, ...	(CNN)Lady Antebellum singer Hillary Scott's to...	88c2481234e763c9bbc68d
4	[deep reddish color caught seattle native tim ...	A fire in Southern Siberia was started by farm...	[1.6666666666666667, 3.6666666666666665, 3.333...	[1.6666666666666667, 3.6666666666666665, 1.666...	[5.0, 5.0, 5.0, 5.0, 4.666666666666667, 5.0, 5...	[2.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, 5.0, ...	(CNN)A fiery sunset greeted people in Washingt...	a02e362c5b8f049848ce71

Appendix 4 - Select one human summery

Expand machine summeries

```
df_sampled = df_sampled.explode('machine_summaries')
df_sampled = df_sampled.reset_index(drop=True)
df_sampled['machine_summaries'].head()
```

```
0    donald sterling , nba team last year . sterlin...
1    donald sterling accused stiviano of targeting ...
2    a los angeles judge has ordered v. stiviano to...
3    donald sterling 's wife sued stiviano of targe...
4    donald sterling 's racist remarks cost him an ...
Name: machine_summaries, dtype: object
```

Appendix 3 - Expand machine summaries

Expand (relevance,coherence,fluency,consistency)

```
df_machine_summaries = df["machine_summaries"].explode().reset_index(drop=True)
df_relevance = df["relevance"].explode().reset_index(drop=True)
df_coherence = df["coherence"].explode().reset_index(drop=True)
df_fluency = df["fluency"].explode().reset_index(drop=True)
df_consistency = df["consistency"].explode().reset_index(drop=True)
df_human_summaries = df_sampled["human_summaries"]

df_text = df["text"].repeat(16).reset_index(drop=True)  # Assuming each row corresponds to 16 summaries

df_final = pd.DataFrame({
    "text": df_text,
    "human_summaries":df_human_summaries,
    "machine_summaries": df_machine_summaries,
    "relevance": df_relevance,
    "coherence": df_coherence,
    "fluency": df_fluency,
    "consistency": df_consistency
})

df_final.head()
```

	text	human_summaries	machine_summaries	relevance	coherence	fluency	consistency
0	(CNN)Donald Sterling's racist remarks cost him...	Donald Sterling lost a NBA franchise for his r...	donald sterling , nba team last year . sterlin...	1.666667	1.333333	1.0	1.0
1	(CNN)Donald Sterling's racist remarks cost him...	Donald Sterling lost a NBA franchise for his r...	donald sterling accused stiviano of targeting ...	1.666667	3.0	4.666667	2.333333
2	(CNN)Donald Sterling's racist remarks cost him...	Donald Sterling lost a NBA franchise for his r...	a los angeles judge has ordered v. stiviano to...	2.333333	1.0	4.333333	4.666667
3	(CNN)Donald Sterling's racist remarks cost him...	Donald Sterling lost a NBA franchise for his r...	donald sterling 's wife sued stiviano of targe...	3.333333	2.666667	4.333333	5.0
4	(CNN)Donald Sterling's racist remarks cost him...	Donald Sterling lost a NBA franchise for his r...	donald sterling 's racist remarks cost him an ...	3.0	4.666667	5.0	5.0

Appendix 5 - Expand (relevance, coherence, fluency, consistency)

```
# Hyperparameter Tuning Using Grid Search and Cross-Validation
def hyperparameter_tuning(df, candidate_col, reference_col, human_scores, k_folds=5):
    # Define parameter grid
    alpha_values = np.arange(0.1, 1.0, 0.1)  # α ranges from 0.1 to 0.9
    lambda_values = [0.1, 0.3, 0.5, 0.7, 0.9]  # λ penalty values

    best_params = None
    best_score = -1  # Store the best correlation score

    kf = KFold(n_splits=k_folds, shuffle=True, random_state=42)

    for alpha in alpha_values:
        beta = 1 - alpha  # Ensure α + β = 1
        for lambda_penalty in lambda_values:
            correlations = []

            for train_index, test_index in kf.split(df):
                train_df, test_df = df.iloc[train_index], df.iloc[test_index]

                # Apply SynSemScore with selected α, β, and λ
                test_df = calculate_synsem_scores(test_df, candidate_col, reference_col, alpha, beta, lambda_penalty)

                # Compute correlation with human evaluation scores
                pearson_corr, _ = pearsonr(test_df["SynSemScore"], test_df["relevance"])  # Using relevance as target
                correlations.append(pearson_corr)

            avg_correlation = np.mean(correlations)  # Average correlation across folds

            # Store best performing parameters
            if avg_correlation > best_score:
                best_score = avg_correlation
                best_params = (alpha, beta, lambda_penalty)

    print(f"Best parameters: α={best_params[0]}, β={best_params[1]}, λ={best_params[2]} with correlation {best_score:.4f}")
    return best_params
```

Appendix 6 - Hyperparameter optimization

Normalized the scores

```
columns_to_normalize = ['relevance', 'coherence', 'fluency', 'consistency']
df_final[columns_to_normalize] = df_final[columns_to_normalize] / 5
df_final.head()
```

	text	human_summaries	machine_summaries	relevance	coherence	fluency	consistency
0	(CNN)Donald Sterling's racist remarks cost him...	Donald Sterling lost a NBA franchise for his r...	donald sterling , nba team last year . sterlin...	0.066667	0.053333	0.04	0.04
1	(CNN)Donald Sterling's racist remarks cost him...	Donald Sterling lost a NBA franchise for his r...	donald sterling accused stiviano of targeting ...	0.066667	0.12	0.186667	0.093333
2	(CNN)Donald Sterling's racist remarks cost him...	Donald Sterling lost a NBA franchise for his r...	a los angeles judge has ordered v. stiviano to...	0.093333	0.04	0.173333	0.186667
3	(CNN)Donald Sterling's racist remarks cost him...	Donald Sterling lost a NBA franchise for his r...	donald sterling 's wife sued stiviano of targe...	0.133333	0.106667	0.173333	0.2
4	(CNN)Donald Sterling's racist remarks cost him...	Donald Sterling lost a NBA franchise for his r...	donald sterling 's racist remarks cost him an ...	0.12	0.186667	0.2	0.2

Appendix 7 - Normalized the scores

Data preprocessing

```
try:
    stop_words = set(stopwords.words('english'))
except LookupError:
    nltk.download('stopwords')
    stop_words = set(stopwords.words('english'))

try:
    nltk.data.find('tokenizers/punkt')
except LookupError:
    nltk.download('punkt')

def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stop_words]
    cleaned_text = ' '.join(tokens)

    return cleaned_text
```

```
df_final['human_summaries'] = df_final['human_summaries'].apply(preprocess_text)
df_final['machine_summaries'] = df_final['machine_summaries'].apply(preprocess_text)
```

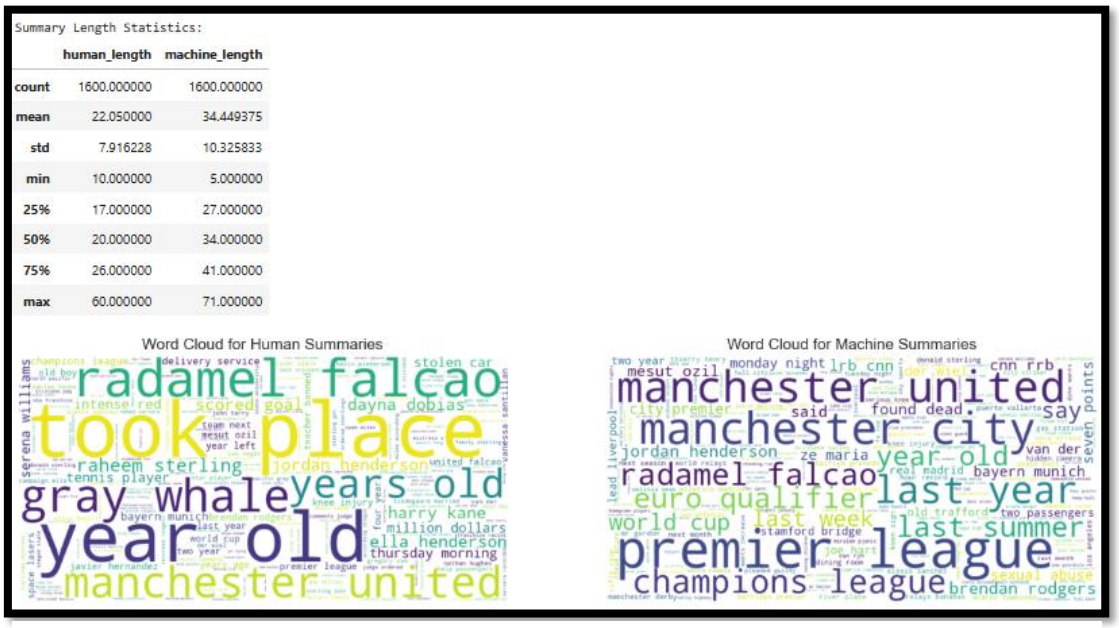
Appendix 8 - Data preprocessing

```python
numeric_cols = ['relevance', 'coherence', 'fluency', 'consistency']
Preprocessed_dataset[numeric_cols] = Preprocessed_dataset[numeric_cols].apply(pd.to_numeric, errors='coerce')

# Summary Length Analysis
Preprocessed_dataset['human_length'] = Preprocessed_dataset['human_summaries'].apply(lambda x: len(str(x).split()))
Preprocessed_dataset['machine_length'] = Preprocessed_dataset['machine_summaries'].apply(lambda x: len(str(x).split()))

# Display Summary Length Statistics
summary_length_stats = Preprocessed_dataset[['human_length', 'machine_length']].describe()
print("\nSummary Length Statistics:")
display(summary_length_stats)

# Generate word clouds
human_text = ' '.join(Preprocessed_dataset['human_summaries'].dropna())
machine_text = ' '.join(Preprocessed_dataset['machine_summaries'].dropna())

human_wordcloud = WordCloud(width=800, height=400, background_color='white').generate(human_text)
machine_wordcloud = WordCloud(width=800, height=400, background_color='white').generate(machine_text)

# Compute bigrams for human and machine summaries using whitespace-based tokenization
human_bigrams = list(zip(human_text.lower().split()[:-1], human_text.lower().split()[1:]))
machine_bigrams = list(zip(machine_text.lower().split()[:-1], machine_text.lower().split()[1:]))

human_bigram_counts = Counter(human_bigrams)
machine_bigram_counts = Counter(machine_bigrams)

# Convert to DataFrame for visualization
human_bigram_df = pd.DataFrame(human_bigram_counts.most_common(20), columns=['Bigram', 'Frequency'])
machine_bigram_df = pd.DataFrame(machine_bigram_counts.most_common(20), columns=['Bigram', 'Frequency'])

# Plot word clouds
fig, axes = plt.subplots(1, 2, figsize=(15, 7))
axes[0].imshow(human_wordcloud, interpolation='bilinear')
axes[0].set_title('Word Cloud for Human Summaries')
axes[0].axis('off')

axes[1].imshow(machine_wordcloud, interpolation='bilinear')
axes[1].set_title('Word Cloud for Machine Summaries')
axes[1].axis('off')

plt.show()

# Display bigram frequency bar charts (Human Summaries)
plt.figure(figsize=(12, 6))
sns.barplot(
    y=[' '.join(b) for b in human_bigram_df['Bigram']],
    x=human_bigram_df['Frequency'],
    hue=[' '.join(b) for b in human_bigram_df['Bigram']],
    palette='Blues_r',
    legend=False
)
plt.xlabel("Frequency")
plt.ylabel("Bigrams")
plt.title("Top 20 Bigrams in Human Summaries")
plt.show()

# Display bigram frequency bar charts (Machine Summaries)
plt.figure(figsize=(12, 6))
sns.barplot(
    y=[' '.join(b) for b in machine_bigram_df['Bigram']],
    x=machine_bigram_df['Frequency'],
    hue=[' '.join(b) for b in machine_bigram_df['Bigram']],
    palette='Reds_r',
    legend=False
)
plt.xlabel("Frequency")
plt.ylabel("Bigrams")
plt.title("Top 20 Bigrams in Machine Summaries")
plt.show()
```

