University of Trento

Centro Interdipartimentale Mente/Cervello

Master's in Cognitive Science

~ · ~

Academic Year 2023–2024

Evaluation Metrics for Retrieval Augmented Generation in the Scientific Domain

University of Trento Supervisor
Prof. Stefano TESO

University of Basque Country Supervisor
Prof. Rodrigo Agerri

Company Supervisor (Iris.AI)
Ivan Vankov

Graduate Student
María Paz OLIVA
247055

Final examination date: October 9th, 2024

This work was co-funded by the Erasmus Mundus Masters Program
in Language and Communication Technologies, EU grant no. 2019-1508.[1]
This work was co-funded by the company Iris.AI,
Machine Learning Thesis internship program.[2]
This work was co-supervised by the University of Trento,
the University of Basque Country and Iris.AI.

With the support of the Erasmus+ Programme of the European Union

[1] `https://lct-master.org/`
[2] `https://iris.ai/`

To Agus, for being by my side every step of the way.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to **Ivan Vankov**, my supervisor from the Iris.AI research team. His daily guidance, insightful feedback, and unwavering support were instrumental in shaping this work. I am also grateful to the entire Iris.AI team for providing me with an incredible learning experience that bridged the gap between research and industry. Special thanks to **Adriana Correira** for her support and for reading this work so carefully, and to **Victor Botev** for giving me this opportunity. I hope my work has been valuable and will contribute meaningfully to the team's efforts.

I would also like to extend my gratitude to my supervisor from the University of Trento, **Stefano Teso**, for his time and valuable feedback throughout this journey.

I am especially thankful to my LCT friends from San Sebastian — **Noumi, Luigi, Anya, Nastya, Toni, Jehad, Ravi and Nati**. Your incredible talent and skills in the field of NLP, along with your generosity, were crucial during the moments when everything was new to me. Without your support, I wouldn't have come this far. Thank you all.

I also want to thank my friends and family back in **Argentina**, who cheered me on from across the ocean. I missed you all, but this journey was worth it. A special mention goes to the **University of Buenos Aires**; as without the excellent and high-quality education I received there, I would not have been so well-prepared for this challenge.

Lastly, I want to express my heartfelt thanks to three remarkable individuals:

To **Dr. Eduardo Mazzaro**, for his exceptional professionalism and compassion in treating me, enabling me to continue with this program. To **Dr. Daniel M. Low**, who opened this path for me and constantly encouraged me to strive for more — I will forever be grateful for your generosity. And to **Agustin Mc Cargo**, for standing by me every step of the way, especially during the toughest moments.

Abstract

Retrieval-Augmented Generation represents the state-of-the-art approach to perform question-answering (QA) tasks in the scientific domain. This system combines a powerful generative component, capable of producing grammatically sound and readable answers, with a retrieval component that efficiently locates specific information within a large corpus of documents. As such, RAG systems are particularly well-suited to address the complexities inherent in this task. However, evaluating the accuracy and quality of the generated answers remains a significant challenge.

The aim of this thesis was to find an effective method for assessing RAG performance in a scientific QA task. To this end, we conducted an extensive review of the current automatic evaluation metrics in use. The most common approach involves comparing generated answers with a reference produced by humans. Such comparison can focus on the form (lexical similarity), the content (semantic similarity), or on a deeper analysis through the use of Large Language Models (model-based). Each of these approaches has well-documented advantages and drawbacks, making it necessary to rigorously test their reliability and effectiveness in this context.

To explore this, we selected representative metrics from each category and designed three progressively complex experiments to challenge them and analyze their behavior: 1) Can the metrics distinguish between correct and incorrect answers? 2) Can they differentiate between answers of varying quality, particularly when they deviate in form or content from the reference? 3) Do the metrics align with human preferences?

The strengths and limitations of these metrics were empirically examined. Findings showed that at the most basic level — distinguishing clearly correct from incorrect answers — all metrics had a good performance to varying degrees. However, when they faced more nuanced challenges, as to differentiate between variations in form or content when comparing higher and lower-quality answers, both lexical and semantic similarity metrics struggled. Therefore, model-based metrics demonstrated greater flexibility and reliability. Nevertheless, in the final experiment, none of the evaluation methods—across all categories—aligned consistently with human judgment. In fact, most of the metrics exhibited divergence from human preferences.

Consequently, no metric met performance expectations in all scenarios. Nonetheless, we were able to provide a comprehensive analysis of their behavior, strengths, and limitations. As a conclusion, we propose that for assessing the performance of a RAG system in scientific QA model-based metrics appear to be the most effective, particularly in distinguishing correct from incorrect answers and in differentiating varying levels of answer quality. However, further research is needed to better align these metrics with human judgment. Moreover, findings suggest that relying solely on human-generated reference answers as benchmarks may not effectively capture human preferences. Instead, future evaluation frameworks could integrate human preferences directly into the evaluation process.

By shedding light on the performance of current evaluation methods and advocating for a shift toward model-based metrics that better incorporate human preferences, this thesis aims to contribute to the field of QA evaluation and guide future research towards developing more reliable and robust evaluation frameworks.

Contents

Introduction

In recent years, humanity has witnessed the rise of Artificial Intelligence (AI), which encompasses technologies that simulate human intelligence in machines. A significant branch of AI is Natural Language Processing (NLP), which enables machines to understand and produce human language. Within NLP, Natural Language Generation (NLG) is dedicated to generating human-like text by utilizing advanced algorithms trained on extensive data. NLG encompasses many tasks, including accurately answering to questions posed in natural language, commonly known as question answering (QA).

To perform QA tasks, specialized models designed to automatically respond to questions posed in natural language are required, known as **QA systems** [5] [65] [14]. Traditional QA systems typically rely on information retrieval techniques, which involve extracting relevant answers directly from pre-existing texts or databases. These systems use predefined algorithms to locate and present the most relevant piece of text that answers the user's question, often resulting in exact or slightly modified excerpts from the source material.

Furthermore, with the advent of more powerful NLG architectures, such as Large Language Models (LLMs), there has been a transition from traditional QA to generative QA [50] [45] [39]. This marked a significant evolution in how machines handle and respond to queries. Generative QA (genQA) leverages advanced models to generate answers from scratch, by synthesizing responses based on their understanding of the language and context learned from vast amounts of data. However, despite their sophistication, they present limitations: first, LLMs can't answer questions about events which were not included (or occurred after training); second, LLMs cannot cite their sources; third, LLMs are prone to generating "hallucinations", fabricated information that can appear credible. These challenges underscore the need for continued advancements and hybrid approaches [31] [66] [67].

To address these issues, **Retrieval-Augmented Generation (RAG)** emerged [53]. RAG is one of the latest advancements in the field, as it combines generative models with information retrieval techniques to enhance the accuracy and relevance of answers. This approach first retrieves relevant passages and then employs generative models to refine or generate answers based on the retrieved information (further details on Section 0.1).

In order to ensure the quality of the text generated by NLG models for their usefulness and safety in different applications, **evaluation** has been fundamental [36] [34]. Reliable evaluation methods are essential, as they help to identify and improve weakness in models, thereby optimizing their development, and prevent dangers to users, avoiding the generation of counterfactual or harmful information. Evaluating generated text involves numerous approaches and can vary significantly based on several factors, including the task being performed, the system being evaluated, and the specific aspects under scrutiny.

At a basic level, one can assess **readability** (fluence and coherence for readers' comprehension) and **reliability or accuracy** (providing outputs that accurately address the required task). Particularly, there are various ways to assess whether a generated answer to a question is correct. For instance, one can evaluate specific criteria such as *relevance* (the extent to which the answer directly addresses the question), *factual consistency* (the accuracy of the information in the answer based on the provided context), and *completeness* (whether the answer fully addresses all aspects of the question). In a more in-depth analysis, one can analyze **truthfulness** (maintaining factual consistency and preventing hallucinations), **robustness** (exhibiting consistent behavior across different inputs and scenarios) and **safety and biases** (avoiding the generation of harmful, violent, or biased content) [59] [87].

However, evaluating the accuracy of generated answers on a QA task is non-trivial. It is very challenging as there is an inherent difficulty in objectively determining if a given answer is correct [40] [87] [10] [81] and, in human language, the same content can be expressed in countless ways. Consequently, the evaluation process depends heavily on the type of questions and answers involved, as well as the QA system used.

QA tasks can be classified based on several dimensions, chiefly the type of questions and answers involved, their form, content, and choice of topic. Regarding their form, they can be short (multiple-choice, yes-no, a couple of words) or long-form (a couple of sentences). Regarding their content, they can address straightforward questions with specific facts or handle complex questions requiring reasoning and explanations. Additionally, they can pertain to an open domain (any topic) or a closed domain (specific area). Finally, they can rely on verifiable and objective data (fact-based) or in subjective opinions. Concerning the method of obtaining answers, they can be extracted (directly from context) or generated (by the system). A common categorization distinguishes between closed-book answers, based solely on pre-existing knowledge embedded in the model, and open-book answers, which rely on external knowledge sources that the system accesses during the answering process. Evidently, simpler QA forms, such as yes-no or multiple choice, are easier to assess since it is more straightforward to determine whether the model produced the correct answer through exact matching. However, with more complex open-domain questions and long-form answers, the difficulty increases [50] [16].

One of the most challenging QA scenarios is **scientific QA**, which involves answering questions based on scientific papers. This particular task has ample applications, both in the industry and the academia, which leads to a significant incentive to provide high quality performance. This task is particularly difficult because the scientific domain requires very specific knowledge and terminology, and it constantly evolves. As a result, not every model can effectively handle it. RAG systems, in particular, represent the state-of-the-art (SOTA) in QA systems for this task because they possess the ability to focus on the content of the papers [67]. Scientific QA with RAG systems ensures that the evaluation is thorough, as it pushes the model to really focus on the retrieved context rather than relying solely on its pre-existing general knowledge.

Evaluating the accuracy of RAG systems in answering questions based on scientific text involves retrieving the correct pieces of information from the corpus and composing the appropriate answer. Assuming that the retrieved information is both accurate and sufficient, the key challenge lies in assessing whether the generated text is a relevant response that faithfully reflects the information contained within the context. Given the difficulties of assessing "correctness" in answers, it is generally accepted that a good answer is one that aligns most closely with human expectations and preferences. There is consensus that human experts are the ultimate authority in defining any quality aspect of generated text, including the correctness of an answer [39] [66] [38] [16]. Consequently, human evaluation methods, where human judges assess the quality of the generated text based on predefined criteria, are considered optimal. In QA task, this would mean to assess if the answer is acceptable to expert human annotators.

Unfortunately, these evaluation processes are highly time-consuming and resource-intensive. As a result, many alternative **automatic evaluation metrics** have been developed over the years [40] [16] [10] [85]. The most popular approach involves assuming that humans know the correct answer and, therefore, using human answers as a reference ground truth and analyzing the similarity of generated responses to these. This comparison can be based on superficial form (*lexical similarity*), content (*semantic similarity*), or a more *complex analysis* (using advanced models and considering the context). Vast and diverse automatic evaluation metrics have been developed since the early 2000s (see Section 0.2). They are based on varied methodologies with specific characteristics that have become increasingly complex and sophisticated over time, as new technologies have been developed. As the pace of developments accelerates daily and the number of related papers is expanding exponentially, we conducted a comprehensive and extensive literature review (see Section 0.2), finding that each approach encompasses its own documented advantages and limitations. However, there is no agreement on a single standardized evaluation methodology. What is more, different and new projects try to use a variety of this already questioned metrics (see Section 0.3). This leads to a necessity to critically reflect on and validate these evaluation methods [49] [101] [7] [10].

Therefore, the **objective** of this thesis is to *analyze the performance of various evaluation metrics from diverse approaches to empirically compare their behaviors when evaluating RAG QA systems on a scientific QA task.* We will focus on measuring if the produced text is the right answer to the provided question considering the context, and we will assume that the retrieved information is accurate and sufficient. That means we will work on an artificially ideal scenario, where there is only one chunk of context retrieved, and it is the correct one, so only the generative component will be under evaluation.

The research work will be an exploratory analysis through a series of experiments with increasing complexity to test these metrics and shed light on their effectiveness in this context. The initial inquiry focuses on a fundamental basic question: *1. Can the metrics distinguish between correct and incorrect answers?* Building on this, we further investigate the metrics sensitivity on a more challenging scenario: *2. Can the metrics differentiate between "better" and "worse" answers?* We examine the sensitivity of these metrics to varying degrees of correctness and explore whether they can be misled by incorrect answers that appear correct. Finally, to completely understand the utility of the metrics, we address a key question: *3. How well do these metrics align with human preferences?* These are the core **research questions** guiding our experiments.

This thorough review of evaluation metrics is necessary for the field of long from QA. First, to make it easier and faster for researchers and practitioners to understand the possibilities and make informed decisions about evaluation methods. But also, having more standardized procedures could provide a common foundation for comparing different models and approaches, promoting transparency and fostering a collaborative environment where advancements can be more effectively shared and built upon. The novelty of this work lies in its focus on the specific task of long-form QA in the scientific domain using RAG systems. Additionally, this thesis aims not only to provide a comprehensive description of evaluation metrics, from the oldest to the most recent, but also to implement, test, and compare their behaviors in practice.

In the remainder of this introduction: 1. we will discuss the specific features of RAG QA systems and their generative component (LLMs); 2. we will review the existing evaluation metrics from various approaches, along with a critical documented overview of these metrics; 3. finally, we will provide a literature review of the work done in this area to date, highlighting the key challenges and identifying areas where future research is needed.

0.1 RAG system

The field of NLG has produced numerous models for generating human-like text. Initially, it focused on statistical and rule-based algorithms (like n-gram models and Hidden Markov Models[74]), which used counting and probability estimation from data to predict the next word. However, these methods had limitations in context understanding, data sparsity, and handling ambiguity. The introduction of Neural Networks (NNs) marked significant progress, offering greater flexibility and accuracy by processing input through multiple layers and adjusting weights to minimize prediction error. Recurrent Neural Networks (RNNs) [61] specifically handled sequential data and learned long-term dependencies. Finally, the advent of attention mechanisms further improved performance by helping models focus on relevant parts of the input sequence.

Currently, Transformers [92], which use self-attention mechanisms and positional encodings to weigh the importance of different words in a sequence and maintain the order, are the SOTA in NLG, enabling the development of highly coherent and contextually relevant text generation systems like Large Language Models (LLMs)[99] [63]. Based on Transformer architecture and trained on vast data, LLMs demonstrate unprecedented text generation capabilities. However, they present limitations as they cannot answer questions about events which were not included (or occurred after) training, can not cite their sources, and are prone to generating hallucinations. This motivated the development of Retrieval-Augmented Generation (RAG) systems, a hybrid architecture approach that combines a retrieval and a generative component [37] [33].

Generative component (LLM): Text generation involves creating coherent, contextually relevant text based on a given input. A generative model is a machine learning model that generates new data samples resembling the training data, capturing its underlying distribution and characteristics. Language modeling involves building a model that predicts the next word or sequence in a sentence based on the context provided by preceding words. Among the latest and most advanced generative models are LLMs, which can understand and generate human-like text. These models are trained on vast, diverse text data and use complex neural networks to predict the next token in sentences based on input. Traditional LLMs produce appropriate responses to a wide array of queries, effectively handling QA tasks. However, LLMs rely on finite training data, leading to challenges like outdated information and hallucinations [59] [36]. RAG systems address these issues by integrating external knowledge into the generation process, supplementing LLMs' responses with up-to-date, domain-specific information [64].

Retrieval component: Information retrieval refers to the process of searching for and extracting relevant information from a large text corpus. This process involves sophisticated indexing and querying mechanisms, which transform raw data into usable chunks of information, stored in vectorized form, and optimized for quick access and high relevance. A retrieval model is designed to identify and rank documents or text passages that are most relevant to a query. This is typically achieved through techniques such as keyword matching, semantic search, and the use of embeddings to capture the meaning of text. Embeddings are numerical vectors in a continuous space typically generated by machine learning models trained on large datasets, that map text units to vectors, with each dimension capturing a specific feature of the text. Advanced retrieval may use vector representations of text, enabling more nuanced and precise matching.

In conclusion, given a query or input, the retrieval component searches for a set of relevant documents or pieces of information from a pre-existing corpus. These are then used as additional context by the generative component to produce a coherent and contextually relevant response to the original query.

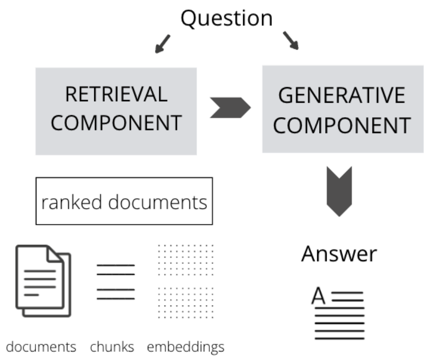

Figure 1: Simplified overview of RAG pipeline workflow.

For all this, RAG systems are particularly well-suited for scientific domains, where the integration of up-to-date knowledge is crucial [67]. Scientific QA tasks require not only accurate retrieval of domain-specific information, but also the ability to reason over complex content. Therefore, applying RAG to scientific document reasoning can significantly enhance the interpretability and verification of model predictions by grounding them in relevant literature. However, challenges remain in ensuring the relevance and accuracy of the retrieved documents, especially given the vast and diverse nature of scientific data. The risk of evidence fabrication—where the system generates plausible but incorrect justifications—poses a significant concern. Furthermore, the retrieval component may return irrelevant or non-contextual information, which can mislead the generative model and degrade output quality. The reliance on external data sources also introduces challenges related to their quality and reliability [33] [32].

To mitigate these risks and ensure the reliable deployment of RAG systems in scientific applications, ongoing evaluation and refinement are critical. However, as will be discussed, the evaluation of RAG systems poses significant challenges and there is no consensus on the optimal evaluation methodology. This underscores the need for a thorough reassessment of current evaluation frameworks.

