Investigating Abstractive Summarization with Metric Reviews, Model Experiments, and a New Consistency Score

---

A

Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Master of Science

by

Yixuan Ren

May 2025

APPROVAL SHEET

This

Thesis

is submitted in partial fulfillment of the requirements
for the degree of

Master of Science

Author: Yixuan Ren

This Thesis has been read and approved by the examing committee:

Advisor: Yangfeng Ji

Advisor:

Committee Member: Shangtong Zhang

Committee Member: Yu Meng

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

May 2025

INVESTIGATING ABSTRACTIVE SUMMARIZATION WITH METRIC REVIEWS, MODEL EXPERIMENTS, AND A CONSISTENCY SCORE

Yixuan Ren
Department of Computer Science
University of Virginia
yra7pn@virginia.edu

ABSTRACT

Despite their impressive performance in natural language generation tasks, Large Language Models (LLMs) still face critical challenges in text summarization. In particular, the performance of LLMs in abstractive text summarization and the limitations of existing evaluation frameworks warrant further investigation. In this work, we present a comprehensive analysis of summarization evaluation metrics, covering lexical overlap, semantic distance, factual consistency, and recent LLM-based methods. Employing these metrics as evaluation tools, we empirically assess the performance of summarization models across the LLaMA, and Gemma model families, utilizing datasets from diverse domains to provide an examination of the capabilities of current LLMs in abstractive text summarization tasks. To address limitations of current metrics, we introduce the concept of self-consistency and propose a novel consistency score to assess the reliability of text summarization models.

***K*eywords** Abstractive Text Summarization · Text Generation · Natural Language Processing

1 Introduction

Abstractive text summarization plays a crucial role in Natural Language Processing (NLP) to generate informative and concise summaries in a natural and readable format, allowing people to understand articles rapidly. With the development of Large Language Models (LLMs), many works produced the promising performance of zero-shot LLMs on abstract summarization tasks [1] [2], demonstrating that state-of-the-art (SOTA) LLMs perform on par with human-written summaries. This reveals the great potential of LLM as an excellent text summarization tool.

Current research on abstractive summarization mainly focuses on faithfulness metrics, prompting engineering, and instruction tuning. For example, some research [3] [4] dedicate to applying LLMs to provide several benchmark datasets and metrics, evaluating summarization from different aspects. SumCoT [5] presents COT-based prompts in abstractive summarization by progressive generation, guiding LLMs to incorporate details into final summaries. Besides, [6] proposes the Dense of Thoughts (DoT), which generates less biased summaries by collecting GPT-4 [7] outputs. Instruction tuning is also a popular topic to improve models for abstractive summarization in specific domains. For example, KEITSum [8] fine-tunes a small-scale LLM by identifying key elements and instructing the LLM to generate summaries with key elements. However, in the field of text summarization using large language models (LLMs), there is a lack of systematic categorization and analysis of limitations in existing metrics, as well as insufficient experimental evaluation of summarization performance across diverse subject areas and mainstream LLMs, particularly those employing zero-shot approaches.

In this work, we conduct a comprehensive survey of text summarization evaluation metrics, encompassing a wide range of approaches. Our investigation covers traditional lexical overlap metrics such as ROUGE [9] and BLEU [10], semantically informed metrics like BERTScore [11] and MoverScore [12], and factual consistency indicators, including classification-based and QA-based methods [13, 14]. Additionally, we explore recent LLM-based metrics [15] and generalized text generation evaluation approaches [16]. We analyze the underlying strategies, inherent strengths and limitations, and specific application domains of typical metrics. Our aim is to provide a thorough survey of text summarization evaluation metrics, offering researchers a holistic perspective.

After systematically investigating metrics, we selected 13 metrics, including reference-based, source-based, and LLM-based types, to evaluate the performance of LLaMA [17] and Gemma [18] in abstractive text summarization. The datasets we selected consist of news, academia, and stories, including classic CNN/DM [19], XSum [20], the latest manually annotated AnnotationNews , StorySumm , and Arxiv 2025. We analyzed the experimental results from evaluation dimensions, dataset topics, and overall model performance, aiming to provide an analysis of the performance of mainstream LLMs in abstractive text summarization tasks.

Considering the limitation of existing metrics, we first formulate a mathematical definition of reliability based on multiple outputs. We define summarization reliability as the consistency of the summarization system outputs for a given source document. This definition is inspired by the observation of the recent LLM summarization systems and aims to provide a complementary measurement of system performance in addition to standard evaluation metrics (e.g., ROUGE and BERTScore). In the zero-shot setting, existing LLMs can generate a summary that considerably overlaps with the reference summary. This explains the promising system performance under existing metrics.

However, we argue that (1) the reliability of a summarization system cannot be measured with single outputs, and (2) the problematic summary is often caused by the difference between multiple outputs from the same input. Specifically, we propose to measure the system outputs with a novel approach that calculates the score of summarization output overlap, quantifying it as the consistency score. This simple measurement does not require deep semantic understanding, which by itself is still an open question, and is sufficient enough to reveal some limitations of existing summarization systems, as demonstrated in 6.

In summary, our contributions include:

- We conduct a detailed analysis and classification of existing text summarization metrics, highlighting their strengths and limitations.
- We evaluate the general performance of abstractive text summarization tasks by assessing six LLMs, across six datasets from diverse fields and thirteen metrics of varying dimensions.
- We introduce a novel consistency score pipeline for evaluating LLM reliability in text summarization, which computes overlap scores based on multiple model outputs after merging words with semantic similarity.

2 Background

2.1 Abstractive Summarization in LLM Era

In the domain of NLP, abstractive text summarization evolved from extractive text summarization, aiming to produce smoother and more concise summaries. Entering the era of deep neural networks, abstractive summarization has made significant progress with Seq2Seq models [21] [22] and attention mechanisms [23], which allow the model to map relationships between input text and output summary by flexibly focusing on the important parts of the original text.

In recent years, there have been two major upgrades in model size and data magnitude, leading people to focus on different approaches to abstractive summarization. The first upgrade, to models with billions of parameters and large-scale datasets [21] [24], introduced pretrained language models (PLMs) such as BART [21] and RoBERTa [24], using bidirectional encoding and autoregressive decoding to greatly enhance summarization performance and giving rise to many fine-tuned PLMs [22] [25] [26] for abstractive summarization. The second upgrade, to models with hundreds of billions of parameters, brought forth LLMs. The main difference in handling NLP tasks is that LLMs are able to perform well via zero-shot prompts. However, current research on abstractive summarization using zero-shot LLMs primarily focuses on human evaluation of large-scale closed-source models [2]. We chose LLaMA [17] and Gemma [18] models as baselines to provide a perspective on small-scale open-source models, complementing existing evaluation results with a starting point for analyzing performance in summary generation within LLMs.

2.2 Consistency in LLM summarization

Researchers point out that when LLMs are hesitant and hallucinate about some content, they tend to produce inconsistent responses to the same input. This assumption is initially from self-consistency [27]. Following that, to generalize self-consistency into universal tasks, universal self-consistency [28] is presented by constructing a prompt that guides the model to choose the most consistent response by itself, resulting in improvement of reliability.

Additionally, SliSum [29] applies consistency to improve model performance in summarization, focusing on local-output consistency and logical consistency, respectively. SliSum divides the source article into multiple overlapping windows and aggregates local summaries by voting to improve fidelity, but with a high complexity resulting in the time spent being approximately double that of the original baseline. In addition, SelfCheckGPT [30] assumes that if the LLM

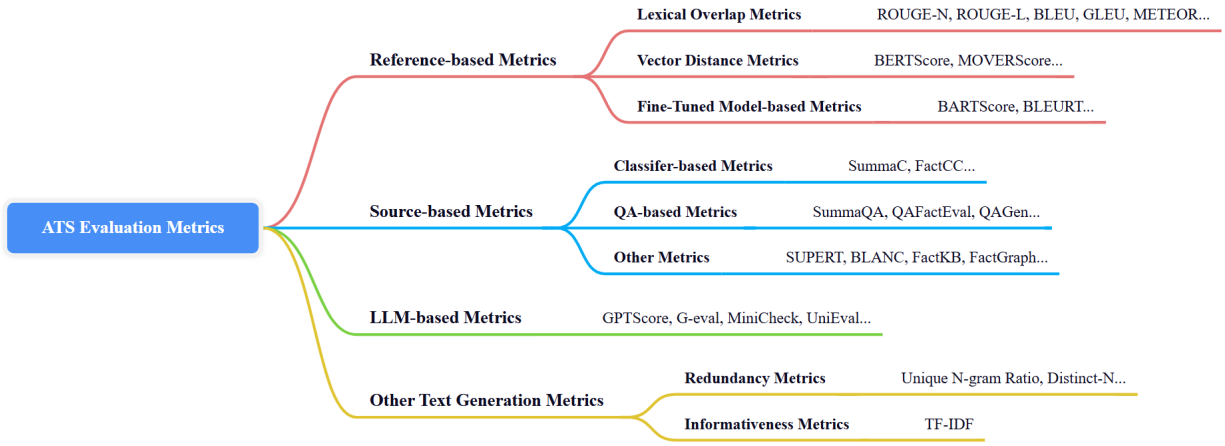

Figure 1: Current Text Summarization Metrics

understands the given concept, the sampled response is likely to contain consistent facts. However, for hallucinations, randomly sampled responses may appear close but contradict each other. Using a sampling-based method, it is possible to detect non-factual and factual sentences and sort paragraphs according to factuality.