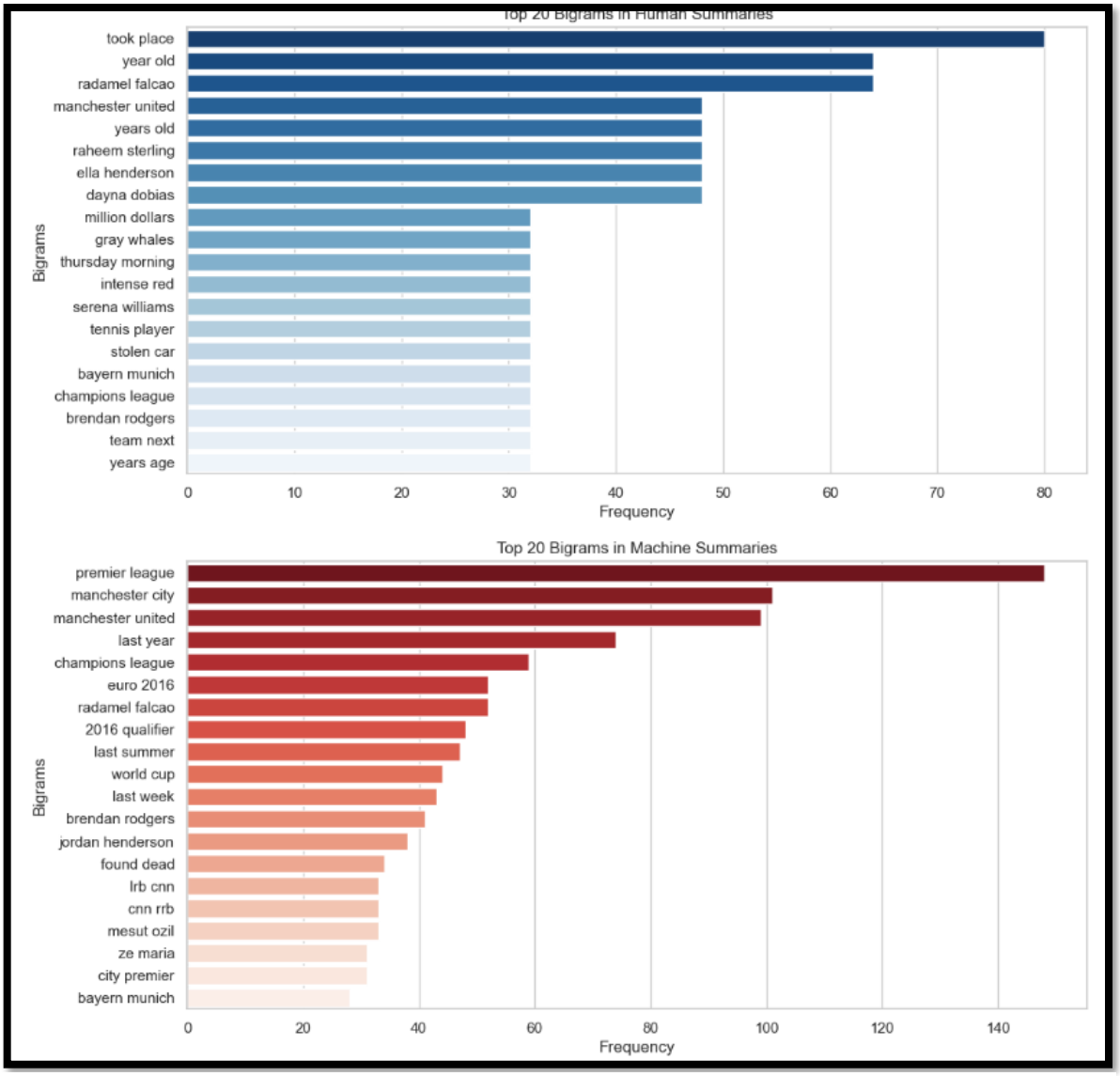
Appendix 9 - Dataset analysis

Summary Length Statistics:

	human_length	machine_length
count	1600.000000	1600.000000
mean	22.050000	34.449375
std	7.916228	10.325833
min	10.000000	5.000000
25%	17.000000	27.000000
50%	20.000000	34.000000
75%	26.000000	41.000000
max	60.000000	71.000000

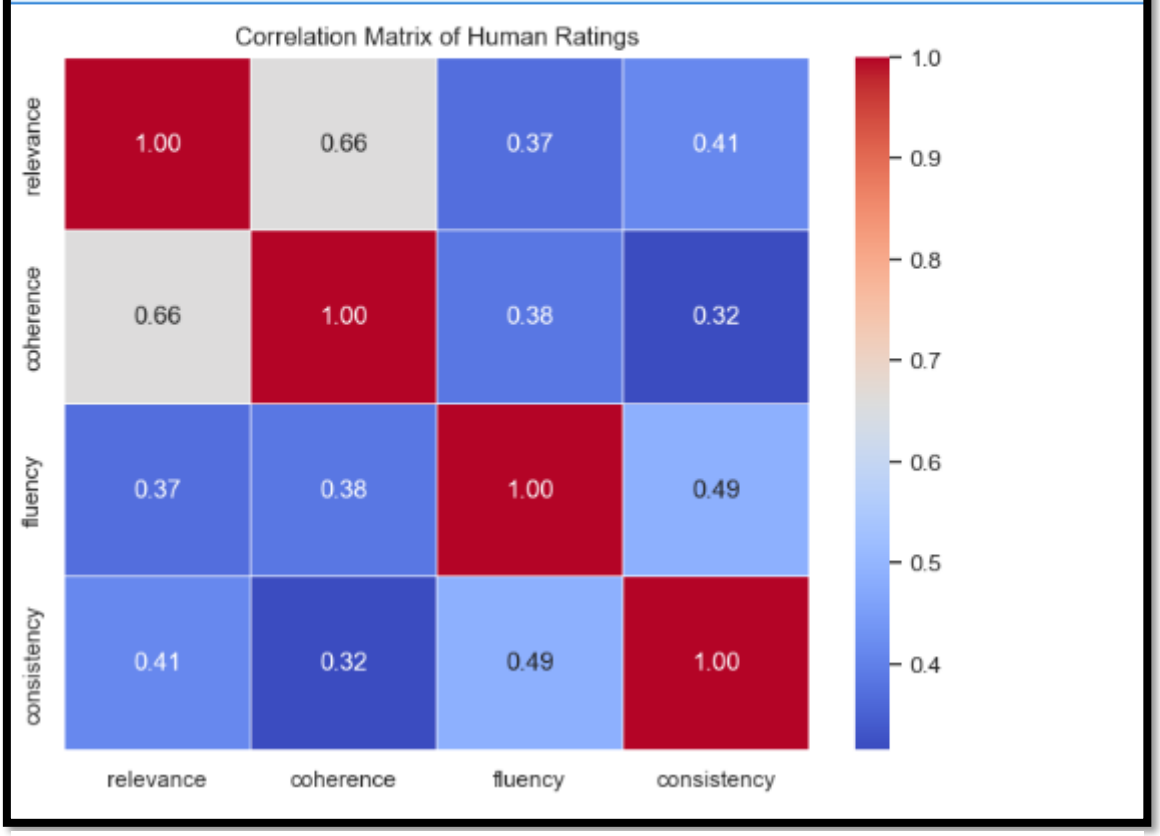


Appendix 11 - Word Clouds



Appendix 10 - Bigram Statistics

```
correlation_matrix = Preprocessed_dataset[['relevance', 'coherence', 'fluency', 'consistency']].corr()

# Plot heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title("Correlation Matrix of Human Ratings")
plt.show()
```