0.2 Automatic evaluation metrics

A metric is a standardized measure used to quantitatively evaluate a specific aspect of an output. An effective metric should be *quantifiable* (expressed in numerical terms or through a ranking system, enabling clear and objective comparisons), *reliable* (produces stable and repeatable results under consistent conditions) and *accurate* (correctly measures what it is intended to by aligning closely with human judgment) [40] [85].

In this thesis, we focus on metrics designed to assess long-form text, where two texts, a reference and a candidate, are compared. Since the early 2000s, a wide range of these metrics has been developed, employing diverse methodologies and approaches. However, there is no consensus on a standardized system for classifying and organizing these metrics based on shared characteristics. To address this, we propose a taxonomy informed by an extensive literature review in the field. We categorize the metrics into three groups based on the focus of the evaluation: comparing superficial form (*lexical similarity*), comparing content (*semantic similarity*), or employing a more complex reasoning (*based on LLMs*). Following, each group will be explained in detail, accompanied by a review of relevant literature highlighting their advantages and limitations.

- **Lexical similarity:**

These metrics assess lexical similarity between the generated text and a reference ground truth by measuring the co-occurrence of sequences of "n" consecutive items (n-grams) at varying levels of granularity, such as characters, words, specific sequences, full strings, or patterns. Commonly referred to as traditional metrics due to their long-standing use and algorithmic simplicity, they are either rule-based or statistical, as they rely on predefined algorithms or probabilistic models to quantify similarity based on exact matches or statistically identified patterns. Examples of this type of metrics are **BLEU[69], ROUGE[55], METEOR[8], ChrF[72] and TER[82].**

Traditional metrics have been widely used for over two decades, proving useful and reliable for tasks like machine translation, where the goal is to achieve the closest match to the reference. During their development, efforts were made to gradually overcome their restrictiveness, allowing for more flexible considerations of matching. However, it is intuitive to recognize that for more complex tasks these traditional metrics are not optimal.

Key research projects have empirically reviewed the strengths and limitations of these metrics [68] [57]. These studies, evaluating ROUGE, BLEU, METEOR, and TER, concluded in general that: 1. these metrics exhibit low correlation with human judgment; 2. they display a length bias, unfairly penalizing added supportive tokens; and 3. they assign low scores to answers that are lexically different from the reference, even if the content is the same (example: using synonyms), due to a lack of semantic understanding. Future research, as we will demonstrate, also compares their behavior with newer metrics, showing they are consistently outperformed.

- **Semantic similarity:**

Given the primary disadvantage of traditional metrics—lack of semantic understanding—there has been a shift towards semantic similarity metrics. These metrics assess the similarity between generated text and the reference by analyzing their semantic equivalence. To do this, both texts are represented with embeddings, as they effectively encode the semantic meaning, with similar meanings resulting in vectors that are close to each other in the vector space. The degree of similarity or dissimilarity between texts is then measured by calculating the distance between these embeddings using mathematical distance metrics. Semantic similarity metrics differ in their approach by utilizing embeddings at various levels of granularity: static word embeddings (fixed representation of a word regardless of the context in which it appears, like the popular Word2Vec[62] and GloVe[70]); static sentence embeddings (aggregate word embeddings to represent entire sentences as single vectors, such as Sentence-Bert); and contextual embeddings (capture the meaning of the word or sentence within the context in which they appear, typically generated by LLMs). They also employ different methods for calculating distances, such as Cosine Similarity or Euclidean Distance. Notable examples include **Mover's Distance[22][47], BERTScore[98], and WISDM[11].** Despite their strength in capturing semantic content, these metrics still have limitations. A key issue is that a single vector may represent multiple meanings, leading to less accurate representations as the complexity of meanings increases. In the literature, for instance, the QAEv study [17] demonstrates that for more complex generative QA tasks, where context comprehension and factual accuracy are crucial, semantic similarity metrics do not significantly outperform traditional ones. Similarly, Sellam et al. (2020) [80] highlights the shortcomings of embedding-based methods in capturing factual correctness and coherence. Additionally, both Celikyilmaz et al. (2020) [15] and Yeh et al. (2021) [96] studies reveal that these metrics struggle with long-form text, primarily due to computational complexity and the loss of sentence relations. These findings suggest the need for metrics that can better handle challenging responses and deeper contextual understanding, incorporating both the question and the context into the evaluation.

- **LLM based:**

Finally, with the advent and widespread adoption of sophisticated model architectures, evaluation metrics have also evolved to leverage these new technologies and address emerging challenges. Mainly, they move beyond assessing lexical similarity through superficial matching or semantic similarity through embeddings. Instead, these powerful models, trained on vast datasets, learn patterns that account for both the original question and the context when evaluating. Currently, these metrics are typically based on Transformer architectures, whether open-source or proprietary, such as BERT[26], LLAMA [91], and GPT [75] models.

These metrics can be divided into two groups: "*learned*" and "*prompt-based*". Learned metrics involve models specifically trained or fine-tuned to evaluate the quality of generated text, utilizing either widely available QA datasets or custom datasets designed for specific needs. Examples include **BLEURT[79], BEM[13], GPT-Judge[56], and Prometheus[46]**. On the other hand, prompt-based metrics use generative models that respond to direct or multi-prompt inputs to compute a score, such as **RAGAS[29]**. Furthermore, prompt-based metrics can be further classified into three types: metrics where the generated answer, the question, and the reference are provided as inputs, and the model is prompted to directly output a score based on specific evaluation criteria (scoring rubric); probabilistic metrics that calculate the likelihood or probability of the output text being correct or appropriate according to a model; and, finally, metrics that simulate pairwise comparisons (human preference) and compute the win rate of a model's answers when compared with answers generated by other models or written by humans.

Research indicates that metrics based on Transformer models often surpass traditional evaluation methodologies, demonstrating a higher correlation with human judgment [100] [27] [60] [94] [83]. However, despite their advantages, research also highlights several significant challenges associated with these metrics [41]. One significant issue is that metrics based on pre-trained models and specialized datasets often inherit the biases present in those models, and they run the risk of having been exposed to certain data during training that they are intended to evaluate afterwards [88] [94]. In addition, these metrics exhibit **scoring biases**. For example, there is a *length bias*, where shorter text is frequently favored. They also tend to prioritize responses with *superficial qualities*—such as verbosity or formality—over the actual quality of the content. This raises concerns about factual inconsistency, as some studies warn that these metrics may assign high scores to linguistically polished but factually incorrect answers [20]. Another issue is *low granularity*, where scores are often unreasonably uniform, failing to capture nuanced differences [19] [86] [54]. Furthermore, there is a bias stemming from *model familiarity,* where responses generated by similar models are preferred. Due to these issues, LLM-based metrics have been shown to lack robustness, with *low agreement* observed across different datasets and tasks. This inconsistency may be attributed to their high sensitivity to *variations in prompts and data*, which has been well-documented [9] [93]. In conclusion, there is a pressing need for empirical reviews and meta-evaluations to ensure the reliability and accuracy of these evaluation metrics [6] [49] [18].

Overall, there are numerous metrics designed to measure various aspects of performance in different ways. Understanding the strengths and weaknesses of each methodology is essential to selecting those that best fit the specific needs of a project. Therefore, to choose the most appropriate method for assessing the performance of a RAG system in answering scientific questions based on papers, it is crucial to have a deep understanding of these metrics. With that in mind, we will now present a review of the relevant literature on the topic.

0.3 Related Work

- **Surveys on evaluation metrics and evaluation malpractices:**

As the various evaluation paradigms exhibit discrepancies that significantly impact the final evaluation results, systematic reviews are imperative. These should be both comprehensive, encompassing the widest range of metrics and the latest methodologies; and empirical, involving experiments to test and validate these metrics, to effectively guide developers and researchers.

To the best of our knowledge, there is no empirical metrics evaluation survey specifically assessing long-form QA in the scientific domain using a RAG system. Although good surveys on LLM evaluation methodologies exist [59] [36] [16] [101] [64] [49], they either focus on LLM evaluation in general without deeply addressing QA tasks, or they remain theoretical without empirical implementation. Some works, such as [87] [43] [3] [24] effectively assess the QA task. However, they have certain limitations: they consider only short-form answers, do not specifically address the scientific domain, and/or omit some of the newest metrics.

Nevertheless, their key takeaways are significant. These works, along with others such as [87] [7] [10], which also reflect on the evaluation process, malpractices and best practices, yield some overall conclusions: 1. evaluating long-form generated text is undoubtedly challenging, necessitating a review of current metrics and their reliability, likely requiring a multiple metrics approach; 2. a key problem in assessing generated text is that the same content can be expressed in various forms, highlighting the intrinsic disadvantage of reference-based metrics that rely on a gold standard answer; 3. this leads to the problematic necessity of incorporating costly human judgement into the evaluation; 4. finally, there is a need for unified and standardized evaluation protocols in the field, addressing concerns such as the lack of clear and detailed reports on evaluation setups for reproducibility and the lack of access to raw model outputs.

- **RAG QA system evaluation: the gap on evaluation practices:**

Initial efforts to augment QA systems by incorporating various retrieval methodologies [50] [37] [45], utilized Exact Match (EM) as the evaluation metric. EM requires the predicted answer to precisely match the ground truth answer, including all words in their exact order, making it a stringent and precise measure of a system's accuracy in retrieving correct answers. Similarly, the paper introducing the RAG architecture [53] employed this metric and also included traditional ones like BLEU and ROUGE. While these offer more flexibility, they still depend on superficial lexical matching. Subsequent research on generative QA [39] [31] [66] continued to use them, with some incorporating semantic similarity metrics such as BERTScore and BLEURT.

More recent studies on specific RAG QA systems [42] [48] have predominantly relied on computing accuracy and F1 scores based on the restrictive EM, using data from various benchmarks and datasets. Some works [35] propose a combination of metrics from different approaches, integrating both traditional metrics (EM, ROUGE, BLEU) and semantic ones (BERTScore). Additionally, Aksoy et al. (2024) [4] introduces the novel "answer correctness" metric from RAGAS [29], which combines semantic similarity with a prompt-based methodology based on LLMs. Lastly, RAG QA Arena [38] represents a significant and necessary attempt to assess RAG QA systems using proper human-written long-form answers and a pairwise preference approach conducted by both humans and LLMs.

Overall, this literature review reveals that despite the well-documented disadvantages, most work still relies on similarity metrics. More importantly, there is no unified criterion for evaluation, neither for the metrics nor for the data, which is understandable given the rapid pace of developments. This inconsistency highlights a significant gap in current QA system evaluation practices and underscores the urgent need to critically reflect on these evaluation methods.

Methodology

The main objective of this thesis is to examine the *effectiveness of various evaluation metrics in assessing RAG question-answering systems*, particularly within the domain of scientific question-answering tasks. To achieve this, we developed three progressively complex experiments to evaluate the metrics' performance. Table 4 provides a quick overview of the experiments, including the datasets and generative models used in each of them.

This section is divided into four subsections, each addressing a key aspect of the research process. First, we introduce the foundations of the three experiments designed to observe the metrics' behavior in different scenarios. Next, we describe the generative models and QA datasets employed for each experiment. Finally, we explain the selected evaluation metrics, grouped into three categories as discussed in 0.2, which will be tested across the experiments.

<table>
<tr><th>Ex.</th><th>Research Question</th><th>Procedure</th><th>DATA</th><th>MODELS</th><th>METRICS</th></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Can the metrics give higher score to "correct" than "incorrect" answers?</td><td>Compute metrics on data aligned (question-answer) vs. shuffled (randomized).</td><td rowspan="2">QASPER
QASA</td><td rowspan="2">LLAMA 2 13b</td><td rowspan="5">Lexical similarity:
BLEU
ROUGE-1/2/L
ChrF and ++
METEOR
TER

Semantic similarity:
BERTScore
WMS
SMS
WISDM

Fine-tuned:
BLEURT
BEM
BARTScore

Prompt-based:
Prometheus
LLM Score
R Faithfulness
R Relevancy
R Correctness
R Similarity
T Consistency
T Similarity</td></tr>
<tr><td>• Observe coverage (instances for which it could compute a score) and variability (coefficient of variation).
• Qualitative analysis (observe example of high and low agreement) and length bias (correlation length and score).
• Metrics correlation (Spearman)</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Can the metrics give higher score to "better" than "worst" answers?</td><td>Compute metrics on data generated by a larger and a smaller model.
+ length bias analysis</td><td>QASPER
QASA</td><td>LLAMA 3.1 8b
LLAMA 3.1 70b</td></tr>
<tr><td>Compute metrics on a dataset with questions with 6 different answer variations.</td><td>Sensitivity Dataset</td><td>LLAMA 3.1 70b</td></tr>
<tr><td>3</td><td>Do the metrics align with human preference?</td><td>Compare metrics ranking with human annotators ranking.</td><td>Sample of 25 q-a from QASPER QASA and Iris.ai</td><td>LLAMA 3.1 8b
LLAMA 3.1 70b
Phi 3
Gemma 9b
Gemma 27b
Mistral</td></tr>
</table>

Table 1: **Summary of experiments.** This table outlines the goals, procedures, data, models, and metrics used across the three experiments performed on this thesis.

1.1 Experiments

To test the different evaluation metrics under diverse and comprehensive conditions, we conducted three experiments of increasing complexity to progressively challenge the metrics and analyze their behavior. The specificities of the models and datasets mentioned will be explained in the following subsections.

The primary motivation behind the experiments was to assess the effectiveness and usefulness of the metrics. At the most fundamental level, we asked: Can the metrics reliably distinguish between entirely "correct" and completely "incorrect" answers? Building on this, we sought to evaluate the metrics in greater detail by investigating their sensitivity to varying degrees of correctness. Specifically, we asked: Can the metrics differentiate between better and worse correct answers? and Can they be misled by incorrect answers that appear correct? Finally, to conclude our analysis, we questioned: Do the metrics align with human preferences?

1.1.1 Experiment 1

For the first analysis, the objective was to *test whether the metrics would assign higher scores to "correct" answers compared to "incorrect" ones.* To create this specific scenario, we based our approach on a very basic logic. We assumed that the answers generated by the LLAMA 2 13b model to the QASPER and QASA datasets were the "correct" answers. Then, to generate "incorrect" answers we randomly shuffled the data, therefore the answer generated for a specific questions were paired with random questions, to which we expected they were not the answer. Then the metrics were calculated for the answers in these two scenarios:

- **Aligned data** (= "correct" answers): each question was paired with its corresponding answer generated by the LLAMA 2 13b model.
- **Shuffled data** (= "incorrect" answers): each question was paired with a random answer from a different question. We assumed that randomizing the answers, they will mostly be paired with questions to which they are not the correct answer.

Consequently, it is expected that the metrics would yield higher scores for aligned data. To assess this, we contrasted the following measures:

1. **Win rate:** this is the proportion of cases in which the metric gave a higher score to the expected condition out of the total cases where the metric could compute a score. It was calculated by counting the instances where the metric gave a higher score to the better condition and dividing that number by the total instances that the metric was able to compute a score on.
2. **Cohen's d:** this is the measure of effect size when comparing the means assigned by each metric to the different conditions [23]. It was computed using the pooled standard deviation of the two groups. This method of calculating Cohen's d accounts for the magnitude of the difference between the group means, providing a standardized measure of effect size.

Afterward, to gain deeper insights on the metrics' behavior, we used the same models and data to conduct some additional evaluations. These focused on the following aspects:

- **Qualitative and Length Analysis:** we conducted a qualitative review of specific examples to better understand the behavior of the metrics. This review included: the top three instances with the lowest and highest overall scores (averaged across all metrics); and the top three instances with the lowest and highest standard deviation in their general scores. This approach allowed us to observe patterns in scoring. Based on this and previous observations, an additional analysis was conducted to examine the correlation (**Spearman Correlation**[84]) between the number of tokens in the answers and the scores.

- **Metric Variation and Coverage:** for each metric, we analyzed the standard deviation (std) of the scores to evaluate the metric's variation. For this, we calculated the **Coefficient of Variation**, which is a statistical measure of the relative variability of data, expressed as the ratio of the standard deviation to the mean. It provides insight into the relative variability of the data compared to its mean, allowing to assess consistency and stability. Additionally, we examined for how many instances of the total amount, the metric was able to compute a score. We express this as a percentage.

- **Correlation Analysis:** we examined the correlation between metrics to determine how closely they relate to each other, particularly when measuring the same or similar constructs (we only consider the scores assigned to QASA and QASPER dataset in aligned condition with answers generated by LLAMA 2 model). Ideally, metrics measuring the same construct should show a moderate to high correlation. To evaluate these relationships, we used **Spearman Correlation** [84], chosen for its flexibility in handling data that does not meet the assumptions of parametric tests. Spearman Correlation measures rank correlation and assesses the monotonic relationship between two variables, allowing us to understand the degree of association between the metrics.

1.1.2 Experiment 2

The objective of this experiment was to conduct a more in-depth analysis to know if the metrics could *distinguish between "better" and "worse" answers*. We wanted assess the sensitivity of the metrics to various answer variations, how sensitive the metrics could be to different levels of "correctness" of an answer, and if they could be tricked by incorrect answers that resemble correct ones in their form or content. To achieve this, we performed two analysis:

- **Analysis 1 - large vs. small model:**

Firstly, we aimed to create a scenario with generally "better" and "worse" answers. For this, we based on the documented hypothesis that larger models generally produce better responses than smaller ones [44][12]. Therefore, we utilized the LLAMA 3.1 model in its two version (70 billion and 8 billion parameters) to create two sets of "correct" but different answers. **Large model** (= "better" answer): answers generated by the LLAMA 3.1 70b were considered the correct "better" ones. **Small model** (= "worse" answer): answers generated by the LLAMA 3.1 8b were considered the correct "worse" ones.

Metrics were calculated for these two conditions, and it was hypothesized that the answers generated by the larger model would receive higher scores than those generated by the smaller one. To assess this, we used the same measures as in Experiment 1 that were already explained: Win rate and Cohen's d.