These works inspired us to design a novel consistency score that introduces the concept of self-consistency as an evaluation metric to smooth out the interruptions in text summarization systems.

3 Automatic Metrics in Text Summarization

To align with human evaluation, **relevance**, **coherence**, **fluency**, and **informativeness** have consistently been the focus of text summarization evaluation. To efficiently measure these dimensions, various strategies for automatic metrics have emerged. We primarily categorize these metrics from a dependency perspective into reference-based metrics and source-based metrics, which respectively rely on reference summaries and source text to evaluate the performance of summarization systems. Reference-based metrics often employ different strategies to measure the similarity between reference summaries and generated summaries, while source-based metrics aim to check whether the generated summaries are faithful to the original text, usually serving as a dimension for evaluating factual consistency. Furthermore, we separately categorize LLM-based metrics to introduce the novel metrics based on the latest LLM technologies. Additionally, we also mention some general metrics for text generation tasks, such as redundancy and readability, which also offer perspectives for evaluating summary quality.

3.1 Reference-based Metrics

3.1.1 Lexical Overlap Metrics

Early in the development of NLP tasks, lexical overlap metrics were the main tools for evaluating text generation tasks, such as ROUGE-N (Equation 1) and ROUGE-L (Equation 2).ROUGE-N measures the recall by calculating the degree of N-gram overlap between predicted summaries and reference summaries, which reflects the coverage of the generated text at the N-gram level; while ROUGE-N measures the recall by calculating the degree of N-gram overlap between predicted summaries and reference summaries, which reflects the recall of the generated text. text coverage at the N-gram level, while ROUGE-L evaluates the similarity of sequence structures based on the longest common suffix (LCS) from the recall perspective. Both metrics are still frequently used in current text summarization tasks.

$$\text{ROUGE-N} = \frac{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{\text{gram}_n \in S} \min\left(\text{Count}_{\text{match}}(\text{gram}_n), \text{Count}_{\text{ref}}(\text{gram}_n)\right)}{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{ref}}(\text{gram}_n)} \tag{1}$$

$$\text{ROUGE-L} = \frac{\text{LCS}(\text{Reference}, \text{Prediction})}{\text{len}(\text{Reference})} \tag{2}$$

In addition, there are BLEU [10] and its variant GLEU [31], which are often used as supplements for summary quality assessment; BLEU evaluates the quality of generated summaries from the accuracy point of view mainly by calculating the n-gram matches between predicted summaries and reference summaries; while GLEU, as an improved version of BLEU, further optimizes the evaluation of the short texts and the diversity of generation. For each n-gram, GLEU counts the number of times it occurs in the generated text and the reference sentence, and then takes the smaller value of the two times as the number of valid matches of the n-gram, in order to avoid the under-penalization or over-penalization problem that may occur in BLEU. METEOR [32], on the other hand, supplements the lexical matching with the consideration of synonyms, stemmed forms and semantically similar expressions, combining precision, recall and semantic alignment to further improve the comprehensive evaluation of the quality of the generated text.

Highlights Vocabulary overlap metrics are simple, easy to understand, and highly interpretable. We can easily understand the logic behind them. In addition, evaluating text with these metrics is fast and extremely inexpensive to implement.

Limitation However, the limitations of these metrics are equally significant. As a purely statistical method, lexical overlap metrics are not directly related to semantic content and logical structure, and can only rely on superficial formal matches that do not capture the deeper meaning or contextual coherence of the text. For example, summaries with perfectly correct semantics but different wordings may be rated low, while summaries with high lexical overlap but confusing logic may receive high scores. In addition, the lack of sensitivity of such metrics to synonymous expressions, sentence variations, or creative uses of language makes it difficult to effectively assess the quality of predictions in complex task generation tasks.

3.1.2 Vector Distance Metrics

In the process of developing pre-trained language models (PLM), the researchers realized the advantage of embedding as a textual representation - it can capture semantic information to a certain extent, thus compensating for the inadequacy of statistical lexical methods in understanding the meaning of text. Based on this realization, they proposed evaluation metrics such as BERTScore [11] and MOVERScore [12].

BERTScore (Equation 3) utilizes the contextual word embeddings generated by BERT to measure the degree of semantic agreement between two texts by calculating the cosine similarity between each word in the generated summary and each word in the reference summary, and adopting the greedy maximum matching strategy. This approach not only takes into account the surface form of the words, but also reflects the semantic information related to the context, making the evaluation results closer to human judgment of text similarity.

$$\text{BERTScore} = F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{3}$$

- $P = \frac{1}{|\hat{x}|} \sum_{\hat{x}_i \in \hat{x}} \max_{x_j \in x} \cos(\hat{x}_i, x_j)$ refer to the Precision, calculating the average cosine similarity of each token in the generated sentence $\hat{x}$ with the most similar token in the reference sentence x;
- $R = \frac{1}{|x|} \sum_{x_j \in x} \max_{\hat{x}_i \in \hat{x}} \cos(x_j, \hat{x}_i)$ refer to the Recall, calculating the average cosine similarity between each token in the reference sentence x and the most similar token in the generated sentence $\hat{x}$;
- $\cos(\cdot, \cdot)$ represents the cosine similarity between the contextual embeddings of two tokens.

MOVERScore (Equation 4) goes a step further on this basis. It combines word embedding with optimal transmission theory to provide a more comprehensive measure of text similarity by calculating the Wasserstein distance between the generated summary and the reference summary in the semantic embedding space. MOVERScore not only focuses on the matching of local word pairs, but also takes into account the alignment of the overall semantic structure, and thus it can better deal with long texts or summaries with complex semantic distribution. These metrics redefine text similarity evaluation from the perspective of word embedding, and provide a more refined analytical tool for semantic alignment between generated summaries and reference summaries.

$$\text{MOVERScore} = F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{4}$$

Where:

- $P = \frac{1}{|\hat{x}|} \sum_{\hat{x}_i \in \hat{x}} \max_{x_j \in x}(1 - WMD(\hat{x}_i, x_j))$ represents precision, which calculates the similarity between each token in the generated sentence $\hat{x}$ and the most matching token in the reference sentence x;
- $R = \frac{1}{|x|} \sum_{x_j \in x} \max_{\hat{x}_i \in \hat{x}}(1 - WMD(x_j, \hat{x}_i))$ Recall, which calculates the similarity between each token in the reference sentence x and the most matching token in the generated sentence $\hat{x}$;
- $WMD(\cdot, \cdot)$ represents the Word Mover's Distance, which calculates the distance between two tokens based on word embedding.

Highlights These metrics introduce word embeddings to represent semantic information, greatly compensating for the shortcomings of traditional lexical overlap metrics. By capturing the semantic relationship between words, they can reflect the meaning of the text to a certain extent, rather than limiting themselves to superficial lexical matching, thus improving the depth and accuracy of the evaluation.

Limitation However, these word embedding-based metrics also have obvious drawbacks. First, they are not sufficiently stable, which means that the evaluation results are highly dependent on the pre-trained word embedding model used. Different models or training data can lead to significant differences in results. Second, word embeddings have limited representativeness and cannot fully capture complex syntactic structures or long-distance dependencies. For example, BERTScore may suffer from scoring bias when dealing with texts with the same semantics but different sentence structures, and MOVERScore may perform poorly in short text evaluation when the computational complexity is high. These issues limit their reliability and generalizability in practical applications.

3.1.3 Fine-tuned Model Metrics

In addition to designing algorithms to compare the similarity between predicted summaries and reference summaries by calculating the distance between word embeddings, the researchers explored another idea - directly fine-tuning the pre-trained model as an evaluation metric. This approach capitalizes on the powerful language comprehension capabilities of PLM by fine-tuning them for a specific task and directly outputting a score for the quality of the generated text.

BARTScore [16] is representative of this idea. As the calculation in Equation 3.1.3 it assumes that high-quality summaries should have a higher probability of being generated, transforms summary evaluation into a conditional generation task, and utilizes the model's likelihood estimation capabilities to measure the performance of predicted summaries in terms of semantics, fluency, and fidelity. Similarly, BLEURT [33] focuses on the semantic consistency between the generated text and the reference text by fine-tuning the pre-trained model.BLEURT first performs supervised training on large-scale manually labeled scoring data to enable it to predict the quality scores of the generated text with respect to the reference text.BLEUR combines pre-training representations with supervised learning, which is more sensitive to semantic nuances and contextual dependencies. sensitive to semantic nuances and contextual dependencies.

$$\text{BARTScore} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_i \in \hat{x}} \log P(\hat{x}_i | x; \theta) \tag{5}$$

Where:

- $\hat{x}$ represents the generated sentence, x represents the reference sentence;
- $P(\hat{x}_i | x; \theta)$ represents the conditional probability of the tag $\hat{x}_i$ in the generated sentence given the reference sentence x and the model parameter θ, calculated by the BART model;
- $|\hat{x}|$ represents the length of the generated sentence $\hat{x}$.