Appendix 12 - correlation matrix of human rating

```
def calculate_bleu(reference, candidate):
    smoothing_function = SmoothingFunction().method1
    return sentence_bleu([reference], candidate, smoothing_function=smoothing_function)

def calculate_rouge(reference, candidate):
    scorer = rouge_scorer.RougeScorer(['rouge1', 'rouge2', 'rougeL'], use_stemmer=True)
    scores = scorer.score(" ".join(reference), " ".join(candidate))
    return scores['rouge1'].fmeasure, scores['rouge2'].fmeasure, scores['rougeL'].fmeasure

def calculate_meteor(reference, candidate):
    return meteor_score([reference], candidate)
```

Appendix 13 - Traditional metric calculation functions

```python
def performance_metrics(candidate, reference, n=1):
    candidate_ngrams = [tuple(candidate[i:i + n]) for i in range(len(candidate) - n + 1)]
    reference_ngrams = [tuple(reference[i:i + n]) for i in range(len(reference) - n + 1)]

    candidate_counter = Counter(candidate_ngrams)
    reference_counter = Counter(reference_ngrams)

    overlap = sum((candidate_counter & reference_counter).values())
    total_possible = sum(reference_counter.values())

    precision = overlap / sum(candidate_counter.values()) if candidate_counter else 0
    recall = overlap / total_possible if total_possible else 0
    f_measure = (2 * precision * recall) / (precision + recall) if precision + recall > 0 else 0

    return precision, recall, f_measure
```

Appendix 14 - syntactic function in SynSemScore

```python
def semantic_similarity(word1, word2):
    synsets1 = wordnet.synsets(word1)
    synsets2 = wordnet.synsets(word2)

    max_similarity = 0
    for syn1 in synsets1:
        for syn2 in synsets2:
            similarity = syn1.wup_similarity(syn2)
            if similarity and similarity > max_similarity:
                max_similarity = similarity

    return max_similarity

def semantic_score(candidate, reference):
    candidate_tokens = word_tokenize(candidate)
    reference_tokens = word_tokenize(reference)

    matches = 0
    for ref_word in reference_tokens:
        for cand_word in candidate_tokens:
            if ref_word == cand_word:
                matches += 1
                break
            elif wordnet.synsets(ref_word) and wordnet.synsets(cand_word):
                if semantic_similarity(ref_word, cand_word) > 0.7:
                    matches += 1
                    break

    return matches / len(reference_tokens) if reference_tokens else 0
```

Appendix 15 - semantic function in SynSemScore

```python
def redundancy_penalty(candidate):
    candidate_tokens = word_tokenize(candidate)
    bigrams = [tuple(candidate_tokens[i:i + 2]) for i in range(len(candidate_tokens) - 1)]
    bigram_counter = Counter(bigrams)

    redundant = sum(count - 1 for count in bigram_counter.values() if count > 1)
    total_bigrams = len(bigrams)

    return 1 - (redundant / total_bigrams) if total_bigrams else 1
```

Appendix 18 - Redundancy penalty function in SynSemScore

```python
def synsem_score(candidate, reference, alpha=0.6, beta=0.4):
    _, _, rouge_l = performance_metrics(candidate.split(), reference.split(), n=1)
    semantic = semantic_score(candidate, reference)
    raw_score = alpha * rouge_l + beta * semantic
    penalty = redundancy_penalty(candidate)
    return raw_score * penalty
```

Appendix 17 - Final SynSemScore calculation function

```python
def calculate_metrics(row, candidate_col, reference_col):
    reference = word_tokenize(row[reference_col])
    candidate = word_tokenize(row[candidate_col])

    bleu = calculate_bleu(reference, candidate)
    rouge_1, rouge_2, rouge_l = calculate_rouge(reference, candidate)
    meteor = calculate_meteor(reference, candidate)
    synsem = synsem_score(" ".join(candidate), " ".join(reference))

    return pd.Series({'BLEU': bleu, 'ROUGE_1': rouge_1, 'ROUGE_2': rouge_2, 'ROUGE_L': rouge_l, 'METEOR': meteor, 'SynSemScore': synsem})
```

Appendix 16 - Metric calculation function

```python
# Perform Cross-Validation
def cross_validate_metrics(df, candidate_col, reference_col, human_scores, n_splits=5):
    kf = KFold(n_splits=n_splits, shuffle=True, random_state=42)
    cv_results = []

    for fold_index, (train_index, test_index) in enumerate(kf.split(df)):
        train_df = df.iloc[train_index]
        test_df = df.iloc[test_index]

        train_metrics = train_df.apply(calculate_metrics, axis=1, candidate_col=candidate_col, reference_col=reference_col)
        test_metrics = test_df.apply(calculate_metrics, axis=1, candidate_col=candidate_col, reference_col=reference_col)

        train_df = pd.concat([train_df, train_metrics], axis=1)
        test_df = pd.concat([test_df, test_metrics], axis=1)

        train_correlations = {}
        test_correlations = {}

        for metric in ['BLEU', 'ROUGE_1', 'ROUGE_2', 'ROUGE_L', 'METEOR', 'SynSemScore']:
            train_spearman = []
            test_spearman = []
            train_pearson = []
            test_pearson = []

            for score in human_scores:
                train_spearman_corr, _ = spearmanr(train_df[metric], train_df[score])
                test_spearman_corr, _ = spearmanr(test_df[metric], test_df[score])
                train_pearson_corr, _ = pearsonr(train_df[metric], train_df[score])
                test_pearson_corr, _ = pearsonr(test_df[metric], test_df[score])

                train_spearman.append(train_spearman_corr)
                test_spearman.append(test_spearman_corr)
                train_pearson.append(train_pearson_corr)
                test_pearson.append(test_pearson_corr)
            # Store train correlations in a consistent nested dictionary structure
            train_correlations[metric] = {
                score: {
                    'spearman': train_spearman[i],
                    'pearson': train_pearson[i]
                } for i, score in enumerate(human_scores)
            }

            # Store test correlations in the same structure
            test_correlations[metric] = {
                score: {
                    'spearman': test_spearman[i],
                    'pearson': test_pearson[i]
                } for i, score in enumerate(human_scores)
            }
```

Appendix 19 - Perform cross-validation

```
    cv_results.append({'fold': fold_index + 1, 'train_df': train_df, 'test_df': test_df, 'train_correlations': train_

# Compute the average correlations across all folds (Train Data)
avg_train_correlations = {
    metric: {
        score: {
            'spearman': np.mean([fold['train_correlations'][metric][score]['spearman'] for fold in cv_results]),
            'pearson': np.mean([fold['train_correlations'][metric][score]['pearson'] for fold in cv_results])
        } for score in human_scores
    } for metric in ['BLEU', 'ROUGE_1', 'ROUGE_2', 'ROUGE_L', 'METEOR', 'SynSemScore']
}

# Compute the average correlations across all folds (Test Data)
avg_test_correlations = {
    metric: {
        score: {
            'spearman': np.mean([fold['test_correlations'][metric][score]['spearman'] for fold in cv_results]),
            'pearson': np.mean([fold['test_correlations'][metric][score]['pearson'] for fold in cv_results])
        } for score in human_scores
    } for metric in ['BLEU', 'ROUGE_1', 'ROUGE_2', 'ROUGE_L', 'METEOR', 'SynSemScore']
}


avg_metric_correlations_train = {
    metric: {
        score: {
            'spearman': avg_train_correlations[metric][score]['spearman'],
            'pearson': avg_train_correlations[metric][score]['pearson']
        } for score in human_scores
    } for metric in avg_train_correlations
}

avg_metric_correlations_test = {
    metric: {
        score: {
            'spearman': avg_test_correlations[metric][score]['spearman'],
            'pearson': avg_test_correlations[metric][score]['pearson']
        } for score in human_scores
    } for metric in avg_test_correlations
}
```

Appendix 20 - Finding average correlation

```
    # Identify best metric based on test average correlations
    best_spearman_metric = max(
    avg_metric_correlations_test,
    key=lambda m: np.mean([
        avg_metric_correlations_test[m][score]['spearman'] for score in human_scores
    ])
)

    best_pearson_metric = max(
    avg_metric_correlations_test,
    key=lambda m: np.mean([
        avg_metric_correlations_test[m][score]['pearson'] for score in human_scores
    ])
)

    return cv_results, avg_metric_correlations_train, avg_metric_correlations_test, best_spearman_metric, best_pearson_metric

cv_results, avg_metric_correlations_train, avg_metric_correlations_test, best_spearman_metric, best_pearson_metric = cross_validate_metrics(
    Preprocessed_dataset, 'machine_summaries', 'human_summaries', ['relevance', 'coherence', 'fluency', 'consistency']
)

# Display train/test metrics and correlations for each fold
for fold_result in cv_results:
    print(f"Fold {fold_result['fold']}:")

    print("\nTrain Metrics:")
    display(fold_result['train_df'][['machine_summaries', 'human_summaries', 'BLEU', 'ROUGE_1', 'ROUGE_2', 'ROUGE_L', 'METEOR', 'SynSemScore']])

    print("\nTest Metrics:")
    display(fold_result['test_df'][['machine_summaries', 'human_summaries', 'BLEU', 'ROUGE_1', 'ROUGE_2', 'ROUGE_L', 'METEOR', 'SynSemScore']])

    print("\nTrain Correlations:")
    display(pd.DataFrame(fold_result['train_correlations']))

    print("\nTest Correlations:")
    display(pd.DataFrame(fold_result['test_correlations']))

    print("\n" + "-"*50 + "\n")
```