- **Analysis 2 - sensitivity data set:**

To gain more granular information about the metrics sensitivity to different degrees of "correctness" and to know specifically if they would get tricked by incorrect answers that resembled the correct ones for their form or content, we decided to create an *ad hoc* dataset. We will call this the *"Sensitivity" dataset*, in which each question has six different answers with varying levels of "correctness" by somehow resembling or not the correct answer in content or form (details in Section 1.4.2). The analysis focused on:

- **Metrics Sensitivity Index:** this aspect evaluates whether the metrics assign higher scores to correct answers compared to incorrect ones. To measure this, we calculated the average score given to correct answers and the average score given to incorrect answers, then subtracted the incorrect score from the correct score. A positive result indicates that the metrics are working as expected, with higher values demonstrating stronger differentiation between correct and incorrect answers.
- **Metrics Sensitivity Pattern:** this aspect examines whether the metrics can detect varying levels of granularity within the correct and incorrect answer sets. Bar charts are used to visualize the scores assigned by each metric across six categories: correct perfect, correct similar, correct different, incorrect similar, incorrect related, and incorrect unrelated. Ideally, correct answers should consistently receive higher scores than incorrect answers, and the chart should display a descending trend, with "correct perfect" having the highest score and "incorrect unrelated" the lowest.

1.1.3 Experiment 3

The goal of this final experiment was to *compare the performance of the evaluation metrics against actual human preferences* to know if the metrics align with human preference and how well they do it. To achieve this, we created a random sample of 25 QA pairs from the QASPER and QASA datasets, as well as a proprietary dataset of Iris.AI (specifically 4 from QASPER, 12 from QASA and 9 from a property dataset from Iris.AI). Each of the questions was answered by six different models (Phi 3, LLAMA 3.1 8b and 70b, Mistral and Gemma 2 9b and 27b), resulting in a diverse set of responses.

These question-answer pairs were then uploaded to Qualtrics platform [3] for distribution among expert annotators, given that the questions and answers were specific to the fields of Machine Learning and Natural Language Processing. We specifically sought knowledgeable annotators, which is why the survey was targeted at employees of Iris.AI and students from the Master's program in Language and Communication Technologies. Annotators were asked to evaluate and rank the answers from each model on a scale from 1 (poor answer) to 5 (excellent answer), reflecting their preferences for the quality of each response. The exact task provided to the annotators is shown on Figure 2.

A total of 55 participants contributed, each of them was asked to rank the answers of only 5 questions. Only 30 participants replied to all the questions, we considered all the collected data to establish a human preference ranking for the different models. Subsequently, we applied each of our automatic evaluation metrics to the same set of question-answer pairs. The primary objective was to compare the **humans rankings** of model responses with the **metrics rankings**, providing insights into how closely the metrics align with human judgment, for this we manually inspect and compared the rankings, and also computed Spearman Correlation between rankings (by metrics and by humans).

[3]https://www.qualtrics.com/

Welcome to the RAG evaluation survey

A Retrieval Augmented Generation (RAG) system combines the power of large language models (LLMs) and state-of-the-art document retrieval methods to answer user questions regarding a set of documents. For each question, the system first retrieves a relevant document (the context) and then uses a LLM to answer it. In this survey, you will have to compare the responses of seven LLMs to the same question and context. All the examples come from the scientific domain, so in order to perform this task you must have at least some experience in working with research publications.

In each trial, you will be presented with a document context (an excerpt from a scientific text), a question and several LLM responses. Your task is to score each response, using a scale from 1 (Poor) to 5 (Excellent), according to how well it answers the question given the context. It is up to you to decide what criteria to use for scoring but keep in mind that a good response should be both factually correct (i.e. based on information provided in the context) and relevant to the question asked (i.e. it should answer the question directly rather than providing unrelated information).

Figure 2: **Survey Task.** Exact task provided to the annotators on Experiment 3.

1.2 Evaluation Metrics

Assessing all existing and available metrics is an impractical endeavor due to their sheer number and the continuous development of new methodologies. Such a comprehensive evaluation would far exceed the scope of this thesis, as well as the constraints imposed by available resources and time. Consequently, we have strategically selected a couple of metrics from each category discussed in Section 0.2. This selection process was driven by the metrics' significance as exemplars of their respective approaches, coupled with the accessibility of their code and the feasibility of implementation. Below, we provide a high-level overview of the operational principles of these metrics. Table 2 outlines the classification criteria adopted, derived from our literature review, and highlights the selected representative metrics from each category.

1.2.1 Lexical Similarity

- **Bilingual Evaluation Understudy (BLEU):**

BLEU [69] is a metric for evaluating the quality of machine-translated text by calculating n-gram precision for different values of n (typically 1 to 4). It measures how many n-grams in the candidate translation appear in the reference translations. To avoid rewarding repeated phrases, BLEU uses "clipping," where each n-gram is counted only as many times as it appears in the reference. It also applies a "brevity penalty" to prevent short translations from receiving high scores for using a few common words correctly. Finally, BLEU combines the n-gram precision scores using a geometric mean, which favors lower values, and multiplies by the brevity penalty to produce the final score.

CLASSIFICATION				METRICS
Analysis	**Method**	**Specificity**		
Lexical similarity	**N-gram matching**	**overlap**	**word**	BLEU - ROUGE - METEOR
			character	ChrF - ChrF ++
		distance	**edit distance**	TER
Semantic similarity	**Embeddings + Distance**		**word**	Word mover's and WISDM
			sentence	Sentence mover's, BERTScore, Ragas Similarity
LLM based		**non-generative**	**learned / fine-tuned**	BLEURT, BEM,
		generative		BARTScore, Prometheus
			prompt-based	LLMScore
				RAGAS - Tonic

Table 2: **Classification of Evaluation Metrics.** The table categorizes evaluation metrics into three main types: Lexical Similarity, Semantic Similarity, and Contextual Understanding.

- **Recall-Oriented Understudy for Gisting Evaluation (ROUGE):**

ROUGE [55] is a set of metrics used to evaluate automatic summarization and machine translation by comparing generated text to reference summaries.

It includes four variants, although the most commonly used are: **ROUGE-N (N-gram Co-Occurrence Statistics):** measures the overlap of n-grams (contiguous sequences of n items) between the generated and reference summaries. ROUGE-1, ROUGE-2, and ROUGE-3 are examples where n equals 1, 2, and 3 respectively. **ROUGE-L (Longest Common Subsequence):** measures the longest common subsequence (LCS) between the generated and reference summaries. It captures sentence-level structure similarity. For each of them, you can measure: precision (indicates the accuracy of the generated text in terms of relevant information); recall (indicates how much of the relevant information from the reference is captured by the generated text); f-score (provides a balanced harmonic mean of precision and recall).

- **Metric for Evaluation of Translation with Explicit Ordering (METEOR):**

METEOR [8] was developed to address some weaknesses of BLEU, particularly around synonyms and word order.

METEOR evaluates a translation by aligning it to a reference translation and considering several factors: exact match (it looks for exact word matches between the machine translation and the reference); stem match (it matches words that share a common root or stem); synonym match (it matches words that are synonyms using resources like WordNet); paraphrase match (it matches phrases that mean the same thing); ordering (it considers the order of words and phrases, penalizing translations where the order is significantly different from the reference).

The metric scores translations based on these factors and combines them into a final score using a weighted harmonic mean of precision and recall. Additionally, METEOR includes a penalty for longer mismatches in word order.

- **ChrF (++):**

ChrF [72] and ChrF++ [73] are evaluation metrics introduced to provide a more flexible and potentially more accurate alternative to traditional word-based metrics like BLEU. These metrics are based on character n-grams and were designed to better handle the diverse range of languages and translation peculiarities.

ChrF calculates the F-score (harmonic mean) of precision and recall over character n-grams. Precision and recall are calculated by comparing the n-grams (sequences of characters) of the machine translation output with the reference translation. It allows for different weighting of n-grams of various lengths, providing flexibility in capturing different granularities of the translation quality. The F-score can be adjusted with a beta parameter to balance the importance of precision and recall according to specific needs.

ChrF++ extends ChrF by incorporating word n-grams along with character n-grams. This combination aims to leverage the benefits of both character-level and word-level evaluation, capturing both fine-grained and more global translation errors. The metric considers both precision and recall of character and word n-grams, allowing a more comprehensive assessment of translation quality.

- **Translation Edit Rate (TER):**

TER [82] was introduced to provide a more intuitive measure of machine translation quality. It calculates the number of edits required to change a system output into one of the references. The edits include insertions, deletions, substitutions, and shifts (i.e., reordering of words or phrases).

First, the machine-translated sentence is aligned with the reference sentence. Then, the number of edits needed to make the machine-translated sentence match the reference sentence is calculated. The types of edits are: insertions (adding a word or phrase); deletions (removing a word or phrase); substitutions (replacing one word or phrase with another); shifts (moving a contiguous sequence of words to a different location. Finally, the TER score is calculated as the ratio of the total number of edits to the average length of the reference translations.

1.2.2 Semantic Similarity

- **Word Movers Distance (WMD):**

WMD [47] is a measure developed to quantify the dissimilarity between two text documents. It leverages the concept of word embeddings, where words are represented as vectors in a high-dimensional space. It leverages the concept of the Earth Mover's Distance (EMD) from optimal transport theory.

First, each word in the documents is represented as a vector using a pre-trained word embedding model (like Word2Vec) to capture semantic similarities between words. WMD formulates the problem of comparing two documents as an optimal transport problem [71]. The goal is to find the most cost-effective way to transform one document into the other. The cost of transforming one word into another is given by the Euclidean Distance between the embeddings. A cost matrix is constructed, where each entry represents the distance between the embedding of a word in the first document and the embedding of a word in the second one. Then, it finds a flow matrix that minimizes the total transport cost. Each entry in the flow matrix indicates how much of one word's embedding is transported to another word's embedding. The total cost associated with the optimal flow matrix is the WMD. Mathematically, WMD is computed as the sum of the product of the flow matrix entries and the corresponding entries in the cost matrix. For consistency and to be able to compare it with other metrics, on this work, the WMD was subtracted from 1 to obtain the Word Mover's Similarity (WMS).

- **Sentence Movers Distance (SMD):**

SMD [22] follows the same logic and computation procedure than WMD but using sentence embeddings. It is a metric used to measure the similarity between two texts by considering how sentences in the first text are mapped onto sentences in the second text one.

Sentences are first represented using word embeddings, typically pre-trained vectors (like Word2Vec, GloVe, or more recent models like BERT). Each word in the sentence is mapped to a high-dimensional vector. The idea is to measure the cost of transforming the distribution of embeddings in one sentence into the distribution of embeddings in another sentence. A cost matrix is constructed, where each element represents the distance between an embedding from the first sentence and an embedding from the second sentence. This distance is often calculated using cosine similarity or Euclidean distance between the word vectors. The goal is to find a flow matrix that minimizes the total transportation cost. The flow matrix indicates how much "weight" should be moved from one sentence in the first text to one sentence in the second text. The SMD is the minimum total cost required to transform one sentence's word embedding distribution into the other. This involves solving a linear programming problem to find the optimal flow matrix that minimizes the total cost defined by the cost matrix. For consistency and to be able to compare it with other metrics, on this work, the SMD was subtracted from 1 to obtain the Word Mover's Similarity (SMS).

- **Word importance-based similarity of documents metric (WISDM):**

WISDM [11] is a method for measuring the similarity between documents that focuses on information-carrying words. It selects the most informative words from the documents using a combination of TF-IDF and word embeddings, which helps to focus on the key terms that are most relevant. It combines the advantages of traditional count-based models like TF-IDF with modern word embeddings techniques, making it both efficient and precise.

First, each document is tokenized into words or phrases (tokens). Tokens are scored using the TF-IDF (Term Frequency-Inverse Document Frequency)[78] method. TF-IDF scores represent the importance of tokens in the document relative to a larger corpus. Then, tokens with a TF-IDF score above a certain threshold are selected. These high-scoring tokens are considered to carry the most significant information about the document. For each selected key-token, its vector representation is retrieved from a pre-trained word2vec model. Word embeddings capture the semantic meaning of words in a high-dimensional space. The document is represented as a matrix, where each row corresponds to the word2vec embedding of a key-token. To measure the similarity between two documents, the RV coefficient is used [77]. This coefficient computes the correlation between the matrices and represents the "similarity/closeness" between them.

- **BERTScore:**

BERTScore [98] evaluates text generation models by comparing the contextual embeddings of the candidate and reference sentences, providing a more nuanced and context-aware assessment than traditional metrics. BERT (Bidirectional Encoder Representations from Transformers) embeddings are used in this metric. BERT is a transformer-based model that provides deep contextual embeddings for words in a sentence, capturing the context around each word, which is crucial for understanding nuanced meanings. The embedding of each text can be computed either by averaging its word embeddings or by directly using sentence embeddings [76]. Then, the metric computes the Cosine Similarity between them.

For each word in the candidate (generated) sentence, BERTScore finds the most similar word in the reference (ground truth) sentence using cosine similarity of their BERT embeddings. This similarity is calculated for each pair of words, and these scores are then aggregated to provide a final similarity score

1.2.3 LLM based

Learned metrics

- **Bilingual Evaluation Understudy with Representations from Transformers (BLEURT):**

BLEURT [79] is a metric designed to evaluate the quality of machine-generated text. BLEURT is a trained regression model that predicts quality scores based on rating data. It utilizes a BERT-like architecture, pre-trained on a vast corpus of text to understand the nuances of language, including syntax and semantics. The pre-trained model is then fine-tuned on a dataset specifically created for evaluation tasks. This dataset includes human-rated examples where text pairs are annotated with quality scores, allowing the model to learn to predict these scores accurately. When given a pair of sentences—one being the reference and the other the candidate—BLEURT processes them using its fine-tuned BERT-like model. The model computes a similarity score that reflects the quality of the candidate text. This score considers how fluent the candidate text is and how well it conveys the intended meaning of the reference text.

- **BERT Matching (BEM):**

BEM [13] is a metric developed to evaluate the quality of answers generated by question answering systems.

The BEM metric operates by leveraging the powerful language understanding capabilities of BERT. The BERT model, pre-trained on a large corpus of text, captures the syntax and semantics of language. BEM further fine-tunes this pre-trained BERT model on a specifically created dataset that includes human-rated answer equivalence annotations. This dataset consists of human judgments where candidate answers produced by QA systems are compared with reference answers and rated for their semantic equivalence. To compute the BEM score, the model processes the context, question, reference answer, and candidate answer. The fine-tuned BERT model then predicts whether the candidate answer is equivalent to the reference answer based on the learned understanding of answer equivalence. The output is a similarity score that reflects the degree to which the candidate answer is considered equivalent to or better than the reference answer. This score takes into account the context and question to ensure a comprehensive evaluation.

- **BARTScore:**

BARTScore [97] is a metric that leverages BART (Bidirectional and Auto-Regressive Transformers)[52], a pre-trained transformer model that excels at text generation tasks.

BARTScore uses the BART model to compute the likelihood of a generated text given a reference text. This is done by measuring how well the BART model can reconstruct one text given the other. Specifically, BARTScore calculates the conditional log-likelihood of the generated text given the reference text. This means it measures how probable the generated text is, considering the reference text as input, thereby providing a quantitative measure of text quality based on the BART model's understanding and generation capabilities.

- **Prometheus:**

Prometheus [46] is an advanced metric designed to evaluate the quality of long-form responses generated by language models.

Originally, the Prometheus fine-tuned model leverages a fine-tuned version of the Llama-2-Chat-13B, utilizing a comprehensive dataset known as the FEEDBACK COLLECTION. This dataset includes detailed score rubrics, reference answers, instructions, and feedback generated by GPT-4. The FEEDBACK COLLECTION consists of 1,000 fine-grained score rubrics, 20,000 instructions, and 100,000 responses, making it a robust resource for training the model. The evaluation process in Prometheus involves providing customized score rubrics and reference answers alongside the responses to be evaluated. The model processes these inputs to generate feedback and assign a score from 1 to 5 based on the specified criteria. Furthermore, **prometheus-eval** is a prompt based approach that uses this same logic of score rubrics, reference answers, instructions, and feedback but can utilize any LLMs, particularly, on this thesis we utilized LLAMA 3 70b (for more detail see Section 1.3) and that is why we will consider Prometheus a "prompt based" metric. We acknowledge that using this model may introduce biases in the experiments where LLAMA family models are used to generate answers. However, it was selected due to its demonstrated effectiveness, as well as its accessibility and cost-free availability.

Prompt based metrics

- **LLMScore:**

Many proposed metrics use some variation of OpenAI GPT models to obtain the evaluation score. For example, *Gpt-score* [30] evaluates the quality of generated texts by calculating the conditional probability that these texts adhere to given instructions and contextual information using GPT-3 [12]. *Gpt-judge* [56] also uses a fine-tuned GPT-3 model to classify answers to questions as either true or false. Another metric, *g-eval* [58] evaluates generated outputs by directly prompting GPT-4 [2]. Based on this, it was decided to develop our own implementation to simply prompt an LLM and asking to output a score. We specifically utilized LLAMA 3.1 70b, which again we acknowledge the limitations of using a model that is also utilized to generate answers, but again, chose it due to its demonstrated effectiveness, accessibility and cost-free availability. (more details about implementation and prompt used in Appendix M.1).

- **RAGAS:**

The RAGAS [29] framework aims to evaluate RAG pipelines. It includes many metrics to evaluate both the generation component and the retrieval component, leveraging power LLMs (mainly OpenAI models). In this thesis, we consider the following metrics:

Faithfulness: measures the factual consistency of the generated answer against the given context. To calculate this, a set of claims from the generated answer is first identified. Then each of these claims is cross-checked with the given context to determine if it can be inferred from the context. Specifically, the metric uses a predefined prompt to instruct the language model to break down the complex answer into simpler, fully understandable statements without pronouns; then, it uses another predefined prompt to instruct the LLM to evaluate each simpler statement against the context, the LLM checks if each statement can be directly inferred from the context; finally, it calculates the faithfulness score as the ratio of faithful statements (faithful statements / total statements) to the total number of statements.