Highlights These trainable models can be flexibly adapted to the requirements of different languages and tasks, taking full advantage of the powerful semantic comprehension capabilities of PLM that can capture lexical, syntactic, and paragraph-level semantic information as well as textual coherence beyond the limitations of the traditional word-embedding distance computation.

Limitations However, there are also risks to the stability and generalizability of these metrics. Evaluation results often depend on the pre-trained model used and its fine-tuning process. For example, BARTScore scores may vary depending on the BART model version or training data, while BLEURT performance is limited by the quality and coverage of the supervised training data. Second, these metrics are computationally expensive, limiting their application in resource-limited environments. And the interpretability of these metrics is poor. Due to the black-box principle of neural networks, the exact reason behind similarity scores is not intuitive.

3.2 Source-based Metrics

For text summarization or even text generation tasks, in addition to assessment using reference texts, assessment can be performed using source texts such as FactCC [34], FactKB [35] and SummaC [13], to name a few, which provide assessment in the absence of high-quality references. These metrics assess the generated text by directly comparing it to the source document, usually focusing on factual consistency, implication or informativeness. Unlike reference-based evaluation metrics, simple overlap or similarity cannot be used to directly compare source documents and abstracts due to differences in structure and length. Therefore, source-based evaluation metrics usually involve training models or specific analysis modules, such as Named Entity Recognition (NER), Knowledge Graph Construction or Syntactic Analysis, Factual Consistency Verification, etc., to refine key information in the original text and abstracts through additional modules to measure the consistency of the abstracts with the original text in terms of information retention, logical consistency and factual correctness.

3.2.1 Classifier-based Metrics

SummaC is a PLM-based classifier designed to evaluate whether the summaries are faithful to the original source documents. It is trained using a natural language reasoning-based approach and is able to detect misstatements or information omissions in the generated summaries. The model operates by performing sentence-level comparisons between the summaries and the source text and gives a fidelity score that reflects how well the summaries agree with the information in the original document.

Moreover, FactCC (Equation 3.2.1) is a tool that specializes in detecting whether machine-generated summaries contain factual errors. It is trained using an adversarial data augmentation strategy and enhances the robustness of the model by artificially constructing erroneous summaries (e.g., substituting entities, numbers, causal relationships, etc.). A classification model is then used to determine whether the content of the summary is consistent with the source document.

$$\text{FactCC}(\hat{x}, d) = \begin{cases} 1 & \text{if } P(\text{Consistent}|\hat{x}, d; \theta) > \tau \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Where:

- $\hat{x}$ represents the generated sentence, d represents the reference document;
- $P(\text{Consistent}|\hat{x}, d; \theta)$ represents the probability that the NLI model (with parameter θ) predicts that the generated sentence $\hat{x}$ is consistent with the reference document d;
- τ represents the classification threshold, which is usually set to 0.5.

Highlights The advantage of this trained classifier is that it uses a natural language inference (NLI) method that specifically detects logical and semantic consistency between the summary and the source document, rather than just superficial text matching. Furthermore, the data used to train this evaluation classifier is often manually constructed with positive and negative examples. By synthesizing data, researchers can customize and cost-effectively construct evaluation classifiers, making them more sensitive to targeted data and factual errors.

Limitations Like fine-tuned model evaluation metrics, classifier-based metrics rely on training data and the generalization ability of the model, and may have limited adaptability to different fields. Since it directly uses the end-to-end model, it cannot analyze the error types in detail, resulting in low interpretability.

3.2.2 QA-based Metrics

In addition to classifier-based metrics, question-answering QA-based metrics are also widely used to assess the factual consistency of summaries. The basic assumption is that a high-quality summary should retain the key factual details from the original text, and thus be able to accurately answer the questions derived from it. QA-based metrics automate this process using pre-trained QA models, which effectively detect omissions, illusions and distortions in the generated summary. They have been shown to correlate closely with human judgments of factual consistency.

SummaQA [36] generates questions from the source document and checks whether the summary can answer these questions correctly. QAFactEval [14] generates questions from the summary and assesses whether the original text can answer them correctly. This process involves question generation and answer checking. For question generation, these metrics usually use Cloze question generation to generate questions from key information in the original text or summary, then have the QA model answer the questions based on the summary or original text, and finally use exact match or F1 score to check the degree of match between the summary answer and the answer. These metrics

assume that the degree of match between question answers reflects the degree of match between the original text and the summary, that is, the fidelity of the generated summary.

In addition to SummaQA and QAFactEval, QuestEval [37]and FEQA [38] are also popular QA-based metrics. They focus on illusion recognition, information omission and distortion by generating questions and answers from the generated summary and source text.

Highlights QA-based metrics do not give a single overall score, but identify specific factual inconsistencies by verifying whether the summary answers the source question correctly. These metrics improve interpretability to some extent. If the summary does not answer a question correctly, it means that this piece of information may be missing or incorrect. And because the method is based on fact-based questions and answers, it can be applied to various fields without the need for a large amount of domain-specific annotations.

Limitations The quality of the indicators based on quality assurance largely depends on the quality of the quality assurance model. If the model performs poorly, it may generate meaningless questions or fail to correctly locate the relevant answers in the original text, leading to misjudgments and affecting the final score.

3.2.3 Other Source-based Metrics

Additionally, there are also some innovative metrics using various methods to measure the faithfulness of generated summaries from different perspectives. For example, **FactGraph** [39] parses the summary and the original text into triples based on the knowledge graph, and matches the factual relationship to determine the consistency. **FactKB** [35] extracts entities and relationships from the summary, and then queries the external knowledge base to evaluate whether the summary contains false facts. The specific calculation process of FactKB refers to Equation 3.2.3. **SUPERT** [40] uses a text clustering algorithm to extract important sentences in the original text and compares them with the reference summary to determine the similarity. **BLANC** [41] masks some words in the source text, and then lets the model supplement the mask based on the summary to evaluate the fluency and information richness. These metrics have different focuses and aim to provide more fine-grained automated quality assessment and improve the reliability of summary evaluation from multiple dimensions..

$$\text{FactKB}(\hat{x}, d) = \frac{1}{|\hat{x}|} \sum_{\hat{x}_i \in \hat{x}} P(\text{Consistent}|\hat{x}_i, d; \theta) \tag{7}$$

Where:

- $\hat{x}$ represents the generated text (such as summary), d represents the reference document or knowledge base;
- $\hat{x}_i$ represents the ith sentence or token in the generated text;
- $P(\text{Consistent}|\hat{x}_i, d; \theta)$ represents the probability that the pre-trained model (with parameter θ) predicts that the sentence $\hat{x}_i$ is consistent with the reference document d;
- $|\hat{x}|$ represents the total number of sentences or tokens that generate the text $\hat{x}$.

3.3 LLM-based Metrics

In recent years, with the rise of billion-level large models such as GPT, researchers have also begun to consider using their powerful semantic understanding capabilities as evaluation tools. The first attempt was **GPTScore** [42], which is similar to BART and can calculate the probability of generating a summary to illustrate the degree of match between the summary and the original text. However, GPTScore is based on the GPT-3/4 model, which has stronger generalization capabilities and does not require fine-tuning. GPTScore has been extensively experimented with on four text generation tasks, 22 evaluation aspects, and 37 corresponding datasets, and has achieved results that are far superior to ROUGE scores and similar to human ratings.

G-eval [43] uses LLM to directly assess the quality of the summary. It designs specific prompts for LLM to score from the dimensions of coherence, consistency, fluency, and relevance, and returns a score of 1-5. The prompts used by G-Eval are combined with a chain of thoughts to guide LLM in step-by-step reasoning, improving the stability and interpretability of the assessment. Their experiments show that G-Eval, with GPT-4 as the backbone model, has a Spearman's correlation coefficient of 0.514 with humans in the summary task, which is much better than most existing evaluation methods.

In addition, the researchers also proposed the lightweight **UniEval** [44] and **MiniCheck** [45] for source text-based text summary evaluation. UniEval converts the criteria into a question-answering task. It can evaluate reference-free

summaries, offers versatility and supports fine-tuning. MiniCheck focuses on verifying facts and assessing factual consistency in generated text. It assesses consistency by comparing generated text with the source text, without reference summaries, and focuses on hallucination detection. MiniCheck uses GPT-4 to construct synthetic training data to build a small fact-checking model with GPT-4-level performance but 400 times lower cost. The synthetic data is created from real but challenging factually incorrect examples through a structured generation process in order to teach the model to check each fact in the generated summary and identify syntactic information between sentences.

These LLM-based Metrics explore the foundations of traditional frameworks and extend classical approaches by leveraging the powerful semantic understanding and generation capabilities of LLMs, with the aim of aligning with human assessments.