Appendix 22 - Display train / test metrics and correlations for each fold

```
Fold 1:

Train Metrics:
```

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
0	donald sterling nba team last year sterling wi...	donald sterling lost nba franchise racist comm...	0.041716	0.433333	0.137931	0.300000	0.266393	0.554589
1	donald sterling accused stiviano targeting ext...	donald sterling lost nba franchise racist comm...	0.016605	0.226415	0.039216	0.188679	0.147679	0.364471
2	los angeles judge ordered v stiviano pay back ...	donald sterling lost nba franchise racist comm...	0.052569	0.535714	0.185185	0.321429	0.526042	0.665188
3	donald sterling wife sued stiviano targeting e...	donald sterling lost nba franchise racist comm...	0.015021	0.214286	0.037037	0.178571	0.125000	0.361316
4	donald sterling racist remarks cost nba team l...	donald sterling lost nba franchise racist comm...	0.074367	0.565217	0.227273	0.434783	0.461069	0.704348
...	...	...	...	...	...	...	...	...
1594	woman named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.016894	0.315789	0.109091	0.245614	0.276414	0.418421
1596	gemma 23 two children five two different fathe...	gemma woman currently risk losing children hor...	0.018012	0.264151	0.117647	0.264151	0.257653	0.402516
1597	gemma named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.022070	0.340426	0.133333	0.297872	0.264121	0.303191
1598	woman named gemma two children five two father...	gemma woman currently risk losing children hor...	0.019355	0.339623	0.117647	0.264151	0.280927	0.414465
1599	woman two children five two different fathers ...	gemma woman currently risk losing children hor...	0.010925	0.274510	0.081633	0.235294	0.261523	0.407843

1280 rows × 8 columns

Appendix 21 - Fold 1 train metric score results

Test Metrics:

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
15	donald sterling remarks cost nba team last yea...	donald sterling lost nba franchise racist comm...	0.047690	0.482759	0.178571	0.413793	0.418090	0.598651
23	varvara north pacific gray whale swam nearly 1...	north pacific gray whale named varvara recorde...	0.041010	0.315789	0.145455	0.280702	0.178772	0.383055
29	whale named varvara swam nearly 14000 miles 22...	north pacific gray whale named varvara recorde...	0.026580	0.238806	0.092308	0.208955	0.188354	0.351365
30	north pacific gray whale swam nearly 14000 mil...	north pacific gray whale named varvara recorde...	0.084264	0.424242	0.187500	0.363636	0.305640	0.463415
32	russian fighter jet intercepted us reconnaissa...	american plane intercepted russian su27 flanke...	0.082172	0.346154	0.200000	0.192308	0.527461	0.546154
...	...	...	...	...	...	...	...	...
1578	manchester city keen sign anderlecht teenager ...	manchester city interested signing young footb...	0.018041	0.204082	0.085106	0.204082	0.231088	0.309509
1581	manchester city keen sign anderlecht teenager ...	manchester city interested signing young footb...	0.018041	0.204082	0.085106	0.204082	0.264824	0.364626
1589	gemma named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.017449	0.327273	0.113208	0.254545	0.278652	0.386061
1591	gemma 23 two children five two different fathe...	gemma woman currently risk losing children hor...	0.018720	0.264151	0.039216	0.226415	0.183673	0.425157
1595	23yearold motheroftwo risk banned seeing child...	gemma woman currently risk losing children hor...	0.013872	0.369231	0.095238	0.246154	0.331820	0.422000

Appendix 24 - Fold 1 test metric score results

Train Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.03853773512777055, 'pearson': 0...	{'spearman': 0.05040063721513588, 'pearson': 0...	{'spearman': 0.034558594432213395, 'pearson': ...	{'spearman': 0.028945605291490744, 'pearson': ...	{'spearman': 0.08330608188237443, 'pearson': 0...	{'spearman': 0.12263110833965059, 'pearson': 0...
coherence	{'spearman': -0.02988749520290 5703, 'pearson':...	{'spearman': -0.012783274485856965, 'pearson':...	{'spearman': -0.03029625819212809, 'pearson': ...	{'spearman': -0.007080701995647717, 'pearson':...	{'spearman': -0.026506360426480184, 'pearson':...	{'spearman': 0.0547460104259 2104, 'pearson': 0...
fluency	{'spearman': 0.01608763631262268, 'pearson': 0...	{'spearman': 0.007909698406455775, 'pearson': ...	{'spearman': 0.007378923040858917, 'pearson': ...	{'spearman': 0.004829436281367886, 'pearson': ...	{'spearman': 0.005497808566145315, 'pearson': ...	{'spearman': 0.015806389401318933, 'pearson': ...
consistency	{'spearman': 0.06762551404416962, 'pearson': 0...	{'spearman': 0.06491841885004171, 'pearson': 0...	{'spearman': 0.07922310829578731, 'pearson': 0...	{'spearman': 0.062482411419777596, 'pearson': ...	{'spearman': 0.09557215924745956, 'pearson': 0...	{'spearman': 0.09706277075505612, 'pearson': 0...

Test Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.08029197731755748, 'pearson': 0...	{'spearman': 0.09958924022390617, 'pearson': 0...	{'spearman': 0.10205310969162316, 'pearson': 0...	{'spearman': 0.04148784939428024, 'pearson': 0...	{'spearman': 0.15865717187244135, 'pearson': 0...	{'spearman': 0.14517273867795494, 'pearson': 0...
coherence	{'spearman': 0.04107836163929036, 'pearson': 0...	{'spearman': 0.02848111006497482, 'pearson': 0...	{'spearman': 0.042972285302056906, 'pearson': ...	{'spearman': 0.00818434392317839, 'pearson': 0...	{'spearman': 0.09099972241332682, 'pearson': 0...	{'spearman': 0.09760959750083396, 'pearson': 0...
fluency	{'spearman': -0.008300677353886303, 'pearson':...	{'spearman': -0.04644855415223115, 'pearson': ...	{'spearman': 0.024800588168189402, 'pearson': ...	{'spearman': -0.026931364413 11644, 'pearson': ...	{'spearman': -0.008228690018687588, 'pearson':...	{'spearman': -0.013984605372403584, 'pearson':...
consistency	{'spearman': 0.01665657455591671, 'pearson': 0...	{'spearman': 0.050887370401373755, 'pearson': ...	{'spearman': 0.06001893549807531, 'pearson': 0...	{'spearman': -0.024984673747336 81, 'pearson': ...	{'spearman': 0.1182819537854364, 'pearson': 0....	{'spearman': 0.10143124539173369, 'pearson': 0...

Appendix 23 - Train and test correlation for fold 1

Fold 2:

Train Metrics:

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
0	donald sterling nba team last year sterling wi...	donald sterling lost nba franchise racist comm...	0.041716	0.433333	0.137931	0.300000	0.266393	0.554589
1	donald sterling accused stiviano targeting ext...	donald sterling lost nba franchise racist comm...	0.016605	0.226415	0.039216	0.188679	0.147679	0.364471
2	los angeles judge ordered v stiviano pay back ...	donald sterling lost nba franchise racist comm...	0.052569	0.535714	0.185185	0.321429	0.526042	0.665188
3	donald sterling wife sued stiviano targeting e...	donald sterling lost nba franchise racist comm...	0.015021	0.214286	0.037037	0.178571	0.125000	0.361316
4	donald sterling racist remarks cost nba team l...	donald sterling lost nba franchise racist comm...	0.074367	0.565217	0.227273	0.434783	0.461069	0.704348
...	...	...	...	...	...	...	...	...
1595	23yearold motheroftwo risk banned seeing child...	gemma woman currently risk losing children hor...	0.013872	0.369231	0.095238	0.246154	0.331820	0.422000
1596	gemma 23 two children five two different fathe...	gemma woman currently risk losing children hor...	0.018012	0.264151	0.117647	0.264151	0.257653	0.402516
1597	gemma named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.022070	0.340426	0.133333	0.297872	0.264121	0.303191
1598	woman named gemma two children five two father...	gemma woman currently risk losing children hor...	0.019355	0.339623	0.117647	0.264151	0.280927	0.414465
1599	woman two children five two different fathers ...	gemma woman currently risk losing children hor...	0.010925	0.274510	0.081633	0.235294	0.261523	0.407843

Appendix 26 - Fold 2 train metric score results

Test Metrics:

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
10	los angeles judge ordered v stiviano pay back ...	donald sterling lost nba franchise racist comm...	0.037727	0.393939	0.156250	0.272727	0.315503	0.531225
12	judge orders v stiviano pay back 26 million gi...	donald sterling lost nba franchise racist comm...	0.044924	0.428571	0.185185	0.285714	0.355324	0.544720
18	lrb cnn rrb north pacific gray whale earned sp...	north pacific gray whale named varvara recorde...	0.086274	0.442308	0.176471	0.365385	0.404808	0.585553
31	north pacific gray whale earned spot record bo...	north pacific gray whale named varvara recorde...	0.111975	0.358974	0.210526	0.307692	0.329557	0.467542
41	us rc135u flying baltic sea intercepted russia...	american plane intercepted russian su27 flanke...	0.102386	0.413793	0.296296	0.206897	0.384294	0.525199
...	...	...	...	...	...	...	...	...
1583	manchester city keen sign anderlecht teenager ...	manchester city interested signing young footb...	0.019966	0.285714	0.085106	0.285714	0.248493	0.413605
1586	woman named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.013872	0.369231	0.095238	0.246154	0.331820	0.422000
1587	woman named gemma two children five two father...	gemma woman currently risk losing children hor...	0.015884	0.305085	0.105263	0.237288	0.296414	0.429379
1590	23 year old mother two risk banned seeing chil...	gemma woman currently risk losing children hor...	0.012312	0.342857	0.088235	0.228571	0.325488	0.392889
1593	woman named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.020087	0.346154	0.120000	0.269231	0.282079	0.417949