Relevance: focuses on assessing how pertinent the generated answer is to the given prompt. It is defined as the mean cosine similarity of the original question to a number of artificial questions, which were generated (reverse engineered) based on the answer. The LLM is prompted to generate an appropriate question for the generated answer multiple times, and the mean cosine similarity between these generated questions and the original question is measured. The underlying idea is that if the generated answer accurately addresses the initial question, the LLM should be able to generate questions from the answer that align with the original question.

Similarity: assess the semantic resemblance between the generated answer and the ground truth. Particularly, the metric calculates the cosine similarity between the normalized embeddings of the ground truth and generated answer.

Correctness: gauges the accuracy of the generated answer when compared to the ground truth. Answer correctness encompasses two critical aspects: semantic similarity (already explained), as well as factual similarity. For the latter, the metric uses a prompt to instruct the LLM to break down the texts into simpler statements. Then it uses another prompt to instruct the LLM to classify each statement as True Positive (statements in the answer that are directly supported by the ground truth) False Positive (statements are not supported), False Negative (statements from ground truth are missing in the answer). Then, it calculates an F1-like score based on the counts of TP, FP, and FN statements. These aspects are combined with the similarity using a weighted scheme to formulate the answer correctness score.

- **Tonic:**

Tonic [4] is a framework developed to evaluate RAG. It includes many metrics to evaluate both the generation component and the retrieval component. In this thesis, we consider the following:

Consistency: measures the percentage of the answer that can be attributed to retrieved context, ensuring that the generated answer maintains factual accuracy and relevance. The metric takes an answer and context and performs the following steps: the LLM is prompted to create a bulleted list of the main points in the RAG system answer. This step uses a predefined prompt to instruct the LLM to break down the complex answer into simpler, fully understandable statements; each main point is evaluated to determine if it can be attributed to the retrieved context. The LLM uses another predefined prompt to assess the faithfulness of each simpler statement against the context, checking if each statement can be directly inferred from the context. The LLM generates a list of verdicts for each statement, where each verdict is either 1 (the statement is faithful to the context) or 0 (the statement is not faithful to the context); the final score is calculated as the ratio of faithful statements to the total number of statements. This score represents the percentage of the answer that is consistent with the context.

Similarity: measures how well the answer corresponds in meaning to a reference answer. The LLM is prompted to grade how well the RAG LLM response matches the reference response on a scale from 0 to 5. This is done using a predefined similarity score prompt to instruct the LLM to evaluate the similarity.

To streamline the experimental process and ensure a consistent evaluation across all metrics, a dedicated **python module** was developed for the automatic computation of all selected metrics' scores. Such module takes as input a data with sets of *question, reference answer, and reference context* and outputs all the *metrics scores* for each input, general statistics on the data and correlation values among different metrics (more technical details about the implementation of the metrics and the module can be found in Appendix M.1).

[4] No published paper available. Information about their metrics can be found on their website: https://docs.tonic.ai/validate/about-rag-metrics/tonic-validate-rag-metrics-reference

1.3 Generative Models

The main models used on this thesis belong to the **LLAMA** (Large Language Model Meta AI) family, a group of advanced open-source language models developed by Meta AI. The LLAMA models are known for their large-scale transformer architecture, which allows them to handle complex language tasks, and are pre-trained on extensive datasets and fine-tuned to generate human-like text with high coherence and relevance. They were particularly chosen as these models support the engine behind the generative component of Iris.AI [5] Chat-Tool.

Particularly, we will use the **LLAMA 2 13b** model [90] (in Experiments 1 and 2) and the **LLAMA 3.1 8b and 70b** [28] (in the Experiment 2 and 3). Finally, for Experiment 3, the following models were also included for comparison: **Phi3** developed by Microsoft [1]; **Gemma 2 (9b and 27b parameters)** by Google [89]; and **Mistral 7b** by Mistral.ai [6].

Specifically, the **LLAMA 2 model 13 billion** parameter version, is a notable member of the LLAMA family. It features a large-scale transformer architecture that has been optimized for enhanced performance in language generation tasks. This model is designed to process vast amounts of text data and generate highly coherent and contextually appropriate responses. LLAMA 2 13b is widely used in various applications, making it a versatile choice for natural language generation. The model was deployed on AWS SageMaker, a managed service by Amazon Web Services (AWS) [7] that supports building, training, and deploying machine learning models at scale, ensuring robust performance and scalability.

Moreover, **LLAMA 3.1** represents the latest iteration in the LLAMA series, bringing further improvements in language understanding and generation capabilities. The 3.1 version was released in 2024 with multiple variants, including models with **8 billion and 70 billion** parameters. These models build upon the architecture of their predecessors, incorporating advanced training techniques and a larger, more diverse dataset to enhance language comprehension and response generation.

We acknowledge that the use of a single model across all experiments would have allowed for more direct and consistent comparisons of results. However, the selection of models was primarily driven by practical considerations, including the availability of computational resources and the preferences of our industry collaborator, Iris.ai. Incorporating multiple models, despite this limitation, enriches the scope of our study by offering a broader evaluation across diverse architectures.

In particular, the decision to employ both the LLAMA 2 model in certain experiments and the LLAMA 3.1 model in others was influenced by timing and performance factors. At the outset of our research, only LLAMA 2 was available. When LLAMA 3 was released in July 2024, we identified the improved performance of LLAMA 3.1 and decided to include it in subsequent experiments to provide a more thorough and up-to-date assessment. Additionally, logistical constraints prevented the continued use of LLAMA 2, further justifying the shift to LLAMA 3.1. Finally, for Experiment 3, all models were selected by Iris.ai in alignment with their research goals.

[5](Iris.AI is the industry collaborator sponsoring this work (more info at https://iris.ai/)
[6]No publication available. Information in: `https://mistral.ai/news/announcing-mistral-7b/`
[7]https://aws.amazon.com/

1.4 Question-Answering Datasets

Regardless of the metric used, a critical component in the evaluation methodology is the data. The reviewed evaluation metrics rely on a reference ground truth answer, and some also require a reference context, including the specific content necessary to answer the question. In any case, the quality of this data is paramount when evaluating a system's performance in QA; biases or inconsistencies can skew the results and mislead conclusions.

The analyses conducted in this thesis required the use of two different types of datasets: 1. existing pairs of scientific questions, contexts, and a single correct answer manually created and curated by humans; 2. an ad hoc generated set of questions, contexts, and multiple answers, including varying degrees of correctness and incorrectness.

1.4.1 Existing data: QASPER and QASA

Ideally, the data necessary to assess a scientific QA task would include questions derived from scientific papers, one or more reference answers created by humans, and a relevant reference context. For the specific purpose of this thesis, such answers should be in the form of long texts, meaning a composition of at least one fully elaborated sentence rather than a few isolated words or simple yes-no responses. Moreover, a comprehensive dataset should encompass questions of diverse and increasing complexity. With these criteria in mind, we reviewed publicly available QA datasets to evaluate their relevance for this study and selected the most suitable ones. More than thirty QA datasets were examined (for more detail, see Appendix **??**). Those that included answers consisting of only a few words or those not originating from the scientific domain were excluded. Consequently, two datasets were chosen, which we will now explain in detail:

- **Question Answering over Scientific Research Papers (QASPER):**

The QASPER [25] dataset was designed to tackle the challenge of building question-answering systems capable of handling complex reasoning across entire research papers, rather than just factoid-based questions. QASPER includes 5,049 question-answer pairs generated from 1,585 NLP research papers. Questions were created by NLP practitioners who read only the title and abstract of each paper and then formulated queries that required detailed answers found within the full text. The answering process was conducted by a different set of NLP practitioners, who also provided supporting evidence from the papers. This dataset was constructed to support a broad spectrum of question types, including those that necessitate evidence synthesis across multiple paragraphs, tables, and figures.

- **Question Answering on Scientific Articles (QASA):**

The QASA [51] dataset was created to address the limitations of existing question-answering datasets by focusing on full-stack reasoning, encompassing associative thinking and logical reasoning. The dataset includes 1,798 question-answer pairs generated from AI and ML research papers. These questions were formulated by AI/ML practitioners through a think-aloud study, ensuring a diverse range of question types such as surface, testing, and deep questions. Annotators, consisting of both general readers and authors, were asked to read entire papers or specific sections, formulate questions, and provide detailed, evidence-based answers. This process aimed to capture the complexity and depth required for advanced question answering, necessitating the synthesis of information across multiple paragraphs and sections. [8]

[8] More detail on how the datasets were processed and used on the Appendix M.3.

The following images explain the compositional process of elaborating the question-context-answer pairs[9]:

On one hand, Figure ?? shows an example taken from QASPER: a question about the paper is written after reading only the title and the abstract. To arrive at the answer, one finds relevant evidence, which can be spread across multiple paragraphs. In this example, to answer the question about "baselines", the reader must realize from evidence from Sections 3 and 4 that "context documents" come pre-ranked in the dataset and the paper's "baselines" select from these "context documents". On the other hand, Figure ?? shows an example instance taken from QASA: a question that the reader/author asks about the paper while reading the paper. To formulate the answer, one classifies whether the paragraph contains evidence to answer the question. Evidential rationales are written for each evidential paragraph and are systematically composed into a comprehensive answer.

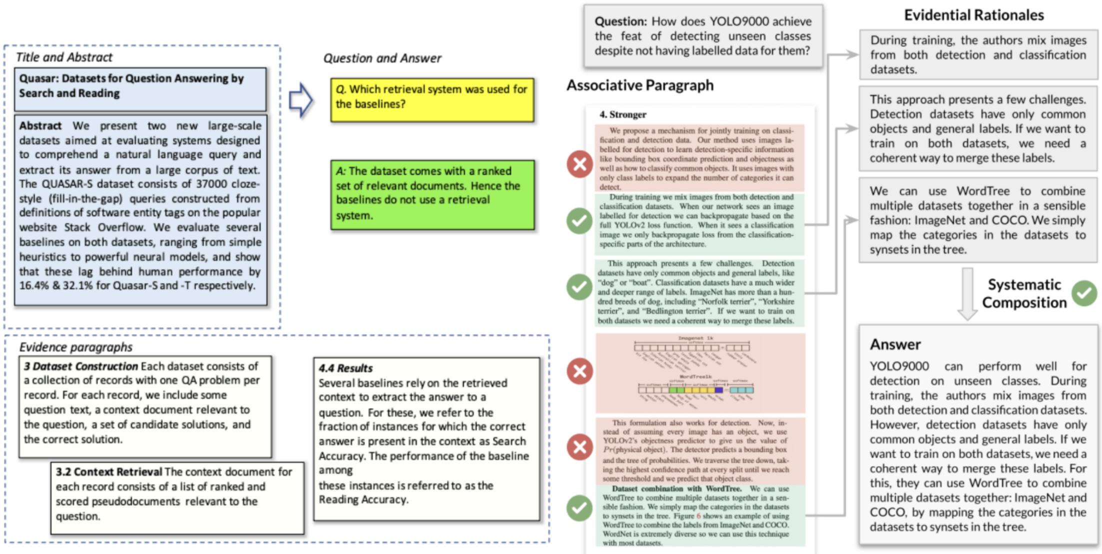

Figure 3: **Example instances from QASPER and QASA datasets.** These figures illustrate example instances from the QASPER (to the left) and QASA (to the right) datasets, where questions are generated based on scientific papers. They were taken from their original publications [25] [51]. In QASPER, questions are posed after reading only the title and abstract, requiring the identification and synthesis of relevant evidence from multiple sections of the paper. In QASA, questions arise during the reading process, with paragraphs evaluated to determine their evidential value in answering the question. Both figures demonstrate the challenges of extracting and composing information systematically from scientific texts to provide comprehensive answers.

For Experiment 1 the full QASPER and QASA datasets were used and populated with answers generated by the LLAMA 2 13b model. For Experiment 3, we used a sample of 25 question-context-answer pairs randomly chosen (4 from QASPER, 12 from QASA and 9 from a property dataset from Iris.AI), they were populated by answer generated by other 6 models. Experiments will be explained in detail at the end of this chapter.

[9]These images were extracted from the original papers where the dataset were presented [25][51].

1.4.2 Ad hoc data: Sensitivity Dataset

This dataset was specifically created to address the second research question: are the metrics sensitive to different levels of “correctness” of an answer? can they be tricked by incorrect answers that seem correct?, and test the metrics’ sensitivity to variations in answers. This dataset includes a set of synthetically generated and verified QA pairs. To encompass questions of varying complexity, two types of questions were generated: **fact-checking questions** (where the answer is a composition based on only one substring of the context, that can be used directly as it is written in the context) and **deep reasoning questions** (where the answer is a composition based on a more elaborated understanding made from more than one substring of the context, and substrings can not be used exactly as they are written, it requires a deeper understanding of the content). Considering the reproducibility, all questions are derived from scientific papers retrieved from PubMed[95], which abstracts served as reference context. Finally, unlike traditional datasets with a single correct answer, this dataset includes six types of answers with varying levels of correctness: *correct perfect, correct similar, correct different, incorrect similar, incorrect related and incorrect unrelated.* Their definition can be found in Table 3.

Correctness type		Definition
CORRECT	PERFECT	the ground truth generated by the model
	SIMILAR	shares a resemblance with the ground truth (paraphrase but shares at least 3 terms)
	DIFFERENT	no resemblance with ground truth but same content (shares less than 2 terms)
INCORRECT	SIMILAR	similar to ground truth (only 1 or 2 of terms change) but completely wrong
	RELATED	related to the topic of the question but does not address it
	UNRELATED	unrelated at all to the topic of the question

Table 3: **Sensitivity Dataset definitions.** This table defines the six types of answers with varying levels of correctness used in the Sensitivity Dataset. The answers are categorized as correct (Perfect, Similar, Different) and incorrect (Similar, Related, Unrelated).

To automate the creation of this dataset, a python module was developed. In a nutshell, this module retrieves abstracts from papers from PubMed (according to specific queries provided by the user) and automatically generates a validates fact-checking and reasoning questions based on them. Then, it also generates and validates the six different answers from each question. The flow diagram of the sensitivity data generation module is shown below on Figure 4:

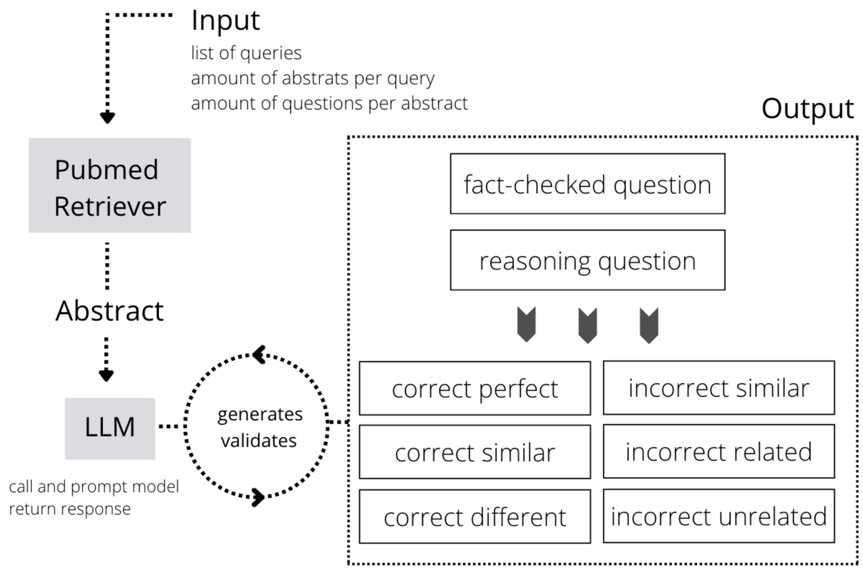

Figure 4: **Workflow of module for Sensitivity data generation.** The image illustrates the pipeline workflow of the sensitivity data generation module developed to create an ad hoc dataset for testing metric sensitivity. The module retrieves abstracts from PubMed based on user-provided queries and generates two types of questions: fact-checking questions and deep reasoning questions. The module then generates and validates six types of answers with varying correctness levels: correct perfect, correct similar, correct different, incorrect similar, incorrect related, and incorrect unrelated, enabling comprehensive testing of answer quality and sensitivity.

After the creation of the module, a *pilot study* was conducted to test various models and prompts, aiming to identify the optimal ones for generating data. The models tested included GPT-3.5[12], GPT-4[2], and LLAMA 3.1 70b[28] (see Section 1.3). Various prompts were used to generate each data item (including questions and all answer types), and the outputs were manually reviewed and verified. Overall, the GPT and LLAMA models demonstrated similar performances, with two notable differences: the LLAMA model adhered more rigorously to the query instructions (for instance, when asked to paraphrase without repeating any terms, GPT repeated more words than LLAMA); additionally, the performance of GPT was somewhat more variable, producing both higher and lower quality outputs across different instances, whereas LLAMA consistently produced outputs of uniform quality throughout the process. Other noteworthy conclusions from this study include: all models struggled to generate variations for answers involving numbers or short names; they all tended to add a brief introductory phrase at the beginning of their responses (such as ”Here is your answer:”); and GPT models generally provided more concise answers, while LLAMA produced longer responses. Additionally, given that GPT models are proprietary and LLAMA is open-source, the decision was made to use LLAMA 3.1 70b.

On the same line, the prompts to be used were selected based on those that provided the output that best aligned with our expectations (see prompts used on Appendix M.4). In most cases, the prompt includes only the instruction and the necessary inputs, with some particular request for each question type. For the “correct different” answer, two examples were included in the prompt, as it was noticed that this improved the models’ performance. As validation, the question and the incorrect answers also used a step of prompting the model to check their quality. The correct perfect answer doesn’t have validation, the other two correct answers (similar and different) use as validation a function that: tokenize the two answers (ground truth and correct similar or different), count the tokens overlap, and returns True if correct similar shares 3 or more and correct different share 2 or less, otherwise returns false. The “correct unrelated” is not generated, instead, a random answer from other questions is used.