Highlights Due to LLM's strong language comprehension capabilities, LLM-based metric scoring is usually closer to manual scoring than traditional automatic evaluation metrics. The correlation between most LLM-based metrics and human assessment is far higher than that of traditional Lexical Overlap. In addition, since LLM can be guided by fine-tuning and prompting engineering, it provides researchers with more ways to evaluate model modeling than full parameter training models, enhancing the flexibility and comprehensiveness of evaluation.

Limitations However, the computational resource consumption of LLM is clearly a challenge. For closed-source models, calling the API to deploy LLM-based Metrics will be billed in tokens. For open-source models, deploying an LLM with hundreds of billions of parameters for inference and evaluation requires a lot of computational resources. Whether training, inferring, or evaluating LLM, the trade-off between effectiveness and cost is always an unavoidable issue.

3.4 Other metrics for text generation

3.4.1 Redundancy

For redundancy, commonly used metrics in text generation include N-gram repetition and Distinct-n [46]. N-gram repetition measures the frequency of repeated n-grams in a text. It assumes that a high frequency of N-gram repetition indicates redundancy. Distinct-n calculates the diversity of n-grams in a text. For example, Distinct-1 and Distinct-2 measure the ratio of unique single letters and bigrams relative to the total number of tokens. For Distinct-n, lower scores indicate fewer unique n-grams and higher redundancy.

3.4.2 Informativeness

An equally important dimension as redundancy is information content. From the perspective of human evaluation, a high-quality summary should contain rich information, which is consistent with the purpose of the summary system. Commonly used informativeness metrics include content density and TF-IDF [47]. Content density uses techniques such as named entity recognition to calculate the ratio of unique concepts or entities to the length of the text, thereby measuring the amount of meaningful or unique information relative to the total number of words. TF-IDF identifies keywords by evaluating their frequency in the generated text relative to their prevalence in the broader corpus. A high TF-IDF score for a keyword indicates that the text emphasizes important concepts rather than generic or overly common content.

4 Experiments

4.1 Datasets

4.1.1 News Summarization Datasets

For news summarization tasks, we select **CNN/Daily Mail** [19] and **XSum** [20] which are two widely used benchmarks for text summarization. CNN/DM consists 312,085 news articles from CNN and Daily Mail. XSum consists 226,711 news articles from BBC. Both of them are with the corresponding human-written summaries. Instead of sampling from the dataset, we evaluate the model performance based on the entire test sets with 11,490 samples and 11,334 samples to reflect a comprehensive results.

Considering researchers [1] have raised the issue of low quality of reference summaries of CNN/DM and Xsum, we also select the **Annotation News dataset** proposed in their work including 600 news articles based on CNN/DM and Xsum and rewrote high-quality summaries. In our work, we select 482 news articles and their summaries with the most consistent human-written summaries for experiments.

4.1.2 Scientific Paper Summarization

For the summarization task in the academic paper domain, we utilize the research paper abstract dataset released by [23] in 2018. To account for the possibility that early academic datasets might have been included in LLM pre-training, we also collect the latest computer science papers published in 2025 from arXiv to evaluate LLM performance. Given the computational resource constraints related to full-text input length, we select the introduction section of each paper as the model's input and used the abstract as the reference summary.

4.1.3 Story Summarization

In addition to traditional news and academic paper summarization, we explore the task of story summarization. For this, we select the StorySumm dataset [48], which contains 32 short stories sourced from Reddit along with manually crafted story summaries.

4.2 Base Models

Table 1: Experimental models version and sizes

Model	Version (Size)
LLaMA	2 (7B) 3 (8B), 3.1 (8B)
Gemma	1 (7B), 1.1 (7B), 2 (9B)

As shown in Table 1, we select the LLaMA 2 [17], LLaMA 3, LLaMA 3.1, and Gemma [18, 49] models to perform zero-shot inference on text summarization. LLaMA and Gemma are both highly popular within the open-source model community.

LLaMA series. LLaMA is developed by Meta AI that outperforms GPT-3 [50] in various downstream tasks. It is based on the Transformer architecture and focuses on smaller model sizes. The model of version 3.1 with 8B parameters performs comparable to larger models in multiple benchmarks.

Gemma series. Gemma is proposed by Google, focusing on training and reasoning in certain fields such as health, science and technology, etc., on large-scale datasets with professional knowledge. The framework of Gemma is similar to GPT-3, within transformer decorder-only architecture.

4.3 Experimental Setup

The experiments are conduct in the UVA Rivanna computing environment equipped with NVIDIA A100 GPUs (80GB memory), running on a Linux-based system with CUDA 12.8. The models are implemented using the Hugging Face Transformers library along with PyTorch 2.1.0. For inference, we use a batch size of 2 and a maximum generation length as the average length of the reference summaries in the evaluation datasets. Statistics of Datasets is shown in Table 2.

Table 2: Statistics of the test sets across six datasets

	CNN/DM	XSUM	Annotation News	Arxiv 2018	Arxiv 2025	StorySumm
# Samples	11,490	11,334	482	424	497	36
Text Avg. # Word	652.4	429.7	550.2	1536.5	1310.3	312.9
Sum Avg. # Word	47.5	23.1	40.7	210.6	243.8	75.3

4.4 Evaluation Metrics

Following our analysis in Sec.3, we select several typical metrics to measure the performance from different perspective. Specifically,

- For lexical overlap, we apply **ROUGE-1**, **ROUGE-L**, **BLEU**, **GlEU** and **METEOR**.

- For semantic similarity, we apply **BERTSc**, **BARTSc**, and **BLEURT**.
- For factual-consistency, we conduct **FactCC** and **FactKB** to measure the faithfulness of generated summaries.
- And **G-Eval** is applied to evaluate the general performance using LLMs. It includes four dimension of Fluency, Coherence, Relevance, Consistency.

4.5 Main Results and Analysis

The complete experimental results include twelve tables, and the appendix presents the results of each indicator in detail. In this section, we will summarize and analyze the main results and rules with intuitive visualization.

4.5.1 General Performance

To obtain the overview of the performance of the models, for each model, we average the metrics of all datasets using *np.mean*, and then normalize the average data to a 0-1 scale using min-max normalization to ensure that all metrics are on the same scale for fair comparison in the radar chart. As shown in Figure 2 and Figure 3, the two radar charts are one for the main metrics (R1, RL, etc.) and the other for G-eval metrics (fluency, coherence, etc.). We can find that LLaMA 3 and LLaMA 3.1 perform well on both traditional metrics and G-Eval, and far exceed Gemma 7B. From the table in the appendix, we can also observe that LLaMA 3 and LLaMA 3.1 perform best on most datasets and indicators. In addition, Gemma 2 9B performs well on non-lexical metrics, which may indicate that it has good similarity and fidelity at the semantic level.

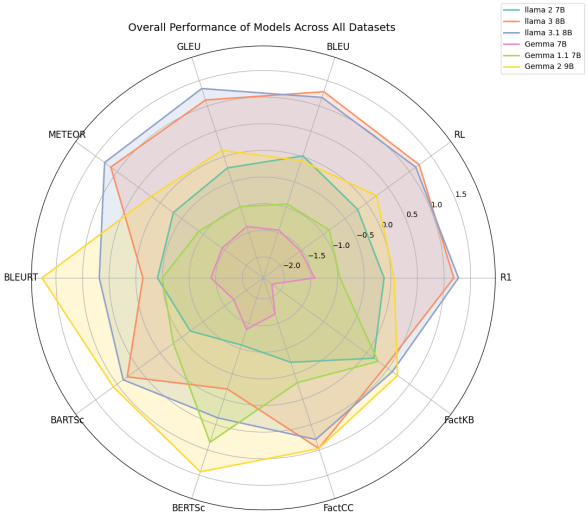

Figure 2: Overall Performance of LLMs

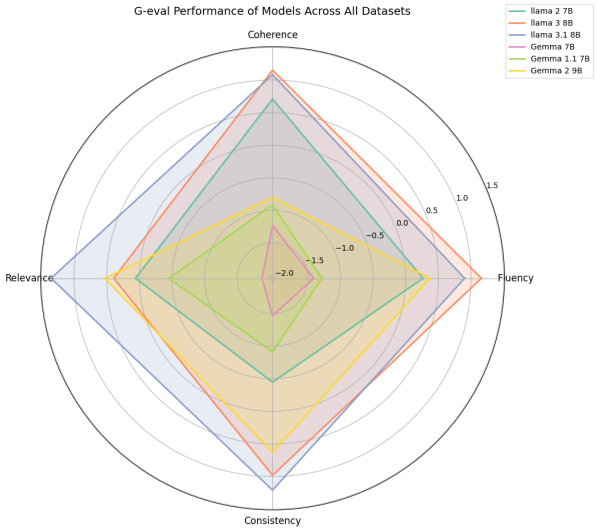

Figure 3: G-eval Performance of LLMs

4.5.2 Task Performance

To evaluate the overall performance on different types of tasks, we group the metrics, standardized the metrics within each group, and average the models for each dataset. As shown in Figure 5, Factual_consistency represents the normalized average of FactCC and FactQA, G_eval represents the average performance of its four internal dimensions, and Text_similarity represents the average performance of the remaining reference-based metrics. For the performance differences between different metric groups, we can found that academic abstract generation performs well in the factual consistency dimension, indicating that the abstract generated by LLM is faithful to the original text. However, the similarity with the reference abstract is not limited, especially on the CNN/DM and XSum datasets. This shows that the generated summaries still have a certain distance away from the reference abstract.