Appendix 25 - Fold 2 test metric score results

Train Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.051532469762516284, 'pearson': ...	{'spearman': 0.0685994227780073, 'pearson': 0....	{'spearman': 0.062017446868367986, 'pearson': ...	{'spearman': 0.0387894008117262, 'pearson': 0....	{'spearman': 0.11345526305420006, 'pearson': 0...	{'spearman': 0.1436508921895781, 'pearson': 0....
coherence	{'spearman': -0.007652132224655172, 'pearson':...	{'spearman': 0.003294983990150284, 'pearson': ...	{'spearman': -0.004119937215869351, 'pearson':...	{'spearman': 0.006166954231576541, 'pearson': ...	{'spearman': 0.003969007342927277, 'pearson': ...	{'spearman': 0.0699549180213178 7, 'pearson': 0...
fluency	{'spearman': 0.009996957937455331, 'pearson': ...	{'spearman': -0.003552331871314987, 'pearson':...	{'spearman': 0.01319687625384266, 'pearson': 0...	{'spearman': -0.0035967858216329504, 'pearson'...	{'spearman': -0.004815441805113032, 'pearson':...	{'spearman': 0.0035049023952304034, 'pearson':...
consistency	{'spearman': 0.056450399385827285, 'pearson': ...	{'spearman': 0.053053009225379 26, 'pearson': 0...	{'spearman': 0.07589005901758418, 'pearson': 0...	{'spearman': 0.027387537906843312, 'pearson': ...	{'spearman': 0.10396592167910704, 'pearson': 0...	{'spearman': 0.0958506828259235, 'pearson': 0...

Test Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.02945357000037013, 'pearson': 0...	{'spearman': 0.018458460063953946, 'pearson': ...	{'spearman': -0.009964570732756153, 'pearson':...	{'spearman': -0.001798301204192807, 'pearson':...	{'spearman': 0.03493660690287189, 'pearson': 0...	{'spearman': 0.0633604839456586 2, 'pearson': 0...
coherence	{'spearman': -0.05090522702466195, 'pearson': ...	{'spearman': -0.04459635580708776, 'pearson': ...	{'spearman': -0.06698656594626295, 'pearson': ...	{'spearman': -0.047632433871935165, 'pearson':...	{'spearman': -0.03930948324492456, 'pearson': ...	{'spearman': 0.035552099443443605, 'pearson': ...
fluency	{'spearman': -0.0002784034961650 6827, 'pearson...	{'spearman': -0.013955275742588641, 'pearson':...	{'spearman': -0.007727196562423074, 'pearson':...	{'spearman': 0.010168598410670068, 'pearson': ...	{'spearman': 0.02058018061928116, 'pearson': 0...	{'spearman': 0.0302941850289739 7, 'pearson': 0...
consistency	{'spearman': 0.053720325062062334, 'pearson': ...	{'spearman': 0.091516417323291 97, 'pearson': 0...	{'spearman': 0.0705131602955637, 'pearson': 0....	{'spearman': 0.12116070354645594, 'pearson': 0...	{'spearman': 0.07024392020675616, 'pearson': 0...	{'spearman': 0.1053039332690332 8, 'pearson': 0...

Appendix 27 - Train and test correlation for fold 2

Fold 3:

Train Metrics:

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
1	donald sterling accused stiviano targeting ext...	donald sterling lost nba franchise racist comm...	0.016605	0.226415	0.039216	0.188679	0.147679	0.364471
4	donald sterling racist remarks cost nba team l...	donald sterling lost nba franchise racist comm...	0.074367	0.565217	0.227273	0.434783	0.461069	0.704348
7	judge orders v stiviano pay back 26 million gi...	donald sterling lost nba franchise racist comm...	0.048369	0.490566	0.196078	0.339623	0.382368	0.602133
8	donald sterlings racist remarks cost nba team ...	donald sterling lost nba franchise racist comm...	0.011230	0.270270	0.057143	0.162162	0.192936	0.388249
10	los angeles judge ordered v stiviano pay back ...	donald sterling lost nba franchise racist comm...	0.037727	0.393939	0.156250	0.272727	0.315503	0.531225
...	...	...	...	...	...	...	...	...
1595	23yearold motheroftwo risk banned seeing child...	gemma woman currently risk losing children hor...	0.013872	0.369231	0.095238	0.246154	0.331820	0.422000
1596	gemma 23 two children five two different fathe...	gemma woman currently risk losing children hor...	0.018012	0.264151	0.117647	0.264151	0.257653	0.402516
1597	gemma named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.022070	0.340426	0.133333	0.297872	0.264121	0.303191
1598	woman named gemma two children five two father...	gemma woman currently risk losing children hor...	0.019355	0.339623	0.117647	0.264151	0.280927	0.414465
1599	woman two children five two different fathers ...	gemma woman currently risk losing children hor...	0.010925	0.274510	0.081633	0.235294	0.261523	0.407843

Appendix 28 - Fold 3 train metric score results

Test Metrics:

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
0	donald sterling nba team last year sterling wi...	donald sterling lost nba franchise racist comm...	0.041716	0.433333	0.137931	0.300000	0.266393	0.554589
2	los angeles judge ordered v stiviano pay back ...	donald sterling lost nba franchise racist comm...	0.052569	0.535714	0.185185	0.321429	0.526042	0.665188
3	donald sterling wife sued stiviano targeting e...	donald sterling lost nba franchise racist comm...	0.015021	0.214286	0.037037	0.178571	0.125000	0.361316
5	rochelle shelly sterling accused stiviano targ...	donald sterling lost nba franchise racist comm...	0.018562	0.240000	0.041667	0.160000	0.149573	0.382943
6	cnn donald sterling racist remarks cost nba te...	donald sterling lost nba franchise racist comm...	0.043002	0.507937	0.163934	0.380952	0.567222	0.668323
...	...	...	...	...	...	...	...	...
1576	anderlecht keen sign anderlecht teenager evang...	manchester city interested signing young footb...	0.011232	0.113208	0.039216	0.113208	0.160216	0.311950
1577	evangelos patoulidis regarded one best talents...	manchester city interested signing young footb...	0.017515	0.097561	0.051282	0.097561	0.138138	0.302981
1582	evangelos patoulidis linked arsenal barcelona ...	manchester city interested signing young footb...	0.025281	0.222222	0.058824	0.166667	0.175347	0.344444
1588	woman named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.016339	0.350877	0.109091	0.280702	0.298795	0.364035
1594	woman named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.016894	0.315789	0.109091	0.245614	0.276414	0.418421

Appendix 30 - Fold 3 test metric score results

Train Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.0508400794163380 6, 'pearson': 0...	{'spearman': 0.0744036388632185, 'pearson': 0...	{'spearman': 0.0597663155734067 5, 'pearson': 0...	{'spearman': 0.0357588653318904 74, 'pearson': ...	{'spearman': 0.1061442392926577 6, 'pearson': 0...	{'spearman': 0.1379591914002495 6, 'pearson': 0...
coherence	{'spearman': 0.004783603635216676, 'pearson': ...	{'spearman': 0.019780626404388903, 'pearson': ...	{'spearman': 0.013851631365586934, 'pearson': ...	{'spearman': 0.021647701279951 97, 'pearson': 0...	{'spearman': 0.026912244102039917, 'pearson': ...	{'spearman': 0.096452442407018 9, 'pearson': 0....
fluency	{'spearman': 0.020323194601080334, 'pearson': ...	{'spearman': 0.011765282428694339, 'pearson': ...	{'spearman': 0.028262928856223737, 'pearson': ...	{'spearman': 0.016187032125890 33, 'pearson': 0...	{'spearman': 0.020656739121263302, 'pearson': ...	{'spearman': 0.031823089741630 77, 'pearson': 0...
consistency	{'spearman': 0.057306807251309176, 'pearson': ...	{'spearman': 0.0669269529990294 3, 'pearson': 0...	{'spearman': 0.080374669751036 58, 'pearson': 0...	{'spearman': 0.045635242040591 35, 'pearson': 0...	{'spearman': 0.105247119996617 01, 'pearson': 0...	{'spearman': 0.103768635690442 38, 'pearson': 0...