Using the developed module, and the chosen model and prompts, already explained in previous sections, one sample data set was created to use in Analysis 2. The queries chosen were: “chemistry”, “environment”, “sports”, “plants”, “linguistics”, and “medicine”. It was set to retrieve 10 abstract per query and 6 questions per abstract (a total of 60 abstracts, and 360 questions, of which 180 are fact-checking and 180 deep-reasoning). The following Table 4 shows one instance as example:

Abstract	Question: fact checking	Answer type	Answer
Pediatric oncology/hematology patients and their families are confronted with a life-threatening situation for which music therapy can be a cross-linguistic field of action. The creative act of making music together offers the possibility to strengthen competences and make conflicts tangible. Besides its complementing of evidence-based biomedical care, there is little research on the feasibility and efficacy of interactive music therapy, including the diagnosed child and their significant others.	What population is confronted with a life-threatening situation for which music therapy can be helpful?	**Correct perfect**	Pediatric oncology/hematology patients and their families are confronted with a life-threatening situation for which music therapy can be helpful.
		Correct similar	Children with cancer and blood disorders, along with their loved ones, face a potentially fatal crisis, and **music therapy** can provide valuable support in this struggle.
		Correct different	Young **patients** with cancer or blood disorders, along with their loved ones, face a critical health crisis, and a melodic intervention can offer valuable support.
		Incorrect similar	Pediatric oncology/hematology patients and their families are confronted with a **routine dental appointment** for which music therapy can be harmful
		Incorrect related	**Music therapy** has been increasingly recognized as a complementary therapy in palliative care settings to reduce anxiety and stress.
		Incorrect unrelated	The main topic of this study is alterations in the **function of K**, specifically K's function.

Table 4: **Sensitivity Dataset example instance.** This table presents a sample instance from the Sensitivity Dataset created using the developed module, showcasing the structure of the dataset used in Analysis 2. The table includes an abstract retrieved from PubMed, a fact-checking question generated based on the abstract, and various answer types classified according to their correctness: Correct Perfect, Correct Similar, Correct Different, Incorrect Similar, Incorrect Related, and Incorrect Unrelated. Each answer type demonstrates varying degrees of correctness and relevance to the question, highlighting the nuanced approach of the dataset in testing QA system sensitivity.

Results

2.1 Experiment 1

Can the metrics reliably distinguish between "correct" and "incorrect" answers? The first experiment tested whether the metrics could *assign a higher score to correct over incorrect answers*. Metrics were computed over QASPER and QASA datasets populated by the LLAMA 2 13b model [10]. The preferred conditions should be **aligned answers** (answer generated by the LLAMA 2 13b paired with their corresponding question) over **shuffled answers** (randomly paired). Table 5 shows the results, metrics were evaluated by **win rate** (percentage of how often the superior condition was preferred) and **Cohen's d** (effect size between conditions).

Category	Metric	Win Rate	Cohen's d
Lexical Similarity	BLEU	80	0,09
	ROUGE-1	83	0,60
	ROUGE-2	70	0,81
	ROUGE-L	83	1,18
	ChrF	86	0,41
	ChrF++	87	0,43
	METEOR	83	1,20
	TER	64	0,56
Semantic Similarity	BERTScore	93	2,10
	WMS	88	1,14
	SMS	94	2,13
	WISDM	87	1,41
LLM based Fine-tuned	BLEURT	96	1,56
	BEM	94	1,68
	BARTScore	80	0,74
LLM based Prompt	Prometheus	91	3,91
	LLMScore	98	5,31
	Faithfulness	76	1,64
	Relevancy	94	0,89
	Correctness	73	0,73
	R similarity	94	2,01
	Consistency	89	2,47
	T similarity	89	2,36

Table 5: **Comparing aligned vs. shuffled data on the QASPER and QASA datasets.** Results include win rates (percentage of how often the superior condition was preferred, darker if >90) and Cohen's d (effect size between conditions, in gray if > 1). Separated results of each dataset can be found in Appendix R.1.

[10] Due to resource constraints, in some cases model-based metrics were calculated on a subsample of 100 instances from each dataset assuming results would roughly generalize on the same direction.

The criteria for interpreting the results was: win rate above 50% indicates good performance favoring the better condition, with results over 80% distinguished as very good; win rate below 50% indicates poor performance, with higher scores given to the worst condition. Effect sizes were considered small for Cohen's d values below 1, moderate 1-3, and large for values above 3, with the sign (+ or -) indicating the direction of the effect. Results reveal the following:

- **Win-rate**: most metrics achieved a win rate above 70%. LLMScore (98%) and BLEURT (96%) had the highest win rates, followed by Ragas similarity and Relevancy, SMS and BEM (94%). Only TER had a lower win rate (64%), being the "lexical similarity" group the one with lowest win-rate on average, although the values remained satisfactory.
- **Cohen's d**: the effect sizes varied across the metrics, with values ranging from small to large, indicating different levels of ability to differentiate between aligned and shuffled conditions. The metric with the strongest effect size was LLMScore (5,31) followed by Prometheus (3,91). Most model-based metrics had a Cohen's d greater than 1, and, in some cases, over 2. Exceptions included BARTScore (0,74) and Ragas Relevancy (0,89) and Correctness (0,73). Similar patterns were observed in the "sentence similarity" metric group, where all metrics achieved values over 1 and 2. In contrast, the "lexical similarity" metrics generally failed this test, with effect sizes below 1 standard deviation. Notable exceptions were ROUGE-L (1,18) and METEOR (1,20), with values slightly above 1.

Regarding the win rates, these findings suggest that **all metrics demonstrated their ability to consistently prefer the expected superior condition.** However, when taking in consideration the effect size, we see that **prompt-based metrics present the most significant differentiation between the conditions**, and therefore the most effective performance. The highlight of the group were LLMScore and Prometheus, although the possible bias of these metrics based on the fact that they utilize a model from the LLAMA family to score should be taken into consideration before jumping to conclusions. The exceptions are Ragas Relevancy and Correctness, further analysis in following results will shed more light on the specific behavior of these metrics.

Similar patterns were observed in the "sentence similarity" metric group, where all metrics demonstrated particularly strong effects. Interestingly, BertScore and SMS showed even better performance than model based "learned" metrics. In contrast, the **"lexical similarity" metrics had the worst performance on this analysis**, with win-rates and effect size lower than the other groups, suggesting that even though in most cases they gave a higher score to the preferred condition, their differentiation between aligned and shuffled data was more limited. Notable exceptions were ROUGE-L and METEOR. Like this, METEOR proves to be the most advanced and sophisticated method of its group.

After conducting the initial analysis, we wanted to have a deeper understanding of their behavior. Therefore, we decided to extend our analysis through a qualitative review of specific examples. From both the QASPER and QASA datasets in the aligned condition (answers paired with their corresponding question), we selected the *top three instances with the lowest and highest overall scores* (averaged across all metrics), and the *top three instances with the lowest and highest standard deviation* in their overall scores. For details on these examples, see Appendix R.2. This qualitative approach allowed us to identify a possible pattern in the evaluation metrics' behavior, where shorter answers tended to have lower scores and lower standard deviation. First, we checked the length of the reference and generated answers in both datasets finding that the average amount of words per answer was: 17,61 for the reference and 18,78 for the generated answer on QASPER; and 38,96 for the reference and 31,97 for the generated answer on QASA. The QASPER data was fairly balanced, on QASA the generated answers were shorter, according to the hypothesis they should have gotten lower scores, but that was not the case.

Based on these, we performed an additional analysis examining the *correlation between answers length and metric scores*. Specifically, we assessed whether there was any significant association between the number of tokens in the answers generated by the LLAMA 2 13b and the evaluation scores assigned by the metrics, we used Spearman correlation coefficients. The results of this analysis are presented in Table 9. Our interpretation of the magnitude of the correlation will be as follows through all this work: 0,00-0,19 very weak or no correlation and 0,20-0,39 weak (marked in red color); 0,40-0,59 moderate (marked in yellow color); 0,60-0,79 strong and 0,80-1,00 very strong (marked in green color). Additionally, only significant correlation will be considered, that is those with p-values < 0.05 (marked in the Table with a (*) next to the coefficient).

Category	Metrics	Spearman Correlation
Lexical Similarity	BLEU	-0,42*
	ROUGE-1	-0,16*
	ROUGE-2	-0,05
	ROUGE-l	-0,13
	ChrF	-0,44*
	ChrF++	-0,44*
	METEOR	0,13
	TER	0,54*
Semantic Similarity	BERTScore	-0,07
	WMS	-0,02
	SMS	0,01
	WISDM	0,02
LLM based Fine-tuned	BLEURT	0,03
	BEM	-0,03
	BARTScore	0,04
LLM based Prompt	Prometheus	0,11
	LLMScore	0,06
	Faithfulness	0,03
	Relevancy	0,24*
	Correctness	0,04
	R Similarity	-0,10
	Consistency	0,05
	T Similarity	0,07

Table 6: **Analysis answer length - score.** This table shows the Spearman Coefficient computed between answer length generated by LLAMA 2 13b and scores. Sign (*) indicates significance in the statistics, p-value < 0,05. Grade of color indicates the strength of the correlation.

It can be observed that for most of the metrics, the correlation is non-significant, with many coefficients near zero and p-values over 0.05. Only the "lexical similarity" group and Relevancy showed significant correlation, although moderate to weak on strength. From this group, the correlation between answer length and metric scores is sometimes positive (which will indicate they increase together) and sometimes negative (which will indicate when one increases, the other variable decreases). Particularly, **BLEU, ChrF (++) and TER**, showed the strongest correlation, although still moderate.

This balanced mix of positive and negative correlations could suggest that answer length can sometimes lead to higher scores and sometimes to lower ones on specific metrics. However, the low correlation strengths and non-significance (p-values > 0,05) in the majority of the cases indicate that the relationship between length and scores is weak or negligible for most metrics. Counter to what was expected, this reinforces the idea that **answer length does not significantly impact the evaluation outcomes**, and overall, the metrics appear not to exhibit substantial biases based on answer length.

Afterward, we wanted to further assess the performance of the metrics. Therefore, *variation and coverage* were analyzed. To capture Coverage, we computed the percentage of instances that the metric succeed to score over the total amount. Variation was assessed using the Coefficient of Variation (CV), calculated as the standard deviation of the scores divided by the mean. A lower CV indicates lower variation, and vice versa. Results are displayed in table 7.

Category	Metric	Variation	Coverage %
Lexical Similarity	BLEU	5,25	99,6
	ROUGE-1	0,91	99,6
	ROUGE-2	1,76	99,6
	ROUGE-L	0,95	99,6
	ChrF	0,73	99,6
	ChrF++	0,74	99,6
	METEOR	0,78	99,6
	TER	1,13	99,6
Semantic Similarity	BERTScore	0,40	99,6
	WMS	0,43	100
	SMS	0,41	100
	WISDM	0,38	**72,2**
LLM based Fine-tuned	BLEURT	0,74	99,3
	BEM	0,70	99,3
	BARTScore	2,08	100
LLM based Prompt	Prometheus	0,22	100
	LLMScore	0,23	95,3
	Faithfulness	0,41	**74**
	Relevancy	0,23	99,1
	Correctness	0,30	99,1
	R Similarity	0,05	99,2
	Consistency	0,44	100
	T Similarity	0,59	100

Table 7: **Metrics Variation and Coverage.** The table shows coverage (% of instances the metric was able to compute) and variation (Coefficient of Variation: shades represent the strength of variation. Lightest represents lower values, while darker shades represent higher values).

- **Coverage**: most metrics successfully computed scores for the majority of instances, **showing robust coverage**. However, **WISDM and Faithfulness** exhibited lower coverage. WISDM struggled with short answer strings, while Faithfulness depended on specific contextual elements that were not always present. This underscores the need to apply these metrics in contexts where their input requirements are met for meaningful evaluations.

- **Variation: BLEU, Bart, ROUGE-2, and TER** showed the highest Coefficient of Variation (CV) values, all above 1, indicating stronger score fluctuations. Assuming that both QASPER and QASA datasets contain all answers of similar quality (as authors express they were carefully and systematically curated by humans), these results suggest that these metrics may provide less reliable results, as there shouldn't have been such variability. A moderate variation can be desirable, as it indicates the metrics aren't systematically giving similar scores. In contrast, the remaining metrics had CV values below 1, showing more consistent performance with scores close to their means. While this consistency is beneficial if all dataset instances are of similar quality, it could be problematic with diverse data, indicating a bias toward assigning similar scores. Overall, **lexical similarity and learned metrics were the most variable, while semantic similarity and prompt-based metrics showed the least variability**.

Finally, the last analysis aimed to determine whether there was any *correlation between the evaluation metrics*. The goal was to explore the level of agreement within each category and determine if simpler metrics correlate with more sophisticated ones. Since advanced metrics typically require more computational resources and time, identifying strong correlations could suggest that using basic metrics—if they perform similarly—would be more efficient, offering a cost-effective and time-saving alternative without compromising evaluation quality.

Figure 5 illustrates the Spearman correlation heatmap. Spearman correlation was chosen for its ability to capture the strength and direction of monotonic relationships between variables, regardless of the distribution of the data. Each cell in the heatmap contains the Spearman correlation coefficient, with an asterisk (*) indicating statistically significant correlations (p-value $\leq$ 0,05).

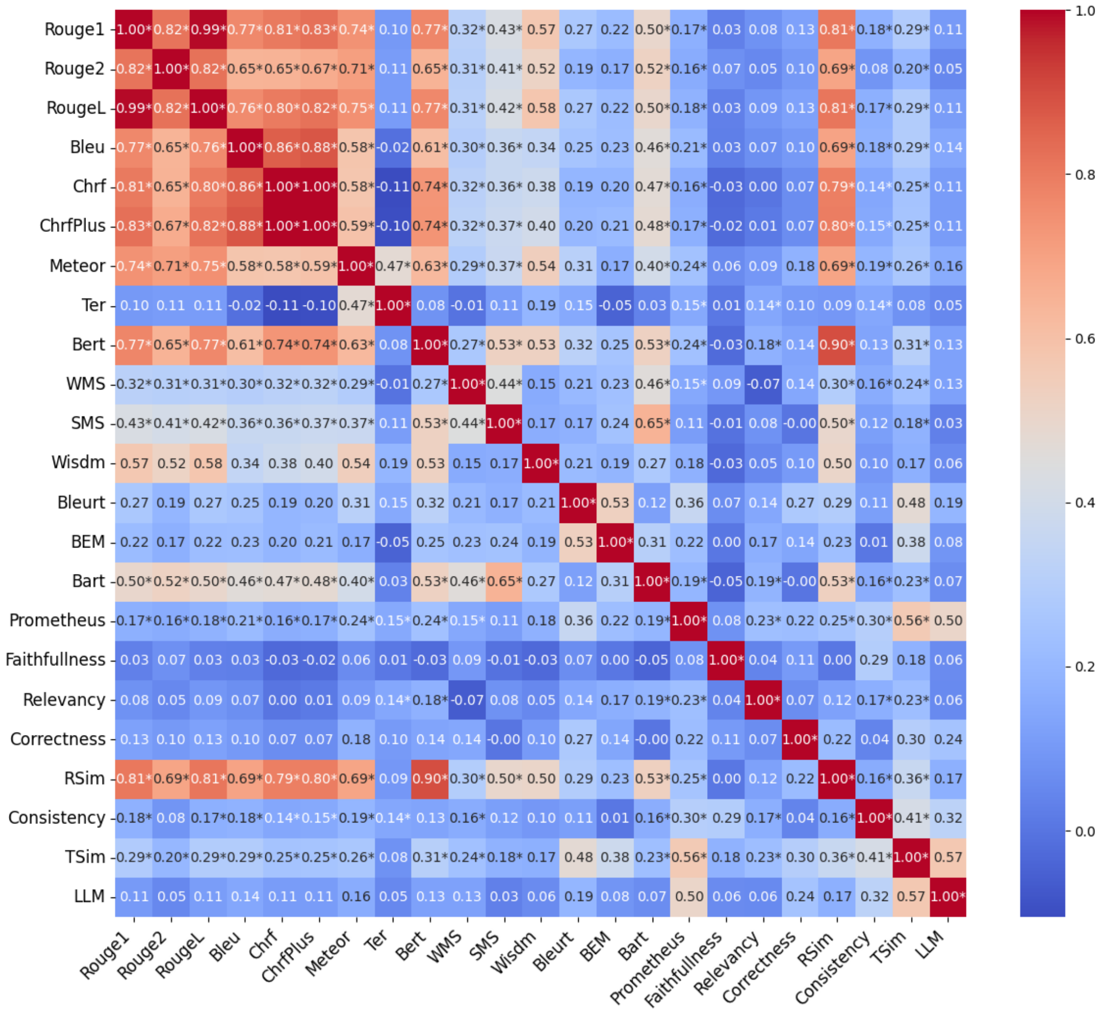

Figure 5: **Spearman Correlation among metrics.** The heatmap presents the relationships between various metrics applied to LLAMA 2-generated answers on QASPER and QASA. Cells marked with an asterisk (*) indicate statistically significant correlations (p-value $\leq$ 0.05). For a more granular breakdown of the results for each dataset individually, refer to Appendix R.3. R.3.

This analysis sheds light on the extent to which metrics designed to assess similar aspects of answer quality behave consistently across different datasets. The heatmap reveals varying correlations among metrics, indicating different degrees of overlap in what each measures. High Spearman correlations suggest metrics capture similar aspects of answer quality, while low correlations indicate distinct evaluations.

- **Lexical similarity Metrics:** metrics like ROUGE-1, ROUGE-2, and ROUGE-L showed **strong correlations** (Spearman ¿ 0.8, significant), confirming their overlap in measuring surface-level lexical similarity, as they all rely on n-gram matching. BLEU, ChrF, and ChrF++ displayed moderate to strong correlations with the ROUGE family, highlighting their shared focus on word and phrase overlap. In contrast, **TER** had little to no significant correlation with other lexical metrics, except for a mild correlation with METEOR, indicating TER's distinct focus on edit distance rather than n-gram similarity.

- **Semantic similarity Metrics:** metrics like BERTScore, WMS, SMS, and WISDM exhibited **low correlations**, suggesting they capture different facets of semantic similarity. Notable exceptions include moderate correlations between BERTScore, SMS, and WMS, indicating some shared evaluation of meaning. **BERTScore** also showed moderate correlations with lexical metrics, likely due to its partial reliance on token embeddings, bridging lexical and semantic evaluations.