4.5.3 Additional Results

From the perspective of similarity indicators, in the task of academic article summarization, the introduction of complex prompts with structural information and detail restrictions can bring a slight improvement in performance. To explore

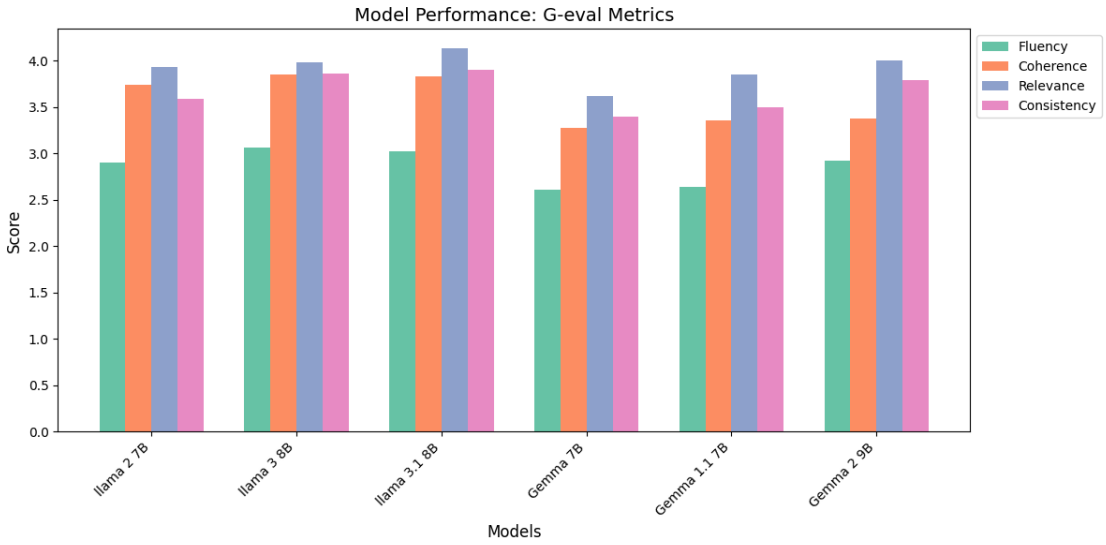

Figure 4: G-Eval Performance Across All Datasets

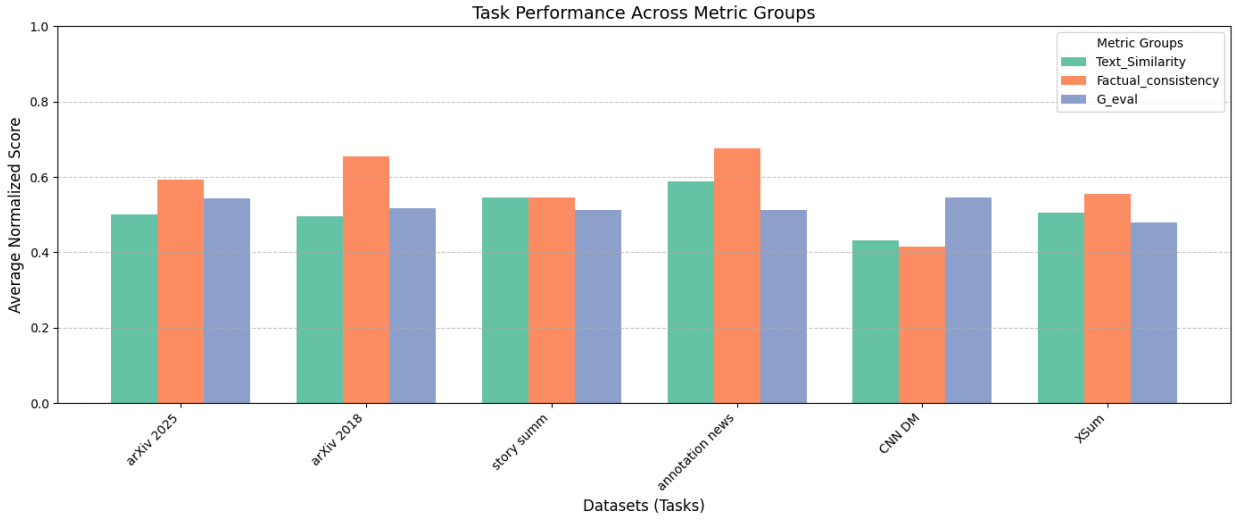

Figure 5: LLM Performance Across Diverse Task

Table 3: LLM Performance based on different Prompts on ArXiv 2025

Model	R1	RL	BLEU	GLEU	METEOR	BLEURT	BARTSc	BERTSc	FactCC	FactKB
Prompt_1	46.23	24.84	9.46	13.65	30.29	-0.35	-4.93	85.20%	0.64	97.09%
Prompt_2	45.92	24.01	9.22	13.17	30.21	-0.31	-4.45	85.36%	0.65	97.65%
Prompt_3	45.74	23.65	9.67	13.68	29.72	-0.37	-4.03	85.39%	0.67	98.16%

Table 4: LLM G-eval Performance based on different Prompts on ArXiv 2025

Model	Fluency	Coherence	Relevance	Consistency
Prompt_1	2.97	3.25	4.31	4.17
Prompt_2	3.05	3.17	4.49	4.52
Prompt_3	3.02	3.30	4.68	4.87

the prompt influence on text summarization, we also conduct different prompts on Arxiv 2025 datasets using LLaMA 3.1 8B. Specifically, the three prompts refer to:

- **Prompt 1** {Source Text} Please summarize the paper above in avg_sent_count sentences. Summary:
- **Prompt 2** {Source Text} Please summarize the paper above in avg_sent_count sentences, including Background, Motivation, Contribution and Results. Summary:
- **Prompt 3** {Source Text} Please generate an abstract for the paper provided above in {avg_sent_count} sentences according to the following guidelines:
  - The abstract must consist of exactly avg_sent_count complete sentences.
  - Without explicitly using the keywords Background, Motivation, Contribution, or Results, ensure that your abstract naturally includes:
  1. A brief description of the research context, highlighting the current challenges or gaps in the field.
  2. A clear explanation of the rationale for conducting the study.
  3. An overview of the innovative methods or ideas introduced.
  4. A summary of the key findings or conclusions reached.
  - Use formal, academic language and maintain a logical, coherent structure throughout the abstract. Your summary:

From Table 3 and Table 4, we can observe that the importing of structural guidance information and detailed instructions and constraints brings small improvements in the performance of semantic similarity and factual consistency.

In addition, for LLaMA series which apply top-p decoding, we also conduct experiment to compare the performance between Sampling Decoding and Greedy Decoding. From the perspective of similarity indicators, Greedy Decoding performs slightly better than Sampling Decoding in the tasks of news summarization and story summarization, as show in Table 5 and Table 6.

Table 5: LLaMA Performance between Sampling and Greedy Decoding (STORYSUMM)

Model	Decoding	R1	RL	BERTSc
LLaMA 2 7b	Default	33.69%	22.62%	87.48%
	Greedy	34.16%	24.03%	87.68%
LLaMA 2 13b	Default	26.74%	18.39%	84.20%
	Greedy	27.98%	18.75%	84.31%
LLaMA 3 8B	Default	39.95%	26.62%	88.49%
	Greedy	39.26%	26.58%	88.48%
LLaMA 3.1 8b	Default	39.95%	26.94%	88.40%
	Greedy	39.96%	25.82%	88.19%

Table 6: LLaMA Performance between Sampling and Greedy Decoding (News_annotation)

Model	Decoding	R1	RL	BERTSc
LLaMA 2 7b	Default	45.41%	31.83%	89.33%
	Greedy	44.88%	31.59%	89.14%
LLaMA 3 8B	Default	44.53%	31.75%	88.83%
	Greedy	46.54%	33.11%	89.30%
LLaMA 3.1 8b	Default	45.60%	31.48%	89.21%
	Greedy	46.64%	33.22%	89.42%

5 A Novel Self-Consistency Score for Text Summarization

Through the investigation of existing LLM techniques and metrics, although self-consistency is used to improve model accuracy and detect factual consistency, this concept has not been introduced into the evaluation system. With the case study based on our previous experiments, we observe that the non-overlapping content output by the multiple summarization system is often hallucination or irrelevant details, as shown in Figure 6.Therefore, to enhance the existing

evaluation framework, we propose a text self-consistency score pipeline by calculating the overlap between multiple generated text sfrom word level to semantic level.