Test Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.036549836288260 05, 'pearson': -...	{'spearman': 0.0027009193661987263, 'pearson':...	{'spearman': 0.005698609895369 84, 'pearson': ...	{'spearman': 0.019603112279517 915, 'pearson': ...	{'spearman': 0.068005577855115 21, 'pearson': 0...	{'spearman': 0.083169537250307 67, 'pearson': 0...
coherence	{'spearman': -0.09501485003630951, 'pearson': ...	{'spearman': -0.10522211184242451, 'pearson': ...	{'spearman': -0.13391699139051472, 'pearson': ...	{'spearman': -0.1049067150302231, 'pearson': -...	{'spearman': -0.125239203923910 64, 'pearson': ...	{'spearman': -0.073678219761347 78, 'pearson': ...
fluency	{'spearman': -0.02417856408470136, 'pearson': ...	{'spearman': -0.0620206211944608, 'pearson': -...	{'spearman': -0.054409251304446835, 'pearson':...	{'spearman': -0.064869867739587 13, 'pearson': ...	{'spearman': -0.068225422383148 48, 'pearson': ...	{'spearman': -0.084371794161865 85, 'pearson': ...
consistency	{'spearman': 0.056322019106521 35, 'pearson': -...	{'spearman': 0.038127845947788 94, 'pearson': 0...	{'spearman': 0.051973087333384 87, 'pearson': 0...	{'spearman': 0.043778318711899 77, 'pearson': 0...	{'spearman': 0.081357491313068 35, 'pearson': 0...	{'spearman': 0.068597076831983 75, 'pearson': 0...

Appendix 29 - Train and test correlation for fold 3

Train Metrics:

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
0	donald sterling nba team last year sterling wi...	donald sterling lost nba franchise racist comm...	0.041716	0.433333	0.137931	0.300000	0.266393	0.554589
1	donald sterling accused stiviano targeting ext...	donald sterling lost nba franchise racist comm...	0.016605	0.226415	0.039216	0.188679	0.147679	0.364471
2	los angeles judge ordered v stiviano pay back ...	donald sterling lost nba franchise racist comm...	0.052569	0.535714	0.185185	0.321429	0.526042	0.665188
3	donald sterling wife sued stiviano targeting e...	donald sterling lost nba franchise racist comm...	0.015021	0.214286	0.037037	0.178571	0.125000	0.361316
5	rochelle shelly sterling accused stiviano targ...	donald sterling lost nba franchise racist comm...	0.018562	0.240000	0.041667	0.160000	0.149573	0.382943
...	...	...	...	...	...	...	...	...
1591	gemma 23 two children five two different fathe...	gemma woman currently risk losing children hor...	0.018720	0.264151	0.039216	0.226415	0.183673	0.425157
1593	woman named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.020087	0.346154	0.120000	0.269231	0.282079	0.417949
1594	woman named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.016894	0.315789	0.109091	0.245614	0.276414	0.418421
1595	23yearold motheroftwo risk banned seeing child...	gemma woman currently risk losing children hor...	0.013872	0.369231	0.095238	0.246154	0.331820	0.422000
1596	gemma 23 two children five two different fathe...	gemma woman currently risk losing children hor...	0.018012	0.264151	0.117647	0.264151	0.257653	0.402516

Appendix 31 - Fold 4 train metric score results

Test Metrics:

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
4	donald sterling racist remarks cost nba team l...	donald sterling lost nba franchise racist comm...	0.074367	0.565217	0.227273	0.434783	0.461069	0.704348
7	judge orders v stiviano pay back 26 million gi...	donald sterling lost nba franchise racist comm...	0.048369	0.490566	0.196078	0.339623	0.382368	0.602133
11	lrb cnn rrb donald sterling racist remarks cos...	donald sterling lost nba franchise racist comm...	0.055178	0.518519	0.192308	0.407407	0.505976	0.676329
16	north pacific gray whale earned spot record lo...	north pacific gray whale named varvara recorde...	0.104743	0.410256	0.210526	0.333333	0.274899	0.484469
17	north pacific gray whale earned spot record bo...	north pacific gray whale named varvara recorde...	0.126060	0.450000	0.230769	0.375000	0.375817	0.576951
...	...	...	...	...	...	...	...	...
1585	gemma 13 two children five two different fathe...	gemma woman currently risk losing children hor...	0.012861	0.312500	0.096774	0.250000	0.248242	0.312286
1592	gemma motheroftwo risk banned seeing children ...	gemma woman currently risk losing children hor...	0.012411	0.382979	0.044444	0.255319	0.188285	0.384865
1597	gemma named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.022070	0.340426	0.133333	0.297872	0.264121	0.303191
1598	woman named gemma two children five two father...	gemma woman currently risk losing children hor...	0.019355	0.339623	0.117647	0.264151	0.280927	0.414465
1599	woman two children five two different fathers ...	gemma woman currently risk losing children hor...	0.010925	0.274510	0.081633	0.235294	0.261523	0.407843

Appendix 32 - Fold 4 test metric score results

Train Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.05693432097360066, 'pearson': 0...	{'spearman': 0.0408929963772315, 'pearson': 0....	{'spearman': 0.051237546426264345, 'pearson': ...	{'spearman': 0.02537833080202288, 'pearson': 0...	{'spearman': 0.0950503029986483, 'pearson': 0....	{'spearman': 0.10863959873959152, 'pearson': 0...
coherence	{'spearman': -0.008771794779829865, 'pearson':...	{'spearman': -0.011996473247405161, 'pearson':...	{'spearman': -0.015352223291586815, 'pearson':...	{'spearman': -0.011063383311481767, 'pearson':...	{'spearman': 0.0017907921900182922, 'pearson':...	{'spearman': 0.05423758220598523 5, 'pearson': ...
fluency	{'spearman': 0.012642329560058141, 'pearson': ...	{'spearman': -0.01842714362081151, 'pearson': ...	{'spearman': 0.010104037899744312, 'pearson': ...	{'spearman': -0.009838103136187826, 'pearson':...	{'spearman': 0.00011157002183065453, 'pearson'...	{'spearman': -0.002248297523828251, 'pearson':...
consistency	{'spearman': 0.061913603385431414, 'pearson': ...	{'spearman': 0.0719076009039777, 'pearson': 0....	{'spearman': 0.08415684272611211, 'pearson': 0...	{'spearman': 0.0552870206940433, 'pearson': 0....	{'spearman': 0.10328489856963682, 'pearson': 0...	{'spearman': 0.10698273333928299, 'pearson': 0...

Test Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.011304326519480856, 'pearson': ...	{'spearman': 0.13776224807498014, 'pearson': 0...	{'spearman': 0.04334935635470928, 'pearson': 0...	{'spearman': 0.057843636471671456, 'pearson': ...	{'spearman': 0.1183967810404513, 'pearson': 0....	{'spearman': 0.20361320940110283, 'pearson': 0...
coherence	{'spearman': -0.040786798391636785, 'pearson':...	{'spearman': 0.025000680617178323, 'pearson': ...	{'spearman': -0.0148119025706696, 'pearson': -...	{'spearman': 0.02367596104715341, 'pearson': 0...	{'spearman': -0.02320091221715989, 'pearson': ...	{'spearman': 0.1033948548524454, 'pearson': 0....
fluency	{'spearman': 0.004340309164400841, 'pearson': ...	{'spearman': 0.05474979347796419, 'pearson': 0...	{'spearman': 0.026134896874761965, 'pearson': ...	{'spearman': 0.03480465949835039, 'pearson': 0...	{'spearman': 0.01676614966272229, 'pearson': 0...	{'spearman': 0.057455451709254104, 'pearson': ...
consistency	{'spearman': 0.02814627584135554, 'pearson': 0...	{'spearman': 0.01883529062545408, 'pearson': 0...	{'spearman': 0.03516265884323075, 'pearson': 0...	{'spearman': -0.0030968676499392917, 'pearson'...	{'spearman': 0.08327157501512189, 'pearson': 0...	{'spearman': 0.05915089816956408, 'pearson': 0...

Appendix 34 - Train and test correlation for fold 4

Fold 5:

Train Metrics:

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
0	donald sterling nba team last year sterling wi...	donald sterling lost nba franchise racist comm...	0.041716	0.433333	0.137931	0.300000	0.266393	0.554589
2	los angeles judge ordered v stiviano pay back ...	donald sterling lost nba franchise racist comm...	0.052569	0.535714	0.185185	0.321429	0.526042	0.665188
3	donald sterling wife sued stiviano targeting e...	donald sterling lost nba franchise racist comm...	0.015021	0.214286	0.037037	0.178571	0.125000	0.361316
4	donald sterling racist remarks cost nba team l...	donald sterling lost nba franchise racist comm...	0.074367	0.565217	0.227273	0.434783	0.461069	0.704348
5	rochelle shelly sterling accused stiviano targ...	donald sterling lost nba franchise racist comm...	0.018562	0.240000	0.041667	0.160000	0.149573	0.382943
...	...	...	...	...	...	...	...	...
1594	woman named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.016894	0.315789	0.109091	0.245614	0.276414	0.418421
1595	23yearold motheroftwo risk banned seeing child...	gemma woman currently risk losing children hor...	0.013872	0.369231	0.095238	0.246154	0.331820	0.422000
1597	gemma named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.022070	0.340426	0.133333	0.297872	0.264121	0.303191
1598	woman named gemma two children five two father...	gemma woman currently risk losing children hor...	0.019355	0.339623	0.117647	0.264151	0.280927	0.414465
1599	woman two children five two different fathers ...	gemma woman currently risk losing children hor...	0.010925	0.274510	0.081633	0.235294	0.261523	0.407843