- **Model-Based Metrics:** learned metrics such as BLEURT, BEM, and BARTScore showed **no significant correlations** with each other or with lexical metrics, reflecting their distinct approaches to evaluating answer quality. However, prompt-based metrics had low and mild correlations in general, with **R Similarity (Ragas similarity)** being the exception, showing strong correlations with both lexical and semantic metrics. This suggests that R Similarity captures both surface-level and deeper contextual similarities, making it a hybrid metric.

Interclass Correlations:

- **Lexical and Semantic similarity:** correlations between these two types of metrics were **generally low**, with BERTScore as an exception, showing strong correlations with most lexical metrics except TER. This suggests BERTScore captures both lexical overlap and contextual meaning.

- **Lexical similarity and Model-Based Metrics:** model-based metrics had **weak correlations** with lexical metrics, except for R Similarity, which showed strong correlations, indicating it captures aspects from both categories.

- **Semantic similarity and Model-Based Metrics:** model-based metrics showed **weak correlations** with semantic metrics, again with RSimilarity being an exception, correlating with BERTScore and SMS.

This correlation analysis highlights the complexity of evaluating answer quality with diverse metrics. Metrics designed to measure similar constructs tend to correlate strongly, but the variation among others emphasizes the need for a combination of metrics to fully assess answer quality. Lexical metrics reliably measure word overlap, while semantic and model-based metrics offer deeper insights into fluency, coherence, and meaning. Low correlations among metrics suggest they **each provide unique perspectives**.

2.2 Experiment 2

Can the metrics reliably distinguish between "better" and "worst" answers? After analyzing the results of Experiment 1, we realized we needed a way to evaluate the metrics in greater detail, with a more challenging test than a simple distinction between clearly correct and incorrect answers. For this, we developed two analysis:

- **Analysis 1 - large vs. small model:**

Table 8 shows the results when comparing scores assigned to answers generated by a larger model (LLAMA 3.1 70b), considered "better", and a smaller model (LLAMA 3.1 8b) considered "worse", although both presumably correct [11]. Metrics were evaluated by **win rate** (percentage of instances where superior condition was preferred) and **Cohen's d** (effect size).

Category	Metric	Win Rate	Cohen's d
Lexical Similarity	BLEU	14	-0,26
	ROUGE-1	16	-0,32
	ROUGE-2	11	-0,23
	ROUGE-L	17	-0,29
	ChrF	24	-0,28
	ChrF++	24	-0,28
	METEOR	16	-0,29
	TER	11	-0,37
Semantic Similarity	BERTScore	22	-0,31
	WMS	34	-0,18
	SMS	23	-0,29
	WISDM	30	-0,20
LLM based Fine-tuned	BLEURT	36	-0,08
	BEM	36	-0,08
	BARTScore	33	-0,06
LLM based Prompt	Prometheus	15	-0,22
	LLMScore	23	-0,05
	Faithfulness	14	**0,09**
	Relevancy	35	**0,02**
	Correctness	42	-0,12
	R similarity	32	-0,28
	Consistency	11	-0,13
	T similarity	14	-0,25

Table 8: **Comparing large vs. small model answers on the QASPER and QASA datasets.** Results include Cohen's d (**bold** if **positive**) and win rates (darker if >30). Separated results of each dataset can be found in Appendix R.1.

- **Win-rate**: most metrics exhibited win rates below 30%, often favoring the smaller 8B model. Then, all learned Metrics, WMS, WISDM and Relevancy, Correctness and R Similarity had relatively higher win rates but still always preferred the worst condition (smaller model). This could suggest a slightly better performance of model based metrics, but still not desirable.
- **Cohen's d**: effect sizes were generally small (all below 1 in absolute value), with many metrics showing negligible values. Notably, even though most values are negative (indicating the effect was in favor of the smaller model), Ragas Faithfulness (0,09) and Relevancy (0,02) were exceptions, which showed very modest but positive effects.

[11] Similar to experiment 1, due to resource constraints, in some cases model-based metrics were calculated on a subsample of 100 instances from each dataset assuming results would roughly generalize on the same direction.

Overall, the analysis revealed persistently low win rates for most metrics, underscoring **metrics exhibited a preference for the outputs of the smaller model**, counter to expectations. Additionally, the small effect sizes observed across all metrics further emphasize their **difficulty in robustly differentiating between the performance of the larger and smaller models**, undermining the assumption that the larger model should produce superior outputs.

This pattern may suggest a shortcoming in current metrics' ability to detect meaningful performance differences based on model size. Alternatively, it could be due to dataset characteristics, as these metrics rely heavily on reference answers, and ground truth quality may affect their performance. The bias toward smaller models may reflect the metrics' sensitivity to answer features like length or complexity. Further qualitative analysis could help identify patterns in the answers produced by each model version. According to the literature [44][12], larger models tend to produce longer, more nuanced responses, so we explored whether answer length influenced evaluations, possibly explaining the metrics' preference for shorter, smaller-model outputs.

In our analysis of the word count, we observed that in the QASPER dataset, the larger model generated answers with an average length of 3,50 words, while the smaller model produced longer answers, averaging 8,53 words. A similar pattern emerged with the QASA dataset, where the smaller model's answers averaged 10,40 words, compared to the larger model's average of 8,79 words. These findings consistently show that, contrary to expectations, the smaller model tended to generate longer answers across both datasets. Consequently, we computed Spearman Correlation between answer length and scores for this data; results are displayed in Table9.

Category	Metrics	Spearman Correlation
Lexical Similarity	ROUGE - 1	0,56*
	ROUGE - 2	0,45*
	ROUGE - L	0,28*
	BLEU	0,28*
	ChrF	0,59*
	ChrF ++	0,58*
	METEOR	0,45*
	TER	0,68*
Semantic Similarity	BERTScore	0,52*
	WMS	0,28*
	SMS	0,49*
	WISDM	0,25*
LLM based Fine-tuned	BLEURT	0,17*
	BEM	0,17*
	BARTScore	0,36*
LLM based Prompt	Prometheus	0,28*
	LLMScore	0,22*
	Faithfulness	0,08
	Relevancy	0,36*
	Correctness	0,12*
	R Similarity	0,47*
	Consistency	0,40*
	T Similarity	0,39*

Table 9: **Analysis answer length - score.** This table shows the Spearman Coefficient computed between answer length generated by LLAMA 3.1 model (8 and 70b) and scores. Sign (*) indicates significance in the statistics, p-value < 0,05. Grade of color indicates the strength of the correlation, only significant ones were considered.

The findings of this length bias analysis do not align entirely with those from the one made on Experiment 1. In this case, almost all metrics (except Faithfulness) present a positive correlation of varying degrees (indicating the longer the answer is the higher score it would get).

Particularly, lexical similarity metrics exhibit a moderate and statistically significant positive correlation, suggesting that longer answers tend to receive slightly higher scores. Notably, BLEU shows a weak correlation, while TER demonstrates a strong one. On the other hand, semantic similarity and model-based metrics generally reveal a low but positive correlation, indicating that while longer answers may achieve higher scores, the effect is relatively modest. Metrics such as BERTScore, SMS, R-similarity, and Consistency, however, show a more moderate correlation. This pattern may suggest why the metrics favored the smaller model, which produced longer answers on average, though the correlations are not strong enough to draw definitive conclusions.

- **Analysis 2 - sensitivity dataset:**

As the results from Experiment 2 were not satisfactory, we strengthen our analysis by utilizing a dataset specifically crafted to challenge the metrics, with answers presenting controlled and different levels of Correctness (see 1.4.2). Like this, we hoped to obtained more specific information about the metrics behavior when having to assess diverse answers, specifically: correct perfect (ground truth), correct similar (similar in form to ground truth), correct different (similar in content but not in form to ground truth), incorrect similar (similar in form to ground truth but incorrect), incorrect related (similar in content to ground truth but incorrect) and incorrect unrelated (completely different to ground truth).

Category	Metrics	Sensitivity Index
Lexical Similarity	BLEU	0,15
	ROUGE-1	0,17
	ROUGE-2	0,15
	ROUGE-L	0,18
	ChrF	0,12
	ChrF++	0,13
	METEOR	0,18
	TER	0,31
Semantic Similarity	BERTScore	0,30
	WMS	0,12
	SMS	0,30
	WISDM	0,20
LLM based Fine-tuned	BLEURT	0,34
	BEM	0,12
	BARTScore	0,18
LLM based Prompt-based	Prometheus	0,50
	LLMScore	0,57
	Faithfulness	0,36
	Relevancy	0,05
	Correctness	0,25
	R similarity	0,09
	Consistency	0,005
	T similarity	0,68

Table 10: **Sensitivity Index.** The table shows the Sensitivity Index values for various metrics, indicating their effectiveness in differentiating between correct and incorrect answers based on the sensitivity dataset. The color shade agrees with value strength.

We propose that an ideal metric would: 1. give higher scores to all the correct answers over all the incorrect ones, 2. present a slope in which the answers will get progressively lower scores as they get farther in form and content from the ground truth. Based on the metrics logic and procedure to score, we expected to be able to show that more rudimentary evaluation approaches (as lexical and semantic similarity metrics) would show the worst and more undesirable performance; while more advanced approaches (model based) would show more flexibility, and therefore a performance closer to the desired one.

The results of this analysis are summarized in Table 10, which presents the *Sensitivity Index* of various metrics. This index measures the ability of each metric to differentiate between correct and incorrect answers based on the sensitivity dataset, and we computed as the average score assigned to all correct answers minus the average score assigned to all incorrect answers. A positive Sensitivity Index indicates that the *metric assigns higher scores to correct answers compared to incorrect ones*, with larger values reflecting a stronger ability to distinguish between these answer types. To be able to compare this numbers with each other, the values of the metrics with a range that was not from 0 to 1 were normalized (BLEU, ChrF and ChrF++ were divided by 100, Prometheus and Tonic similarity were divided by 5).

Based on this *Index*, it can be observed that **all metrics gave on average higher score to the correct group of metrics rather than the incorrect group.** This is confirmed as all the Index values are positive. This aligns properly with the findings of the first experiment. However, the degree of distinction between the two conditions differs from metric to metric, as reflected in the different values obtained.

T. similarity, LLMScore, and Prometheus, from the model prompt-based group, were the most sensitive metrics, generally assigning higher scores to correct answers compared to incorrect ones. These were followed by **R. Faithfulness, BLEURT, TER, BERTScore, and SMS**. The metrics with the least sensitivity, that is the lowest Index indicating they did not significantly differentiate between correct and incorrect answers, were **T. Consistency, R. Relevancy, and R. similarity** from the model prompt based group as well.

In terms of categories, the "lexical similarity" group had the lowest Index values. Follow by "semantic similarity" and model based learned metrics.

To explore this further, a more detailed analysis was conducted. Bar charts were created for each metric to illustrate their *sensitivity across different categories of answers*. Bar charts are displayed on figures from 6 to 28. In these charts, the Y-axis represents the mean score assigned by each metric to each answer category, which is displayed on the X-axis. All green bars refer to correct answers and red bars incorrect ones. These visualizations provide a comparative view of how each metric scores the six predefined answer categories, highlighting their performance in distinguishing between correct and incorrect answers.