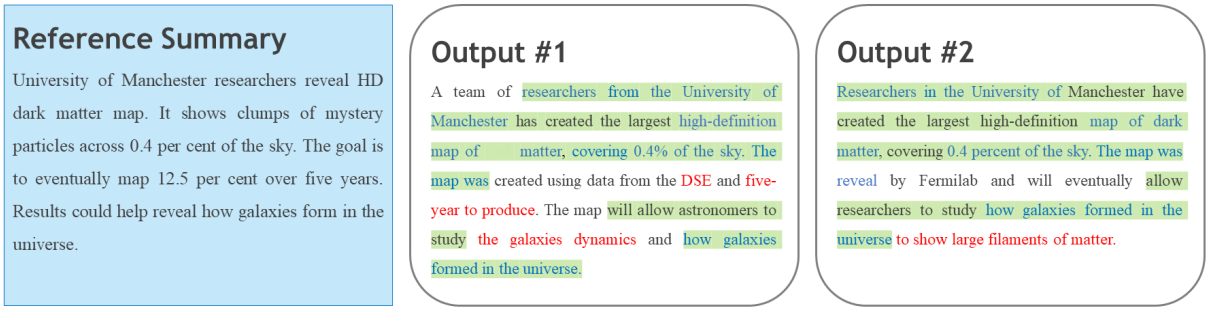

Figure 6: An example illustrates the overlap between generated summaries with the reference summary produced by LLaMA-2-7B-chat. Green Highlighted Texts: Overlap words between two generated summaries; Red Texts: Hallucination.

For a given source document $\vec{x}$, let $\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_n$ represent the outputs when we query the summarization model n times with the corresponding lengths as $l_1, l_2, \ldots, l_n$, and $\vec{y}$ as the reference summary. subsection 5.1 gives the basic definition of the consistency score, and subsection 5.2 further extends the definition to consider the word-level semantic equivalence.

5.1 Consistency Score based on Word Overlap

Although we have the intuition that consistency score should be defined with word overlap, we expect the defined consistency score should satisfy some properties that can be used in summarization systems. We expect the new score satisfies the following properties

Property 1 The score should be between 0 and 1.

Property 2 The score should be proportional to the overlap and inversely proportional to the summary length.

Property 3 With the same overlap and the same output length, the consistency score should be independent of the number of outputs.

The motivation of the first two properties is straightforward. The third property will guarantee the consistency score will not get better or worse by simply adding another output summary.

It turns out that a simple harmonic mean of the ratio between the number of overlapped words and the summary length will satisfy these three properties. Specifically, we define the consistency score as

$$r(\vec{s}_1, \ldots, \vec{s}_n) = \frac{n|\bigcap_{i=1}^{n} \vec{s}_i|}{\sum_{i=1}^{n} |\vec{s}_i|} \tag{8}$$

where $\bigcap_{i=1}^{n} \vec{s}_i$ is the word overlap among all outputs, and $|\vec{s}_i|$ is the length of the summary $\vec{s}_i$.

It is not difficult to verify the first and the second properties. For the third property, if $\vec{s}_{n+1}$ has the same length as the rest of the summary and $\bigcap_{i=1}^{n} \vec{s}_i = \bigcap_{i=1}^{n+1} \vec{s}_i$, then we will have $r(\vec{s}_1, \ldots, \vec{s}_n) = r(\vec{s}_1, \ldots, \vec{s}_n, \vec{s}_{n+1}) = \frac{|\bigcap_{i=1}^{n} \vec{s}_i|}{|\vec{s}_i|}$. Furthermore, if $\vec{s}_{n+1}$ is shorter but maintains the same overlap, we will have $r(\vec{s}_1, \ldots, \vec{s}_n) < r(\vec{s}_1, \ldots, \vec{s}_n, \vec{s}_{n+1})$, which reassures the second property.

Similar to the ROUGE scores, this definition can be easily extended to n-gram-based overlap, but we decided to stay on the unigram level in this mathematical formulation. But, unlike the ROUGE scores, this is symmetric because there is no reason to differentiate any of the outputs.

5.2 Extending Overlap with Word Similarity

Noticing the limitation of simple word overlap, we propose to extend the defined score in Equation 8 with word-level semantic information, particularly on counting the word overlap. For a given two texts $\vec{y}_1 = (y_{1,1}, \ldots, y_{1,l_1})$ and $\vec{y}_2 = (y_{2,1}, \ldots, y_{2,l_2})$, the idea is if two words $y_{1,k}$ and $y_{1,k'}$ are similar to each other, the algorithm will count them

as one overlap. In this project, we measured the similarity of two words based on the cosine similarity of their word embeddings

$$\text{cosine-similarity}(\vec{v}_{1,k}, \vec{v}_{1,k'}) > t \tag{9}$$

where $\vec{v}_{1,k}, \vec{v}_{1,k'}$ are the corresponding word embeddings, and t is a pre-defined threshold. This will give a more optimistic estimate of the consistency score than Equation 8, as shown in the experiments.

5.3 Implemention Details

The default setting of the tokenizer and word embedding used in the consistency score are shown in table Table 7, which is independent from the settings of LLMs. About the extended definition, we do not need to identify the word overlap on the fly when using Equation 8. In practice, a pre-processing of applying Equation 9 on the vocabulary and merging the word pair that passes the similarity threshold is more efficient.

In addition, a preliminary study on different thresholds shows different thresholds produced different consistency scores, but the ranks are the same, as shown in Appendix B. Therefore, in the following experiments, we use the value $t = 1$, which is equivalent to the original definition.

	Word Overlap	**Word Similarity**
n	2 / 3	
If_lower	True	
Stop words	Removed	
Tokenizer	`nltk.tokenize.word_tokenize`[1]	
Stemming	`nltk.stem.PorterStemmer`[2]	
Embedding	N/A	`bert-base-uncased`[3]
Threshold (t)	1	0.60 / 0.75 / 0.90

Table 7: Implemention Details of Word Overlap and Similarity for the consistency score

	CNN/DM					**XSum**				
Model	**R1**	**R2**	**RL**	**R-sum**	r	**R1**	**R2**	**RL**	**R-sum**	r
Gemma1.1-7B	40.75	14.59	26.77	36.74	0.7181	30.73	8.48	22.67	22.68	0.7293
Gemma-7B	40.99	14.91	26.81	34.24	0.6823	29.33	6.766	21.08	21.09	0.7221
Gemma2-9B	43.08	14.24	27.23	34.73	0.6562	35.09	11.10	25.88	25.907	0.6720
LLaMA-2-7B	43.09	16.07	27.69	35.26	0.6337	39.00	11.51	27.87	27.87	0.6551
LLaMA-2-13B	43.29	16.17	27.74	35.42	0.6428	40.06	12.53	28.85	28.86	0.6494
LLaMA-3-8B	48.33	18.87	31.10	39.58	0.6600	42.28	13.73	30.67	30.83	0.6120
LLaMA-3.1-8B	48.44	18.80	31.13	39.54	0.6280	42.53	13.56	30.76	30.80	0.6115

Table 8: Performance and Consistency Score Ratio (r) of Various Models on CNN/DM and XSum Datasets

6 Discussion

6.1 NLG Evaluation System

The evaluation of text generation tasks has always been an important issue in the NLP domain. Looking back at the development of evaluation technology, we can find that intuitive statistical indicators with high interpretability are difficult to measure from a semantic perspective, while indicators that can be applied to understanding ability and semantic level representation are weakly interpretable and unstable, making it difficult for us to truly locate the specific advantages and problems of the generated text. In addition, a complete text evaluation system needs to be evaluated from multiple dimensions. Each dimension contains indicators with highlights and limitations, which undoubtedly increases the complexity of the evaluation pipeline.

From our perspective, there are two promising directions worth exploring. The first direction is to select representative metrics from existing metrics that can complement each other for aggregation, such as the metric Texygen [51] proposed

[1] `https://www.nltk.org/api/nltk.tokenize.html`

[2] `https://www.nltk.org/api/nltk.stem.porter.html`

[3] `https://huggingface.co/bert-base-uncased`

in early work and the recent work FRANK [52], providing researchers with a relatively complex but comprehensive aggregation evaluation platform. The second direction leverages the powerful understanding and generation capabilities of current large language models (LLMs) [53] to measure semantic dimensions, emphasizing the reasoning process and enhancing the interpretability of evaluation.

6.2 LLM Performance and Reliability

In the analysis, we divide the causes of inconsistency into two categories. The first is irrelevant details, which refer to unnecessary information. Although not necessarily wrong, it will increase noise and reduce the quality and relevance of the text. The second is logical errors, which refer to logical incoherence errors in the content, which reflects the model's inadequacy in factual consistency. Both situations refer to consistency issue and have a negative impact on the reliability and coherence of the generated text. However, conceptually, improving consistency cannot eliminate hallucinations completely because the definition scope of the two is different. To completely eliminate hallucinations, the model needs to have many capabilities, such as alignment with source documents and reference summaries, coverage of world knowledge, self-consistency, etc. Therefore, we regard reducing hallucinations as a by-product of improving consistency rather than a direct goal of consistency, since solving hallucinations requires a more complex knowledge integration and verification process.