Appendix 33 - Fold 5 train metric score results

Test Metrics:

	machine_summaries	human_summaries	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
1	donald sterling accused stiviano targeting ext...	donald sterling lost nba franchise racist comm...	0.016605	0.226415	0.039216	0.188679	0.147679	0.364471
8	donald sterlings racist remarks cost nba team ...	donald sterling lost nba franchise racist comm...	0.011230	0.270270	0.057143	0.162162	0.192936	0.388249
13	rochelle shelly sterling accused stiviano targ...	donald sterling lost nba franchise racist comm...	0.014558	0.210526	0.036364	0.140351	0.124481	0.371902
14	donald sterling racist remarks cost nba team l...	donald sterling lost nba franchise racist comm...	0.022209	0.339623	0.078431	0.264151	0.313924	0.476286
20	north pacific gray whale swam nearly 14000 mil...	north pacific gray whale named varvara recorde...	0.071570	0.277778	0.114286	0.222222	0.199690	0.423171
...	...	...	...	...	...	...	...	...
1571	manchester city want sign anderlecht teenager ...	manchester city interested signing young footb...	0.027318	0.256410	0.108108	0.256410	0.243716	0.345299
1574	manchester city keen sign anderlecht teenager ...	manchester city interested signing young footb...	0.014578	0.178571	0.074074	0.178571	0.255556	0.352381
1579	manchester city keen sign anderlecht teenager ...	manchester city interested signing young footb...	0.015423	0.185185	0.076923	0.185185	0.258137	0.355556
1584	woman named gemma two children five two differ...	gemma woman currently risk losing children hor...	0.011381	0.202899	0.059701	0.173913	0.172414	0.387681
1596	gemma 23 two children five two different fathe...	gemma woman currently risk losing children hor...	0.018012	0.264151	0.117647	0.264151	0.257653	0.402516

Appendix 36 - Fold 5 test metric score results

Train Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.040174210741610795, 'pearson': ...	{'spearman': 0.06546404771393574, 'pearson': 0...	{'spearman': 0.03673609075205126, 'pearson': 0...	{'spearman': 0.029290830901724803, 'pearson': ...	{'spearman': 0.09783805983572867, 'pearson': 0...	{'spearman': 0.12310503561772979, 'pearson': 0...
coherence	{'spearman': -0.03303169429496429, 'pearson': ...	{'spearman': -0.02473115241475901, 'pearson': ...	{'spearman': -0.040783801647470926, 'pearson':...	{'spearman': -0.032564067119709626, 'pearson':...	{'spearman': -0.0183870370068787, 'pearson': -...	{'spearman': 0.041529730367725275, 'pearson': ...
fluency	{'spearman': -0.005171887931932182, 'pearson':...	{'spearman': -0.015055386026549458, 'pearson':...	{'spearman': -0.0027436204239586516, 'pearson'...	{'spearman': -0.01479269911586921, 'pearson': ...	{'spearman': -0.007399873014971833, 'pearson':...	{'spearman': -0.002544548265088744, 'pearson':...
consistency	{'spearman': 0.037748127311725985, 'pearson': ...	{'spearman': 0.048729840139286384, 'pearson': ...	{'spearman': 0.05520008140118063, 'pearson': 0...	{'spearman': 0.0324726979108542, 'pearson': 0....	{'spearman': 0.09041715022322921, 'pearson': 0...	{'spearman': 0.08349242405991272, 'pearson': 0...

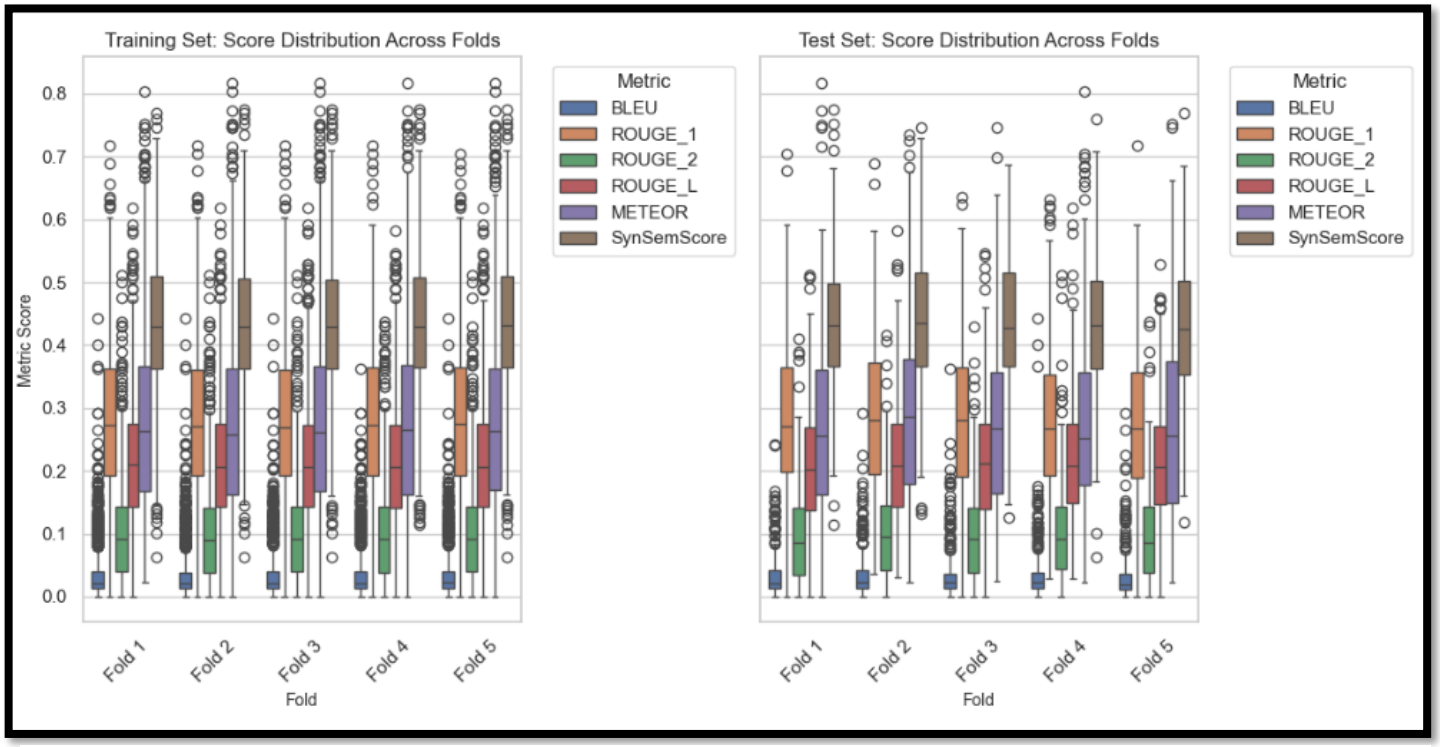
Test Correlations:

	BLEU	ROUGE_1	ROUGE_2	ROUGE_L	METEOR	SynSemScore
relevance	{'spearman': 0.07833305747866216, 'pearson': 0...	{'spearman': 0.03745044502602599, 'pearson': 0...	{'spearman': 0.09820755824226114, 'pearson': 0...	{'spearman': 0.0426541461218286, 'pearson': 0....	{'spearman': 0.10663354687679577, 'pearson': 0...	{'spearman': 0.14248303562336964, 'pearson': 0...
coherence	{'spearman': 0.05490421736430635, 'pearson': 0...	{'spearman': 0.06588841135110514, 'pearson': 0...	{'spearman': 0.07757000835052671, 'pearson': 0...	{'spearman': 0.10105144552144397, 'pearson': 0...	{'spearman': 0.05432714164520409, 'pearson': 0...	{'spearman': 0.1424589584495433, 'pearson': 0....
fluency	{'spearman': 0.0785685692225401, 'pearson': 0....	{'spearman': 0.0389328211581037, 'pearson': 0....	{'spearman': 0.06511363844082138, 'pearson': 0...	{'spearman': 0.05029775478587517, 'pearson': 0...	{'spearman': 0.04670715539559601, 'pearson': 0...	{'spearman': 0.05501401169919704, 'pearson': 0...
consistency	{'spearman': 0.1316781872872551, 'pearson': 0....	{'spearman': 0.10713530166906636, 'pearson': 0...	{'spearman': 0.15192461907049543, 'pearson': 0...	{'spearman': 0.09451181127766566, 'pearson': 0...	{'spearman': 0.1362986792352909, 'pearson': 0....	{'spearman': 0.15087692223144644, 'pearson': 0...

Appendix 35 - Train and test correlation for fold 5

```python
metrics = ['BLEU', 'ROUGE_1', 'ROUGE_2', 'ROUGE_L', 'METEOR', 'SynSemScore']

train_data, test_data = [], []

for fold_result in cv_results:
    fold_num = f'Fold {fold_result["fold"]}'

    for metric in metrics:
        train_scores = fold_result['train_df'][metric].dropna()
        train_data.extend(zip(train_scores, [fold_num] * len(train_scores), [metric] * len(train_scores)))

        test_scores = fold_result['test_df'][metric].dropna()
        test_data.extend(zip(test_scores, [fold_num] * len(test_scores), [metric] * len(test_scores)))

train_plot_data = pd.DataFrame(train_data, columns=['Score', 'Fold', 'Metric'])
test_plot_data = pd.DataFrame(test_data, columns=['Score', 'Fold', 'Metric'])

sns.set(style="whitegrid")
fig, axes = plt.subplots(1, 2, figsize=(12, 6), sharey=True)

for ax, data, title in zip(axes, [train_plot_data, test_plot_data],
                           ['Training Set: Score Distribution Across Folds', 'Test Set: Score Distribution Across Folds']):
    sns.boxplot(x='Fold', y='Score', hue='Metric', data=data, ax=ax)
    ax.set_title(title, fontsize=12)
    ax.set_xlabel('Fold', fontsize=10)
    ax.set_ylabel('Metric Score', fontsize=10)
    ax.legend(title='Metric', bbox_to_anchor=(1.05, 1), loc='upper left')

for ax in axes:
    for label in ax.get_xticklabels():
        label.set_rotation(45)

plt.tight_layout()
plt.show()
```