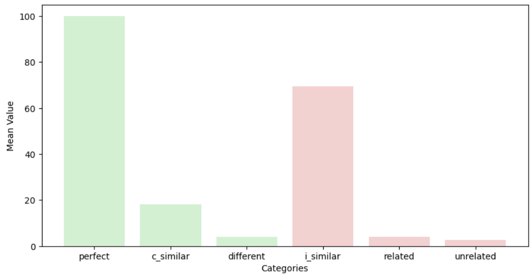

Figure 6: BLEU Sensitivity

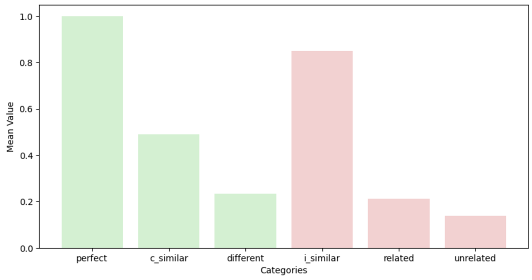

Figure 7: ROUGE 1 Sensitivity

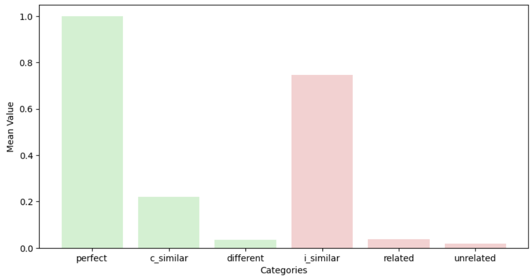

Figure 8: ROUGE 2 Sensitivity

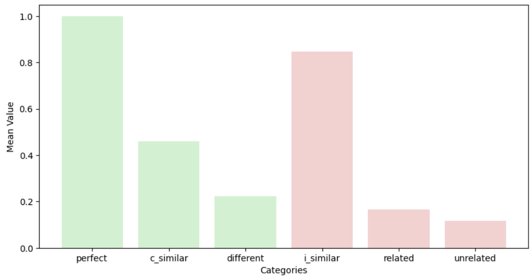

Figure 9: ROUGE L Sensitivity

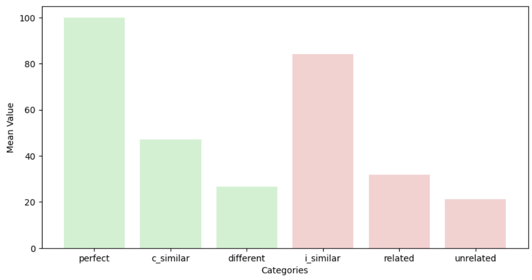

Figure 10: ChrF Sensitivity

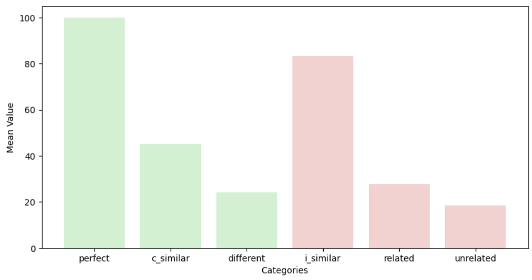

Figure 11: ChrF + Sensitivity

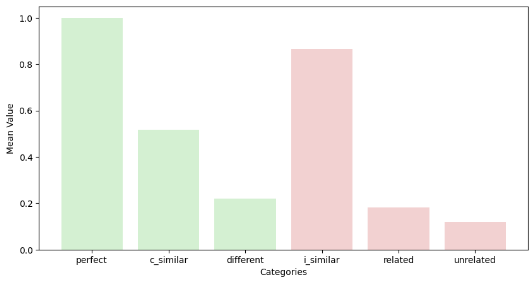

Figure 12: METEOR Sensitivity

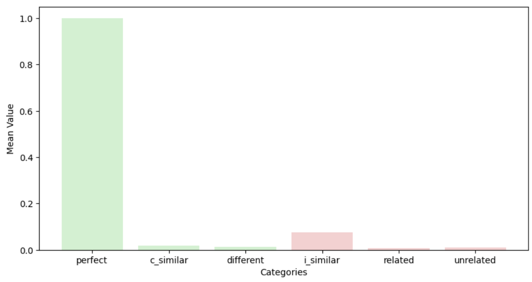

Figure 13: TER Sensitivity

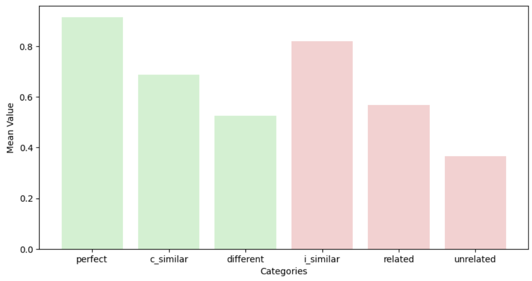

Figure 14: WMS Sensitivity

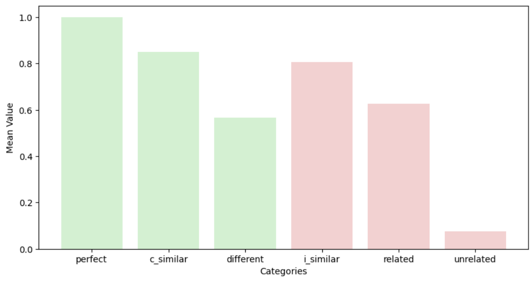

Figure 15: SMS Sensitivity

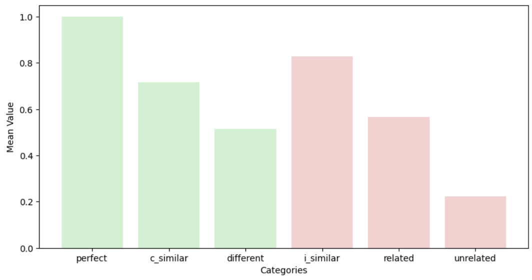

Figure 16: WISDM Sensitivity

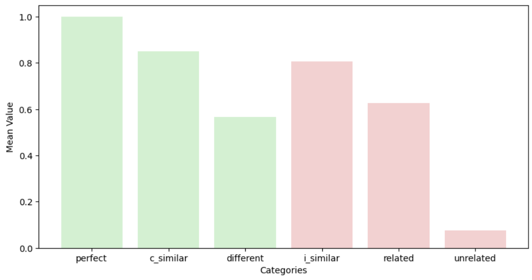

Figure 17: BERTScore Sensitivity

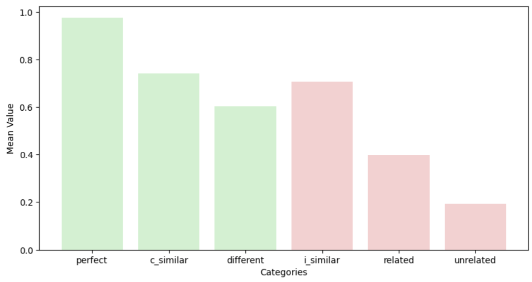

Figure 18: BLEURT Sensitivity

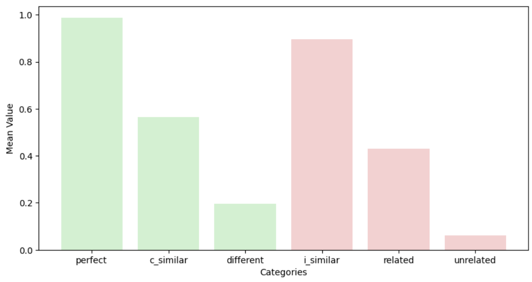

Figure 19: BEM Sensitivity

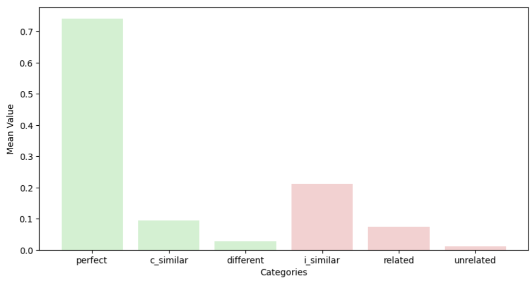

Figure 20: BARTScore Sensitivity

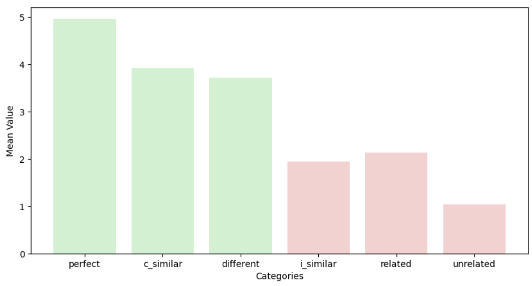

Figure 21: Prometheus Sensitivity

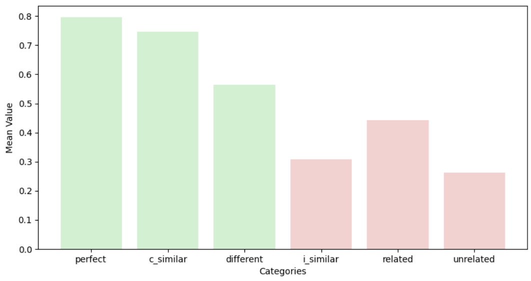

Figure 22: Faithfulness Sensitivity

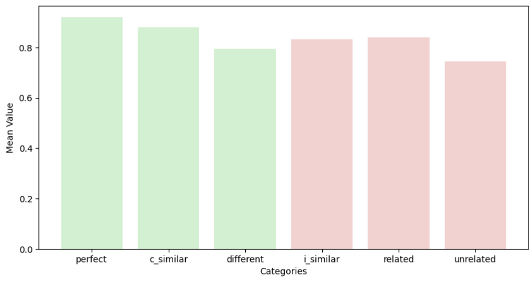

Figure 23: Ragas Relevancy Sensitivity

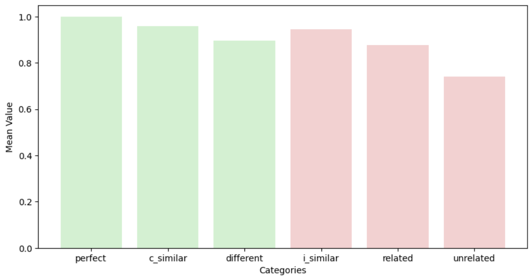

Figure 24: Ragas similarity Sensitivity

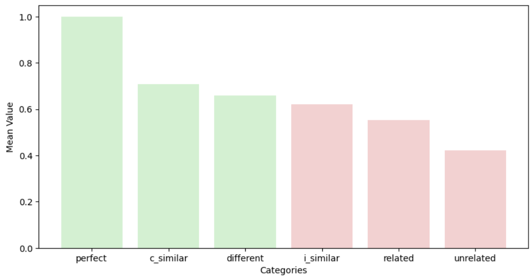

Figure 25: Ragas Correctness Sensitivity

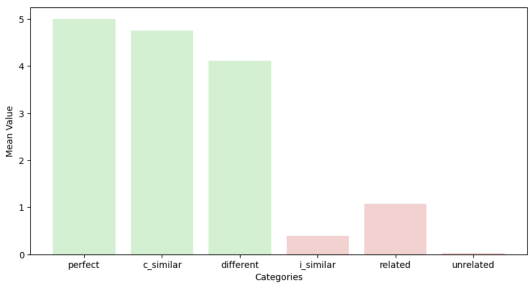

Figure 26: Tonic similarity Sensitivity

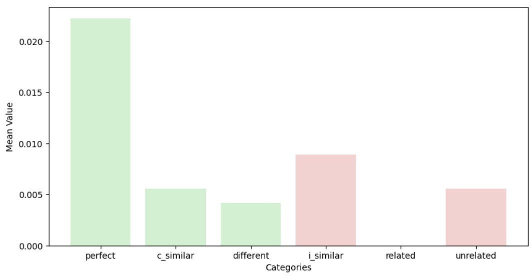

Figure 27: Tonic Consistency Sensitivity

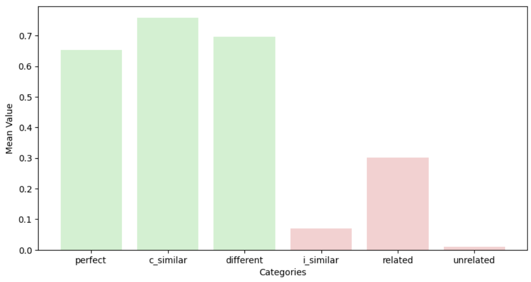

Figure 28: LLMScore Sensitivity

We identified five distinct “patterns” in the behavior of the metrics:

1. **High Score Only for the “Correct Perfect” Answer (Ground Truth):** In this pattern, metrics are highly conservative and assign a high score solely to the “correct perfect” answer, which matches the ground truth exactly. All other answer categories, even those that are correct but differ slightly from the perfect answer (e.g., correct similar or correct different), are penalized with lower scores. This behavior indicates that these metrics are rigid in their evaluation, heavily prioritizing answers that resemble the reference answer in both structure and content. As a result, answers that may be semantically correct but vary in phrasing, structure, or detail are unfairly penalized. This rigidity can be problematic for tasks that require flexibility in how correct answers are expressed. Metrics following this pattern include: *TER, BARTScore, and Tonic Consistency.* These metrics prioritize surface-level similarity (word choice or structure) over deeper semantic Correctness.

2. **High Scores for “Correct Perfect” and “Incorrect Similar” Answers:** Metrics exhibiting this pattern assign high scores to both the “correct perfect” and “incorrect similar” answers. This means they value surface-level similarity, even when it leads to incorrectly identifying answers as correct. For example, an answer that is factually incorrect but shares significant lexical overlap with the correct answer would still receive a high score. This pattern reveals that such metrics have a bias toward lexical similarity, making them prone to over-rewarding incorrect answers if they appear similar on the surface. Metrics following this pattern include: *BLEU, all ROUGE metrics, ChrF, and ChrF++.*

3. **Descending Score Pattern Within Categories:** In this pattern, metrics show a descending score trend within each category: *correct perfect* > *correct similar* > *correct different* and *incorrect similar* > *incorrect related* > *incorrect unrelated.* Although “correct perfect” and “incorrect similar” receive the highest scores, other categories still obtain meaningful scores, suggesting that the metrics recognize gradations of Correctness or similarity. This behavior is a middle ground, capturing the relative quality of answers rather than strictly favoring either perfect Correctness or surface similarity. Metrics following this pattern include: *METEOR, WMS, SMS, Bert, WISDM, BLEURT, and BEM.*

4. **Mild Descending Slope with Generally High Scores, Even for Incorrect Answers:** These metrics exhibit a mild descending slope, but all scores remain relatively high, even for incorrect answers. This suggests that the metrics struggle to differentiate strongly between correct and incorrect answers, leading to high scores being assigned across the board. This pattern may indicate that the metrics are not sufficiently sensitive to factual Correctness and tend to overestimate the quality of incorrect answers. Metrics following this pattern include: *Ragas Relevancy and Correctness.*

5. **Higher Scores for Correct Answers and Lower Scores for Incorrect Answers:** Considered the most desirable behavior, this pattern demonstrates that metrics consistently assign higher scores to correct answers, regardless of their variations from the ground truth, and lower scores to incorrect answers. These metrics appear to capture a balance between rewarding Correctness and flexibility, showing a more comprehensive understanding of the answers’ content. Metrics following this pattern include: *LLMScore, Tonic similarity, Ragas similarity and Faithfulness, and Prometheus.*

Upon analyzing the patterns of metrics from each evaluation category, the following observations were made:

- **Lexical similarity Metrics:** these metrics generally assign the highest scores to "correct perfect" and "incorrect similar" responses, while scoring other categories significantly lower. The exception is TER, which consistently scores only "correct perfect" responses highly. This behavior indicates that lexical metrics are **highly sensitive to surface-level variations** that deviate from the ground truth and are easily misled by answers that resemble the correct response superficially, even when they contain incorrect information.
- **Semantic similarity Metrics:** these metrics generally assign higher scores across all correct answers, demonstrating a **greater tolerance for variations** compared to lexical metrics. They effectively recognize correct answers that differ from the original ground truth, distinguishing themselves by rewarding variations that are still accurate. However, this tolerance also extends to incorrect answers, leading to high scores not only for "correct" responses but also for "incorrect similar" and "incorrect related" ones. Although "incorrect unrelated" answers typically receive the lowest scores, the metrics still tend to **overestimate the quality of superficially similar incorrect responses**, reducing their effectiveness in clearly distinguishing between correct and incorrect answers.
- **Learned Metrics:** BEM exhibits behavior similar to lexical metrics, showing limited ability to differentiate based on answer quality. BLEURT performs reasonably well but is often misled by "incorrect similar" responses, assigning high scores to answers that are superficially similar but incorrect. BARTScore tends to favor ground truth responses disproportionately, failing to generalize well to other correct answers. This variability within the "learned metrics" group highlights their **differing effectiveness**, influenced by distinct training and fine-tuning approaches.
- **Prompt-Based Metrics:** Ragas Relevance and similarity perform poorly, often assigning high scores indiscriminately to both correct and incorrect answers, thus failing to distinguish effectively between them. Ragas Correctness shows potential with a descending score trend, yet the small difference between correct and incorrect answers makes it prone to misleading evaluations. Tonic similarity aligns with the pattern observed in lexical metrics, lacking strong differentiation capabilities. The top-performing metrics in this group are **Prometheus, Ragas Faithfulness, Tonic Consistency, and LLMScore,** which clearly distinguish correct answers by consistently assigning significantly higher scores to them while scoring incorrect answers much lower. This behavior is the most desirable in evaluation metrics, as it effectively separates correct from incorrect responses, demonstrating superior utility and reliability.

2.3 Experiment 3

Do the metrics align with human preferences? Figure 29 presents the results of Experiment 3, where we *compared the performance of the metrics against human preference rankings for a random sample of 25 question-and-answer pairs* (4 from QASPER, 12 from QASA and 9 from a property dataset from Iris.AI). These answers were generated by six different models (Gemma 2 9B, Gemma 2 27B, LLAMA 3.1 8B, LLAMA 3.1 70B, Mistral 7b, and Phi3), the original reference answer from the corresponding dataset was also included. Then the different answers were ranked by human annotators based on their preferences, forming a human judgment baseline. Metrics were computed on the same data, and we created a ranking from each one as well. Metrics' and humans' ranks were then compared.

The chart visualizes the rankings, with the first column showing human rankings from 1st to 7th position for each answer. The subsequent columns display the rankings assigned by different evaluation metrics, allowing us to assess how well each metric's ranking aligns with human preferences. Colors in the chart represent answers generated by different models, including the reference answer.

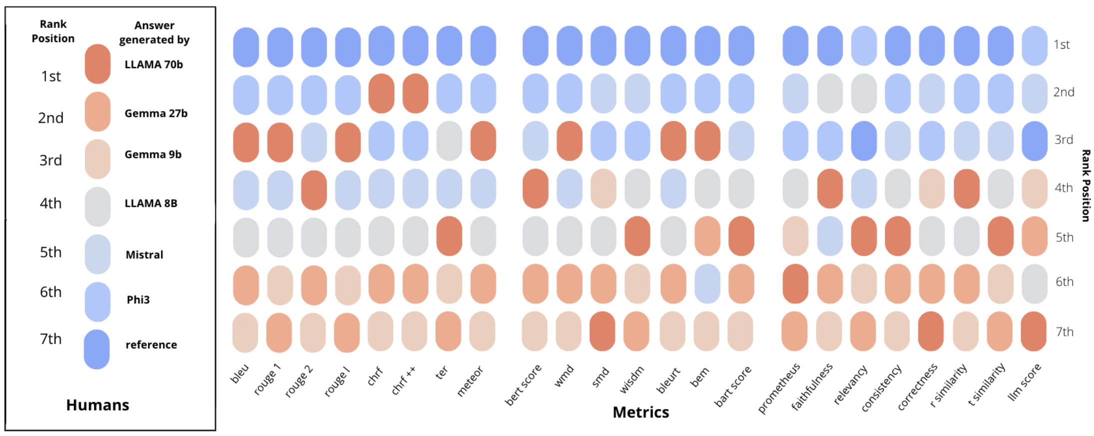

Figure 29: **Human and Metrics Ranks on Survey.** This graph compares the ranks assigned by the metrics to those assigned by human annotators for a subsample of 25 QA pairs. The answers were generated by six different models. A reference answer was also included. The first column represents the ranks given by human, while the remaining columns show the ranks assigned by metrics. Each color corresponds to a specific model, and the rows represent rank positions from 1st to 7th.

As shown in the graph, most metrics ranked the reference answers (the original ones from the datasets) in the highest position, with two notable exceptions: Ragas Relevance and LLMScore, which placed the reference in third. This was expected, as most metrics rely on the reference answer for evaluation, making it the natural top choice. The exception in Ragas Relevance is understandable since it is one of the few metrics that does not depend on the reference answer, instead evaluating based on the question and candidate answer alone. While Ragas Faithfulness could have exhibited a similar behavior, it did not. A similar outcome was observed with the LLMScore. Although the metric takes a reference answer as input, the prompt used in this specific experiment did not account for it. As a result, the model was not explicitly instructed to prioritize aligning with the reference answer (see Appendix M.1).

Many metrics ranked answers from the Mistral and Phi3 models in second and third positions, except for ChrF and ChrF++, which placed LLAMA 3.1 70b higher (as humans did), and Ragas Faithfulness and Relevancy, which favored LLAMA 3.1 8b. For the LLAMA 3.1 model, the 8b version was often ranked 4th or 5th by most metrics, while the 70b version performed better in lexical and semantic similarity metrics but ranked lower with LLM-based metrics. Notably, Prometheus and LLMScore did not exhibit a bias towards the LLAMA family models, including LLAMA 3.1 70B, the model upon which these metrics are based. This alleviated concerns about potential bias, confirming that these metrics fairly assess answers without favoring the model used to compute the score. Finally, both versions of Gemma, however, consistently placed at the bottom of most rankings, with few exceptions: SMS, Ragas Correctness, and LLMScore rated the 9b version slightly higher.

Interestingly, **the metrics showed significant divergence from human judgment.** None of the metrics aligned with human preferences for the top or bottom ranks. In particular, no metric placed the top three or bottom two models in the same order as the human annotators. Some metrics did show agreement in the middle ranks: WISDM, BEM, Bart, Prometheus, Ragas Consistency, and Tonic similarity assigned the 4th position similarly to humans, while Ragas Faithfulness matched the human ranking for 5th.

What stands out even more is that the answer ranked highest by most metrics—**the reference answer—was actually placed at the bottom by human experts.** This is particularly notable because the foundational assumption behind all the metrics we evaluated is that the reference, being a human-produced answer, should be the gold standard for comparison. However, this finding suggests that a human-produced answer is not always the most preferred according to human judgment.

Interestingly, LLAMA 3.1 70B, the model rated highest by human evaluators, was ranked in the middle or lower positions by most metrics. Lexical and semantic similarity metrics awarded it relatively high rankings, while model-based metrics consistently placed it near the bottom. This result is noteworthy because, in the previous experiments, lexical similarity metrics had shown the least desirable performance. Yet, in this experiment, they demonstrated the strongest alignment with human judgments. Furthermore, when disregarding the reference answer (since these metrics tend to favor it by default), it becomes evident that this group of metrics positioned LLAMA 3.1 70B—humans' preferred model—at the top. This highlights their unexpected reliability in reflecting human preferences.

This clear divergence is significant: the answers that metrics rated highly, such as the reference and Phi3, were at the bottom of human preference, while the models that metrics ranked lowest, like Gemma, were favored by humans. Metrics and human preferences only showed partial agreement for the Mistral and LLAMA 8b models, which consistently placed around the middle of the rankings.

To conduct a more detailed analysis, we calculated the Spearman correlation between the rankings assigned by each metric and those assigned by human experts, as presented in Table 11. The results show correlations ranging from moderately negative (between -0,4 and -0,7) to strongly negative (below -0,7), with many correlations being statistically significant (p-value $<$ 0,05). Notably, the metrics exhibiting the strongest negative correlations—ROUGE-2, METEOR, BERT, SMS, WISDM, BARTScore, and all prompt-based metrics except LLMScore—were also those with statistical significance. In overall, **all metrics demonstrated negative correlations with human rankings, indicating poor alignment.**

Although in general the analysis revealed that automatic evaluation metrics generally do not align well with human judgments, somo specifications and distinctions can be made.

Firstly, most lexical similarity and learned metrics showed moderate negative correlations, which were often not statistically significant. This suggests these metrics are insufficient in capturing human preferences, but, at the least, in this experiment they got the least undesirable performance. Conversely, semantic similarity and prompt-based metrics displayed the strongest negative correlations, with statistically significant p-values (below 0,05). This indicates a significant inverse relationship between these metrics' rankings and human preferences, meaning they tend to rank answers in opposition to human experts' evaluations, which for this occasion was the most undesirable.