6.2.1 Randomness and Reliability of LLM System

Based on our analysis of the results, LLaMA/Gemma (7-13B) exhibits limited capability in following instructions. Even with prompt engineering and preprocessing to filter out non-overlap words, the consistency-guided strategy performs relatively limited on large-scale datasets since these models tend to fail to satisfy strictly to the requirement of using only overlapping words, with a certain level of randomness and hallucination. However, based on the statistical results of unigram tokens, we calculated that if the consistency-guided mechanism is strictly followed by LLMs, the model-generated summaries would have 2-3% improvement in F1 score. This provides a theoretical basis of consistency-guided feasibility to enhance model ability to balance summary length and information. We are considering more robust methods for aggregating multiple outputs in the future.

Similarly, we believe it is meaningful to continue exploring the randomness and reliability of LLM outputs for text summarization task, especially in terms of consistency. In future research, we plan to design an evaluation system that operates on various aspects, from word/unigram to sequence, semantics, and overall summary coherence. This work will be a starting point of our research to measure both performance and reliability, addressing the limitations of current evaluation metrics and helping users determine which models are comprehensively better for text summarization.

7 Conclusion

Currently, there exists a wide variety of text summarization evaluation metrics. To provide researchers with a comprehensive reference and facilitate the analysis of their limitations, we have systematically categorized and thoroughly investigated the most commonly used evaluation metrics in text summarization. Additionally, we conducted an extensive evaluation of the leading open-source large language models (LLMs), covering six different datasets and employing fifteen distinct evaluation metrics. Our study aims to offer valuable insights into the strengths and weaknesses of these metrics, helping to guide future research and improvements in summarization evaluation methodologies.

Moreover, to address the issue of inconsistency in summarization systems, we introduced a mathematical definition as the consistency score based on multiple outputs by measuring the overlap between summarization outputs. This method is quite straightforward, because it avoids the complexities of deep semantic understanding, introducing self-consistency score into current summarization systems evaluation framework.

A Appendix: Complete Experimental Results

Table 9: LLM Performance on Main Metrics (arXiv 2025)

Model	R1	RL	BLEU	GLEU	METEOR	BLEURT	BARTSc	BERTSc	FactCC	FactKB
LLaMA 2 7B	35.46	18.84	6.60	11.67	24.71	-0.51	-9.26	86.32	0.61	0.96
LLaMA 3 8B	45.58	23.83	9.16	13.41	29.95	-0.45	-5.04	85.00	0.58	0.97
LLaMA 3.1 8B	46.23	24.28	9.46	13.65	30.29	-0.35	-4.93	85.20	0.54	0.97
Gemma 7B	28.74	14.30	5.82	10.90	20.49	-0.68	-13.36	85.24	0.55	0.93
Gemma 1.1 7B	32.58	15.98	6.91	11.16	23.18	-0.54	-10.16	86.71	0.61	0.95
Gemma 2 9B	38.45	19.60	7.56	12.75	26.02	-0.48	-8.25	86.19	0.63	0.96

Table 10: LLM Performance on Main Metrics (arXiv 2018)

Model	R1	RL	BLEU	GLEU	METEOR	BLEURT	BARTSc	BERTSc	FactCC	FactKB
LLaMA 2 7B	26.01	22.35	5.49	11.76	21.90	-0.57	-13.88	84.01	0.46	0.89
LLaMA 3 8B	43.79	29.51	7.61	14.87	28.83	-0.49	-11.83	85.98	0.59	0.97
LLaMA 3.1 8B	42.93	28.38	7.34	12.34	27.56	-0.30	-11.95	85.99	0.67	0.97
Gemma 7B	31.87	19.01	4.39	9.48	20.62	-0.59	-12.84	85.67	0.57	0.94
Gemma 1.1 7B	33.94	23.77	5.26	9.97	21.30	-0.46	-9.73	86.52	0.63	0.96
Gemma 2 9B	36.59	28.59	1.77	10.95	22.50	-0.65	-7.91	86.89	0.60	0.95

Table 11: LLM Performance on Main Metrics (story_summ)

Model	R1	RL	BLEU	GLEU	METEOR	BLEURT	BARTSc	BERTSc	FactCC	FactKB
LLaMA 2 7B	33.67	22.60	2.93	8.56	17.62	-0.48	-11.23	87.43	0.49	0.75
LLaMA 3 8B	40.01	26.73	5.66	11.42	22.80	-0.37	-10.01	88.49	0.64	0.76
LLaMA 3.1 8B	39.07	26.33	5.35	11.69	22.87	-0.42	-10.31	88.49	0.62	0.76
Gemma 7B	32.47	21.69	2.75	7.46	16.58	-0.58	-13.07	87.52	0.44	0.74
Gemma 1.1 7B	34.21	22.97	3.04	8.77	16.70	-0.50	-12.35	88.46	0.49	0.74
Gemma 2 9B	39.45	26.49	5.17	11.35	23.21	-0.49	-9.79	88.59	0.57	0.76

Table 12: LLM Performance on Main Metrics (annotation_news)

Model	R1	RL	BLEU	GLEU	METEOR	BLEURT	BARTSc	BERTSc	FactCC	FactKB
LLaMA 2 7B	45.41	31.81	17.47	20.08	36.77	-0.42	-4.34	89.33	0.48	0.83
LLaMA 3 8B	44.53	31.76	15.82	18.69	38.77	-0.49	-4.23	88.83	0.60	0.82
LLaMA 3.1 8B	45.33	31.49	15.72	19.23	38.98	-0.36	-3.79	89.21	0.59	0.87
Gemma 7B	39.47	27.42	13.22	17.45	32.97	-0.58	-4.69	88.21	0.43	0.80
Gemma 1.1 7B	41.52	28.66	13.97	17.86	35.62	-0.53	-4.37	89.45	0.50	0.81
Gemma 2 9B	44.31	30.29	15.06	18.82	36.01	-0.48	-3.25	89.33	0.55	0.82

Table 13: LLM Performance on Main Metrics (CNN/DM)

Model	R1	RL	BLEU	GLEU	METEOR	BLEURT	BARTSc	BERTSc	FactCC	FactKB
LLaMA 2 7B	43.09	27.69	12.10	8.74	20.37	-0.47	-5.82	87.72	48.92	67.54
LLaMA 3 8B	46.33	31.10	15.66	11.26	21.60	-0.52	-5.39	87.33	58.31	-0.69
LLaMA 3.1 8B	48.41	31.13	15.04	13.49	23.08	-0.48	-5.14	87.89	56.24	67.12
Gemma 7B	40.99	26.81	10.02	7.05	15.46	-0.54	-6.02	87.10	45.17	69.40
Gemma 1.1 7B	40.75	26.77	9.82	7.29	16.01	-0.46	-5.67	87.64	51.79	68.00
Gemma 2 9B	43.08	27.23	11.43	9.16	19.97	-0.51	-5.46	87.17	59.14	78.60

Table 14: LLM Performance on Main Metrics (XSum)

Model	R1	RL	BLEU	GLEU	METEOR	BLEURT	BARTSc	BERTSc	FactCC	FactKB
LLaMA 2 7B	39.00	27.87	8.21	7.92	17.85	-0.55	-6.24	86.20	47.32	67.42
LLaMA 3 8B	42.28	30.67	9.46	10.13	19.02	-0.53	-6.07	87.49	48.01	65.51
LLaMA 3.1 8B	42.53	30.76	9.52	11.25	20.36	-0.49	-5.92	87.73	48.98	68.29
Gemma 7B	29.33	21.08	4.37	6.84	16.12	-0.58	-6.44	86.52	45.26	67.28
Gemma 1.1 7B	30.73	22.67	5.87	7.38	17.49	-0.56	-6.29	86.90	46.73	67.30
Gemma 2 9B	35.09	25.88	7.03	8.56	19.23	-0.50	-6.10	88.93	47.25	68.91

Table 15: LLM Performance on G-eval (arXiv 2025)

Model	Fluency	Coherence	Relevance	Consistency
LLaMA 2 7B	2.85	3.06	4.03	3.86
LLaMA 3 8B	2.94	3.33	4.25	3.91
LLaMA 3.1 8B	2.97	3.25	4.31	4.17
Gemma 7B	2.51	2.81	3.60	3.23
Gemma 1.1 7B	2.43	2.97	3.51	3.24
Gemma 2 9B	2.64	2.21	3.93	3.45

Table 16: LLM Performance on G-eval (arXiv 2018)

Model	Fluency	Coherence	Relevance	Consistency
LLaMA 2 7B	2.91	3.47	4.02	3.65
LLaMA 3 8B	3.12	3.68	3.89	3.94
LLaMA 3.1 8B	2.78	3.25	4.15	3.81
Gemma 7B	2.63	2.95	3.52	3.16
Gemma 1.1 7B	2.49	2.72	3.77	3.33
Gemma 2 9B	2.87	2.72	4.08	3.60