Appendix 38 - Metric score distribution



Appendix 37 - Metric score distribution box plot

```
print("Average Correlations (Train Data):")
avg_corr_train_df = pd.DataFrame(avg_metric_correlations_train).T
print(avg_corr_train_df)

print("\nAverage Correlations (Test Data):")
avg_corr_test_df = pd.DataFrame(avg_metric_correlations_test).T
print(avg_corr_test_df)

spearman_avg_train = avg_corr_train_df.apply(
    lambda row: np.mean([val['spearman'] for val in row]),
    axis=1
)
pearson_avg_train = avg_corr_train_df.apply(
    lambda row: np.mean([val['pearson'] for val in row]),
    axis=1
)

avg_results_df_train = pd.DataFrame({
    'avg_spearman': spearman_avg_train,
    'avg_pearson': pearson_avg_train
}, index=avg_corr_train_df.index)

print("Train data")
print(avg_results_df_train)

spearman_avg_test = avg_corr_test_df.apply(
    lambda row: np.mean([val['spearman'] for val in row]),
    axis=1
)
pearson_avg_test = avg_corr_test_df.apply(
    lambda row: np.mean([val['pearson'] for val in row]),
    axis=1
)

avg_results_df_test = pd.DataFrame({
    'avg_spearman': spearman_avg_test,
    'avg_pearson': pearson_avg_test
}, index=avg_corr_test_df.index)

print("Test data")
print(avg_results_df_test)

print("\nBest Metric (Spearman - Test Data):", best_spearman_metric)
print("Best Metric (Pearson - Test Data):", best_pearson_metric)
```

Appendix 39 - Average correlation for all folds and dimensions and overall correlation average

Average correlation for Train data

	avg_spearman	avg_pearson
BLEU	0.024919	0.039265
ROUGE_1	0.028075	0.049783
ROUGE_2	0.029933	0.050927
ROUGE_L	0.017566	0.033414
METEOR	0.049806	0.068461
SynSemScore	0.074318	0.109130

Appendix 41 - Overall Correlation scores for train data

Average correlation for Test data

	avg_spearman	avg_pearson
BLEU	0.024094	0.039091
ROUGE_1	0.027164	0.048453
ROUGE_2	0.028384	0.049308
ROUGE_L	0.018750	0.033737
METEOR	0.047063	0.065534
SynSemScore	0.073645	0.109393

Appendix 40 - Overall Correlation scores for test data

	Metric	Relevance (Spearman)	Relevance (Pearson)	Coherence (Spearman)	Coherence (Pearson)	Fluency (Spearman)	Fluency (Pearson)	Consistency (Spearman)	Consistency (Pearson)
0	BLEU	0.0476	0.0406	-0.0149	0.0182	0.0108	0.0507	0.0562	0.0476
1	ROUGE_1	0.0599	0.0748	-0.0053	-0.0003	-0.0035	0.0398	0.0611	0.0848
2	ROUGE_2	0.0489	0.0577	-0.0153	-0.0029	0.0112	0.0533	0.0750	0.0957
3	ROUGE_L	0.0316	0.0417	-0.0046	-0.0015	-0.0014	0.0304	0.0447	0.0630
4	METEOR	0.0992	0.1079	-0.0024	0.0028	0.0028	0.0397	0.0997	0.1234
5	SynSemScore	0.1272	0.1652	0.0634	0.0763	0.0093	0.0664	0.0974	0.1286

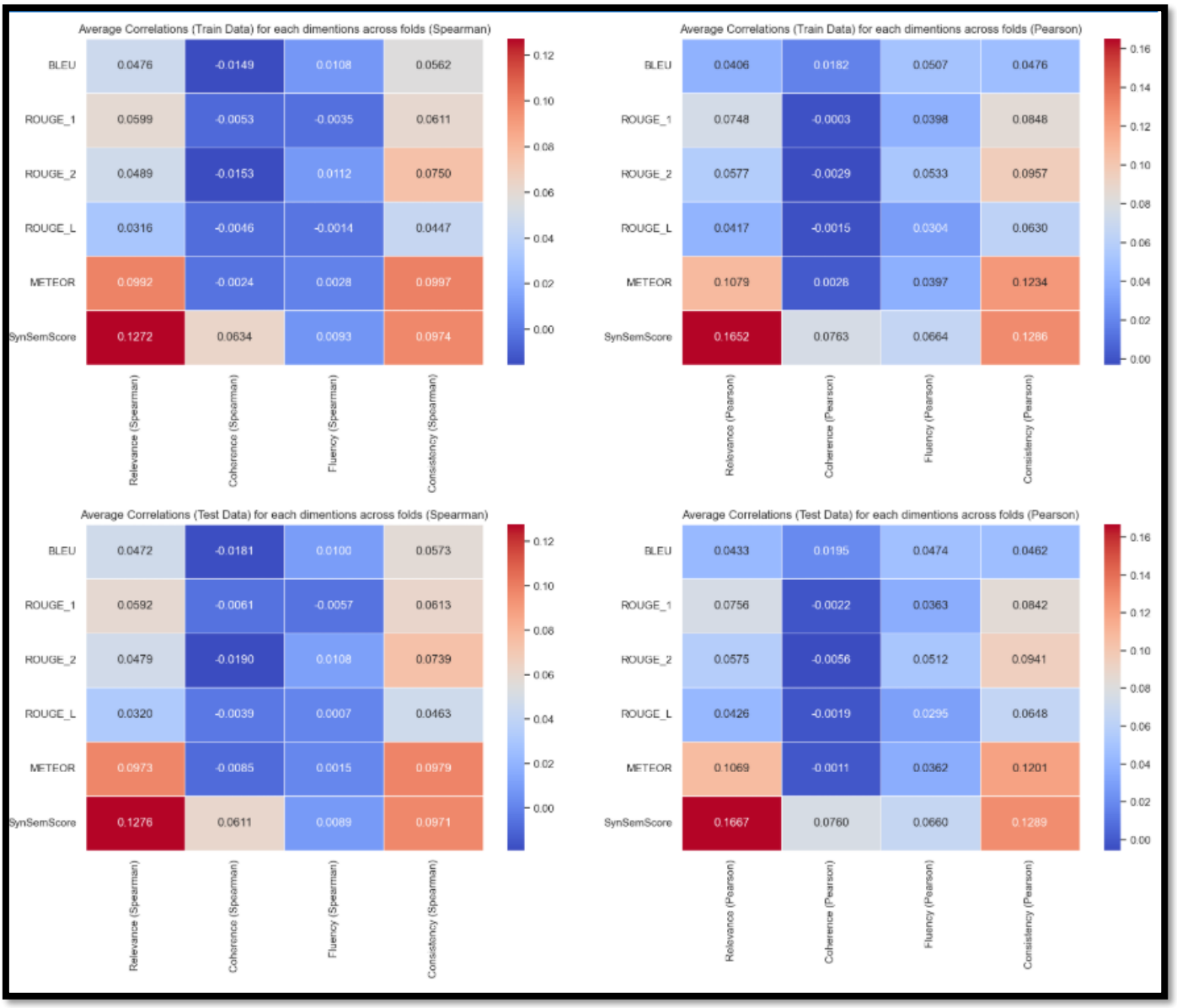
Appendix 42 - Average Correlation between metrics and dimensions for train data

	Metric	Relevance (Spearman)	Relevance (Pearson)	Coherence (Spearman)	Coherence (Pearson)	Fluency (Spearman)	Fluency (Pearson)	Consistency (Spearman)	Consistency (Pearson)
0	BLEU	0.0472	0.0433	-0.0181	0.0195	0.0100	0.0474	0.0573	0.0462
1	ROUGE_1	0.0592	0.0756	-0.0061	-0.0022	-0.0057	0.0363	0.0613	0.0842
2	ROUGE_2	0.0479	0.0575	-0.0190	-0.0056	0.0108	0.0512	0.0739	0.0941
3	ROUGE_L	0.0320	0.0426	-0.0039	-0.0019	0.0007	0.0295	0.0463	0.0648
4	METEOR	0.0973	0.1069	-0.0085	-0.0011	0.0015	0.0362	0.0979	0.1201
5	SynSemScore	0.1276	0.1667	0.0611	0.0760	0.0089	0.0660	0.0971	0.1289

Appendix 43 - Average Correlation between metrics and dimensions for test data

```
# Creating combined heatmaps for train (Spearman & Pearson)
fig, axes = plt.subplots(1, 2, figsize=(20, 6))

sns.heatmap(df1.filter(like="Spearman", axis=1), annot=True, cmap="coolwarm", fmt=".4f", linewidths=0.5, ax=axes[0])
axes[0].set_title("Average Correlations (Train Data) for each dimentions across folds (Spearman)")

sns.heatmap(df1.filter(like="Pearson", axis=1), annot=True, cmap="coolwarm", fmt=".4f", linewidths=0.5, ax=axes[1])
axes[1].set_title("Average Correlations (Train Data) for each dimentions across folds (Pearson)")

plt.show()

# Creating combined heatmaps for test (Spearman & Pearson)
fig, axes = plt.subplots(1, 2, figsize=(20, 6))

sns.heatmap(df2.filter(like="Spearman", axis=1), annot=True, cmap="coolwarm", fmt=".4f", linewidths=0.5, ax=axes[0])
axes[0].set_title("Average Correlations (Test Data) for each dimentions across folds (Spearman)")

sns.heatmap(df2.filter(like="Pearson", axis=1), annot=True, cmap="coolwarm", fmt=".4f", linewidths=0.5, ax=axes[1])
axes[1].set_title("Average Correlations (Test Data) for each dimentions across folds (Pearson)")

plt.show()
```

Appendix 44 - Heat maps creation for average correlation



Appendix 45 - Heat maps for average correlation