Table 17: LLM Performance on G-eval (story_summ)

Model	Fluency	Coherence	Relevance	Consistency
LLaMA 2 7B	3.62	4.21	3.89	3.45
LLaMA 3 8B	3.78	4.33	3.67	3.91
LLaMA 3.1 8B	3.54	4.02	3.83	3.70
Gemma 7B	3.29	3.95	3.58	3.68
Gemma 1.1 7B	3.41	3.87	4.05	3.52
Gemma 2 9B	3.73	4.10	4.06	3.78

Table 18: LLM Performance on G-eval (annotation_news)

Model	Fluency	Coherence	Relevance	Consistency
LLaMA 2 7B	3.44	4.66	3.94	3.38
LLaMA 3 8B	3.71	4.53	4.01	3.69
LLaMA 3.1 8B	3.78	4.79	4.09	3.68
Gemma 7B	2.94	3.81	3.65	3.47
Gemma 1.1 7B	2.98	3.90	3.88	3.54
Gemma 2 9B	3.42	4.28	4.10	4.17

Table 19: LLM Performance on G-eval (CNN/DM)

Model	Fluency	Coherence	Relevance	Consistency
LLaMA 2 7B	2.70	4.01	4.22	3.95
LLaMA 3 8B	2.78	4.05	4.36	4.21
LLaMA 3.1 8B	2.91	4.27	4.59	4.39
Gemma 7B	2.46	3.24	3.98	3.66
Gemma 1.1 7B	2.53	3.64	4.30	4.03
Gemma 2 9B	2.81	3.72	4.25	4.31

Table 20: LLM Performance on G-eval (XSum)

Model	Fluency	Coherence	Relevance	Consistency
LLaMA 2 7B	1.92	3.05	3.48	3.25
LLaMA 3 8B	2.05	3.18	3.72	3.51
LLaMA 3.1 8B	2.13	3.42	3.85	3.68
Gemma 7B	1.85	2.89	3.39	3.18
Gemma 1.1 7B	1.97	3.01	3.58	3.35
Gemma 2 9B	2.08	3.25	3.61	3.46

B Appendix: Consistency Scores regarding Different Thresholds

Table 21: Consistency scores of different datasets under varying thresholds on LLaMA-2-7B-chat

Threshold (t)	0.60	0.75	0.90	1(word-level)
CNN/DM	0.740	0.702	0.653	0.634
XSUM	0.781	0.719	0.675	0.655

Table 22: Consistency scores of different datasets under varying thresholds on Gemma-1.1-7B-it

Threshold (t)	0.60	0.75	0.90	1(word-level)
CNN/DM	0.762	0.753	0.730	0.718
XSUM	0.780	0.766	0.741	0.729

References

[1] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57, 2024.

[2] Tanya Goyal, Junyi Jessy Li, and Greg Durrett. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*, 2022.

[3] Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. FineSurE: Fine-grained summarization evaluation using LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 906–922, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[4] Liyan Tang, Igor Shalyminov, Amy Wong, Jon Burnsky, Jake Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. TofuEval: Evaluating hallucinations of LLMs on topic-focused dialogue summarization. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4455–4480, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[5] Yiming Wang, Zhuosheng Zhang, and Rui Wang. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. In *Annual Meeting of the Association for Computational Linguistics*, 2023.

[6] Griffin Adams, Alex Fabbri, Faisal Ladhak, Eric Lehman, and Noémie Elhadad. From sparse to dense: GPT-4 summarization with chain of density prompting. In Yue Dong, Wen Xiao, Lu Wang, Fei Liu, and Giuseppe Carenini, editors, *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 68–74, Singapore, December 2023. Association for Computational Linguistics.

[7] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa

Dataset	**Llama 3 8B**	**Llama 3.1 8B**	**Gemma 2 9B**
CNN/DM	0.42/0.35/0.73/0.65	0.45/0.37/0.70/0.68	0.38/0.33/0.77/0.62
XSum	0.35/0.30/0.69/0.60	0.37/0.32/0.67/0.63	0.33/0.29/0.70/0.58
Annotation-News	0.48/0.40/0.65/0.70	0.50/0.42/0.64/0.72	0.45/0.42/0.67/0.67
ArXiv 2018	0.39/0.32/0.72/0.62	0.41/0.34/0.62/0.65	0.37/0.31/0.73/0.60
ArXiv 2025	0.36/0.28/0.55/0.58	0.38/0.30/0.58/0.56	0.35/0.29/0.59/0.60
STORYSUMM	0.44/0.36/0.68/0.67	0.46/0.38/0.68/0.70	0.41/0.35/0.69/0.64

Table 23: Average scores: Lexical/Semantic/Faithful/LLM-eval. General Performance of Zero-shot LLMs

Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[8] Sangwon Ryu, Heejin Do, Yunsu Kim, Gary Geunbae Lee, and Jungseul Ok. Key-element-informed sllm tuning for document summarization, 2024.

[9] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[11] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[12] Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

[13] Philippe Laban, Arman Cohan, Krysta Svore, and Daniel Weld. Summac: Re-visiting nli-based models for inconsistency detection in summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1439–1453, 2021.

[14] Alexander R Fabbri, Wojciech Kryściński Zhang, Liqun Wei, Dragomir Radev, and Kathleen McKeown. Qafacteval: Improved qa-based factual consistency evaluation for summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5892–5909, 2022.

[15] Chunting Liu et al. Gpteval: Nlg evaluation using gpt. In *ArXiv preprint*, 2023.

[16] Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: Evaluating generated text as text generation. *ArXiv*, abs/2106.11520, 2021.

[17] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[18] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[19] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[20] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[21] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[22] Yixin Liu and Pengfei Liu. Simcls: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1065–1078, 2021.

[23] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[25] Ziqiang Cao, Yang Gao, and Wenjie Li. Consum: Conceptual summarization via guided generative models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3790–3802, 2022.

[26] Mathieu Ravaut, Hailin Chen, Ruochen Zhao, Chengwei Qin, Shafiq R. Joty, and Nancy F. Chen. Promptsum: Parameter-efficient controllable abstractive summarization. *ArXiv*, abs/2308.03117, 2023.

[27] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Huai hsin Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171, 2022.

[28] Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation, 2023.

[29] Taiji Li, Zhi Li, and Yin Zhang. Improving faithfulness of large language models in summarization via sliding generation and self-consistency. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8804–8817, Torino, Italia, May 2024. ELRA and ICCL.

[30] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore, 2023. Association for Computational Linguistics.

[31] Andrew Mutton, Mark Dras, Stephen Wan, and Robert Dale. GLEU: Automatic evaluation of sentence-level fluency. In Annie Zaenen and Antal van den Bosch, editors, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 344–351, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[32] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss, editors, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[33] Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. Bleurt: Learning robust metrics for text generation. In *ACL*, 2020.

[34] Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online, November 2020. Association for Computational Linguistics.

[35] Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. FactKB: Generalizable factuality evaluation using language models enhanced with factual knowledge. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 933–952, Singapore, December 2023. Association for Computational Linguistics.

[36] Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Answers unite! unsupervised metrics for reinforced summarization models. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China, November 2019. Association for Computational Linguistics.

[37] Thomas Scialom et al. Questeval: Summarization asks for fact-based evaluation. In *EMNLP*, 2021.

[38] Esin Durmus, He He, and Mona Diab. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online, July 2020. Association for Computational Linguistics.

[39] Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Dreyer Markus, and Mohit Bansal. Factgraph: Evaluating factuality in summarization with semantic graph representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.

[40] Yang Gao, Wei Zhao, and Steffen Eger. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354, Online, July 2020. Association for Computational Linguistics.

[41] Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. Fill in the BLANC: Human-free quality estimation of document summaries. In Steffen Eger, Yang Gao, Maxime Peyrard, Wei Zhao, and Eduard Hovy, editors, *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20, Online, November 2020. Association for Computational Linguistics.

[42] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*, 2023.

[43] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, December 2023. Association for Computational Linguistics.

[44] Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. Towards a unified multi-dimensional evaluator for text generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[45] Liyan Tang, Philippe Laban, and Greg Durrett. Minicheck: Efficient fact-checking of llms on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2024.

[46] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, 2016.

[47] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL workshop*, 2004.

[48] Melanie Subbiah, Faisal Ladhak, Akankshya Mishra, Griffin Adams, Lydia B. Chilton, and Kathleen McKeown. Storysumm: Evaluating faithfulness in story summarization, 2024.

[49] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[50] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[51] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Junghyun Lee, and Yong Yang. Texygen: A benchmarking platform for text generation models. *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100, 2018.

[52] Artidoro Pagnoni, Kevin Liu, and Yi-Lin Tuan. Frank: Factuality evaluation of summaries with robust annotation and knowledge. *arXiv preprint arXiv:2104.13346*, 2021.

[53] Wayne Xin Zhao, Kun Zhou, Jun Li, Ming Tang, Xiang Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhu, and Ji-Rong Wen. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.