Interestingly, even state-of-the-art metrics, including some of the most advanced methods, showed high negative correlations with human rankings. This highlights that even the most recent metrics may diverge significantly from human judgments. This finding suggest that using answers produced by humans as reference might not be the optimal approach, as they might diverge from answers actually preferred by humans' judgement.

Category	**Metric**	**Spearman Correlation**
Lexical Similarity	BLEU	-0.60
	ROUGE-1	-0.64
	ROUGE-2	-0.75*
	ROUGE-L	-0.64
	ChrF	-0.42
	ChrF++	-0.42
	METEOR	-0.85*
	TER	-0.60
Semantic Similarity	BERTScore	-0.75*
	WMS	-0.60
	SMS	-0.92*
	WISDM	-0.85*
LLM based Fine-tuned	BLEURT	-0.60
	BEM	-0.46
	BARTScore	-0.92*
LLM based Prompt	Prometheus	-0.85*
	LLMScore	-0.60
	Faithfulness	-0.71
	Relevancy	-0.85*
	Correctness	-0.92*
	R similarity	-0.75*
	Consistency	-0.82*
	T similarity	-0.78*

Table 11: **Correlation between human rank and metric ranks.** This table shows the Spearman correlation coefficients between answers' rank generated by human experts and answers' ranks generated by the different metrics. Only significant correlations ($p < 0.05$) are colored, the darker colors show higher values, therefore the strongest correlation.

Discussion

The primary goal of this thesis was to analyze the *effectiveness of various evaluation metrics in assessing RAG systems*, particularly within the domain of a scientific QA tasks. We classified the metrics into three categories: lexical similarity, semantic similarity, and model-based (see Section 0.2), and designed a series of experiments of increasing complexity to evaluate their performance. The results were thoroughly analyzed in the previous chapter. Building on the findings, we will now engage in assessing which evaluation methodology appears most suitable for measuring the performance of RAG systems in answering scientific questions derived from research papers.

Figure 30: **Overview of metrics' performance.** Look at the reference for more detailed explanation. The strength of the color indicates better performance.

Figure 30 shows a summary of the performance of all metrics on different experiments. To decide if a metric performance was above expectation, desirable, or non-desirable, we took in consideration the criteria already explained in the previous chapter and the analysis provided there. Based on these findings, we can now revisit our research questions:

- **Can the metrics distinguish between "correct" and "incorrect" answers?** When presented with the simplest task of distinguishing between correct and incorrect answers, under the conditions already explained and taking into account win rate, Cohen's d and sensitivity index, all metrics could do it. However, it was noticeable that as the metrics were more advanced and elaborated, they performed better, with a stronger and more reliable distinction among conditions. That is, lexical similarity metrics showed the poorest differentiation between correct and incorrect, and model based the best one. In general, semantic similarity and prompt based metrics showed the best performance as a group. Specifically, **LLM Score and Prometheus** appeared to be the best metrics at this task.
- **Can the metrics reliably distinguish between "better" and "worse" answers?** Since all metrics failed in the initial analysis by assigning higher scores to the answers of the smaller model, the results from the second analysis will address this question. The findings suggest that most lexical and semantic similarity metrics are overly rigid, heavily reliant on the reference answer. As a result, when these metrics encounter correct answers that differ in form or content from the ground truth, or incorrect answers that closely resemble it, they are often misled. In contrast, the metrics that performed best in this task were the model-based metrics, particularly **LLM Score, Tonic Similarity, Ragas Similarity, and Prometheus**. These metrics demonstrated greater robustness in evaluating answers more effectively, irrespective of misleading similarities to the reference answer.
- **Do the metrics align with human preferences?** No metric aligned perfectly with human preferences, as reflected in the ranking generated by expert annotator scores. However, if we were to select a group of metrics for tasks where alignment with human judgment is crucial, the findings suggest that lexical similarity metrics, despite their limitations, showed the smallest—albeit not statistically significant—correlation. Notably, these metrics managed to rank the human-preferred answer higher, with **ChrF** on its two versions performing best in this regard.

Based on the analysis of all experiments, the metric that demonstrated the best overall performance was **LLM-Score**, derived from a straightforward prompting of the LLAMA 3.1 70B model, as detailed in Section 1.2. It effectively differentiated between correct and incorrect answers, showed strong coverage and reliability, and consistently assigned higher scores to correct answers despite misleading form or content modifications. Although its correlation with human judgments was moderately negative, it was not statistically significant. Following closely was the **Prometheus** metric, which also involved model prompting but used a more specific scoring rubric. Prometheus matched LLM-Score in most areas and even surpassed it in coverage. However, it exhibited a significant negative correlation with human judgments.

Overall, these metrics seem to offer the most effective approach for assessing RAG systems, though their alignment with human judgment remains an area for further exploration. It is also important to note that, although these more advanced metrics tend to perform better, they are considerably more resource-intensive in terms of time and computational requirements. In contrast, simpler metrics, such as those based on lexical and semantic similarity, are significantly easier to implement, faster to compute, and typically free. Despite it was not covered on this work, this trade-off should be carefully considered when selecting metrics for practical applications.

3.1 Analysis per category

One of the motivations behind this thesis was the desire to understand the strengths and weaknesses of different evaluation methodologies. Based on the findings of each experiment, we will provide a comprehensive discussion of the performance of each category of metrics, exploring their respective strengths and limitations:

- **Lexical Similarity metrics:**

1. Lexical similarity metrics demonstrated some desirable properties, though their overall performance in distinguishing correct from incorrect answers was poor. While these metrics generally assigned marginally higher scores to correct answers (both in the "aligned vs. shuffled" test and the "sensitivity index"), the effect sizes and index values indicated a weak distinction between correct and incorrect answers. This suggests a limited capacity to reliably differentiate between them.

2. These metrics also exhibited high sensitivity to superficial variations between the generated answers and the reference ground truth. They frequently assigned high scores to incorrect answers that closely resembled the ground truth and penalized correct answers that deviated in form. While this sensitivity is not ideal for evaluating Retrieval-Augmented Generation (RAG) systems in scientific question-answering (QA) tasks, it may be useful in other specific applications.

3. Interestingly, when comparing the rankings produced by these metrics with human judgments, the correlation was negative (as observed for all metrics), with the lexical similarity group showing the lowest correlation, which was statistically non-significant. This outcome was unexpected, given that this group had performed poorly in prior experiments but exhibited the most favorable result in this particular comparison. This finding warrants further investigation to better understand the reasons behind this behavior.

4. Among this group, **METEOR** stood out as the most refined metric, showing better distinction between correct and incorrect answers and a better sensitivity pattern, indicating greater flexibility to answer variations. Nevertheless, it was sometimes misled by incorrect answers that resembled the ground truth.

- **Semantic Similarity metrics:**

1. This group of metrics performed *noticeably better than the lexical similarity metrics* in terms of differentiating between correct and incorrect answers. They achieved higher win rates and displayed a stronger, though still moderate, effect size and sensitivity index.

2. In terms of sensitivity, all metrics in this group exhibited a descending slope, indicating they were still somewhat misled by incorrect but semantically similar answers. Therefor, while these metrics showed improvement over lexical similarity metrics, their ability to distinguish more complex answers remained limited, which is again not ideal for evaluating RAG systems in scientific QA tasks.

3. Furthermore, their correlation with human judgments was negatively stronger and statistically significant, suggesting that while these metrics were useful, they didn't always align with human evaluations.

4. In this group, the performance of all metrics was relatively similar.

- **Model learned based metrics:**

1. Learned metrics behaved in a way that was similar to semantic similarity metrics, managing to differentiate between correct and incorrect answers with a moderate effect size and sensitivity.

2. The sensitivity pattern of learned metrics was comparable to that of semantic similarity metrics, as they were still susceptible to being misled by incorrect but superficially correct variations of the ground truth.

3. The correlation with human judgment for this group was closer to that of lexical similarity metrics—negative but moderate, and not statistically significant. BARTScore was an exception in this group, as it behaved more like lexical similarity metrics, showing suboptimal performance and struggling to make meaningful distinctions.

4. Among this group, the metric with the best performance in overall was **BLEURT**.

- **Model prompt based metrics:**

1. This group showed the best performance in distinguishing between correct and incorrect answers, with effect sizes ranging from moderate to large (except for Ragas Faithfulness) and the largest sensitivity index values. Therefor, this group of metrics proved to be the best on this task.

2. Model-based metrics were the only group where some metrics achieved the ideal sensitivity pattern, scoring all correct answers higher and all incorrect ones lower, indicating they could be particularly useful and powerful. This would make them the best evaluation approach to assess the performance of a RAG system in scientific QA task. An exception was Tonic Consistency, which showed high sensitivity to superficial variations from the ground truth, similar to lexical similarity metrics. Ragas Relevancy, Correctness, and Similarity also displayed unusual behavior; although they followed the expected slope, they generally assigned high values across the board.

3. Surprisingly, this group also exhibited the strongest and most significant negative correlation with human judgment, underscoring their alignment with more objective assessments. The exception was the LLM Score, which had a moderate and non-significant correlation. This is not a desirable behavior on an evaluation metric, and further analysis should be done regarding this.

4. Among this group, the metrics with the best performance were **LLMScore and Prometheus**.

In summary, the findings of this thesis indicate that no single metric demonstrated consistently optimal performance across all evaluated scenarios. Instead, metrics from different categories—lexical, semantic, and model-based—showed varying strengths and weaknesses depending on the specific task or evaluation context. This outcome aligns with the insights from the literature, which highlight the limitations inherent in each type of metric. Despite these limitations, metrics continue to be employed in complementary ways, leveraging their unique advantages to provide a more holistic assessment. The diverse performance of these metrics underscores the importance of a multifaceted evaluation approach, especially when assessing complex systems like RAG in scientific QA tasks. This study reaffirms the need for continued development and refinement of evaluation metrics to address specific challenges.

Conclusions

The primary aim of this thesis was to find a metric to assess the performance of a RAG system within the context of scientific QA tasks. Our objective was to identify automatic evaluation metrics capable of assessing the correctness of generated answers, taking as reference human preferences, and analyze and compare their behavior. In the literature, we identified three main approaches to evaluation, each employing human-written answers as ground truth. Comparison with generated answers could be made based on form (lexical similarity), content (semantic similarity), or utilizing Large Language Models, either fine-tuning or directly prompting them.

While these approaches have well-documented advantages and limitations, they are still commonly used, often in combination, to leverage their respective strengths. Therefore, we understood the necessity to investigate their reliability and determine whether they are suitable for evaluating RAG system performance in QA tasks. To achieve this, we designed a set of experiments to address the following questions: *1. Can the metrics distinguish between correct and incorrect answers? 2. Can the metrics differentiate between "better" and "worse" answers? 3. How well do these metrics align with human preferences?* Through these experiments, we explored the metrics' ability to differentiate between varying degrees of correctness, as well as their potential to be misled by incorrect answers that appear superficially correct. Moreover, we assessed the degree of alignment between metric outputs and human evaluative preferences.

The results of our experiments revealed a notable disparity between the anticipated performance of current evaluation metrics and their actual effectiveness. While all metrics performed adequately in distinguishing between clearly correct and incorrect answers, model-based metrics demonstrated superior performance, followed by semantic similarity metrics and lastly, lexical similarity metrics. However, when the task became more challenging—such as distinguishing between better and worse correct answers, especially in complex scenarios involving misleading answers with variations in form and content—many metrics encountered difficulties. Lexical and semantic similarity metrics, in particular, proved to be rigid and highly dependent on ground truth, making them susceptible to manipulation by surface-level differences in form or content. In contrast, model-based metrics exhibited greater robustness in handling these variations. Nonetheless, in our third experiment, we observed an unexpected negative correlation between the rankings produced by the metrics and those determined by human experts. Strikingly, the most advanced model-based metrics showed the strongest negative correlation with human judgments, followed by semantic similarity metrics. Interestingly, lexical similarity metrics did not exhibit a significant negative correlation. This finding suggests that even the most sophisticated methods frequently fail to align with human preferences, highlighting a critical challenge.

In conclusion, the findings from the experiments suggest that no single metric demonstrated optimal performance across all evaluation scenarios. Instead, metrics from different categories — lexical, semantic, and model-based—demonstrated variable strengths and weaknesses depending on the task or context in which they were applied.

This aligns with insights from the literature that emphasize the limitations inherent in each type of metric. Despite these shortcomings, metrics continue to be used in complementary ways, with their unique advantages contributing to a more holistic evaluation. These findings underscore the importance of employing a multifaceted evaluation strategy, particularly when assessing the performance of complex systems like RAG in scientific QA tasks. This work reaffirms the need for further refinement of evaluation metrics to address specific challenges and encourages the use of multiple metrics to achieve more comprehensive and nuanced evaluations.

Reflecting on these results, it is clear that new evaluation approaches are necessary and further research is required to develop metrics that capture the true nuances of answer quality as perceived by humans. Notably, most of the metrics evaluated in this study were benchmarked against human-generated reference answers, which may inherently limit their ability to fully capture the subtleties of human judgment. This became particularly evident in Experiment 3, where human-generated answers, traditionally considered the gold standard, were poorly ranked by human annotators.

Our findings challenge the assumption that human-written reference answers serve as the optimal standard for evaluation. Alternatively, other approaches can be explored. On one hand, the question or context itself could serve as a reference (as attempted by metrics like RAGAS and Tonic), though their performance has been inconsistent, and further research is needed. On the other hand, human judgment may provide a more reliable basis for evaluating answer quality. The top-performing metrics in our experiments—such as LLM-Score and Prometheus—utilized model prompting to assess the contextual quality of answers, rather than relying solely on direct comparisons with reference texts.

These insights suggest a pivotal shift in QA evaluation, moving beyond static reference answers toward metrics that directly integrate human preferences and judgments. Future research should explore models that simulate human preferences, such as pairwise comparison models fine-tuned on human feedback, as it is done in Reinforcement Learning from Human Feedback [21]. Such approaches could offer more dynamic and adaptable evaluations, ensuring closer alignment between model performance and human expectations.

The quest for effective evaluation metrics in QA is complex, and this thesis critically assesses current approaches, highlighting both strengths and limitations. Recognizing that human judgment, rather than static references, should serve as the benchmark for quality represents a potential paradigm shift in how AI-generated content is evaluated. Adopting this shift will be crucial for advancing the field and ensuring AI systems are held to the evolving standards of human expectations.

In conclusion, this thesis offers several key insights regarding the evaluation of QA systems, particularly in the context of scientific tasks: 1. model-based metrics have shown the most promise in assessing generated answers, as they successfully overcome the limitations of traditional similarity-based metrics that rely heavily on a single reference text. This is important because there can be many valid variations of a correct answer, and model-based approaches capture this nuance; 2. the persistent gap between metric performance and human judgment highlights the need for continued research. Metrics must evolve to better capture the subtleties of human preferences and evaluative criteria; 3. a paradigm shift may be required, moving away from reference-based evaluations and toward methods that incorporate human preferences, such as pairwise comparison models fine-tuned on human judgment. This shift could enable the development of more adaptable, robust, and trustworthy evaluation frameworks, ensuring that QA systems align with the complexity and variability of human expectations.

4.1 Limitations and Future Directions

This thesis presents several limitations that warrant acknowledgment, and also highlights promising directions for future research. First, the study did not encompass the full spectrum of evaluation metrics available in the field. The selection of metrics was constrained by practical considerations, such as accessibility, ease of implementation, and time constraints. As a result, some metrics that could have provided valuable insights were not included, potentially limiting the scope of the analysis. Moreover, the datasets employed in this research may not fully capture the complexity and diversity of real-world scientific QA scenarios. During the analysis, certain data points did not sufficiently represent the nuanced challenges inherent in these contexts, potentially impacting the robustness of the findings R.2. This reliance on a limited dataset raises concerns about the generalizability of the results.

Additionally, the study primarily focused on two models within the same family—LLAMA. Expanding the analysis to include a broader range of models from different architectures could provide insights into whether the findings generalize across diverse model families. Such comparisons are crucial for determining whether evaluation metrics behave consistently across various model types, which remains a key area for future research. In Experiment 2, we also encountered a methodological limitation: we evaluated the performance of metrics based on model-generated answers, introducing a circular challenge. Although we manually inspected the generated answers to mitigate potential inaccuracies, this limitation persisted. Furthermore, some metrics relying on language models were implemented using the same LLAMA models being assessed, which could have introduced bias, though Experiment 3 addressed this concern to some extent.

Another area of improvement involves the survey analysis, where the guidelines provided to human annotators for evaluating answers were somewhat general. A more detailed and specific instruction set could yield more consistent and reliable human annotations.

The findings of this thesis underscore the need for future research to focus on evaluation metrics that more directly capture human preferences, addressing the limitations of static reference-based comparisons. Current practices that rely on predefined reference answers often fail to align with human judgment, as evidenced by the negative correlation between certain automatic metrics and human evaluations observed in our experiments. Future research should prioritize the refinement of existing model-based evaluation methods or explore more ambitious approaches, such as developing novel metrics that learn directly from human rankings and preferences.

Promising approaches could include fine-tuning models on datasets curated to reflect human preferences or developing interactive evaluation systems that incorporate real-time human feedback during assessments. These systems should aim to create adaptive, human-aligned metrics that dynamically respond to the complexities and nuances inherent in human judgment, setting a new standard for quality assessment in QA systems and beyond.

Moreover, several unexplored avenues remain that could contribute valuable insights. While this study focused solely on the generative component of retrieval-augmented generation (RAG) systems, future research should also evaluate the impact of the retrieval component on overall system performance. Additionally, in Experiment 3, incorporating deliberately incorrect answers could help assess whether human evaluators can distinguish these errors more effectively than the metrics, addressing the vulnerability to misleading answers seen in Experiment 2.

In conclusion, we hope that this research has contributed to a deeper understanding of evaluation metrics for generated text in the context of question answering (QA). We aspire for this work to serve as a foundation for the development of more comprehensive and generalizable frameworks for evaluating Retrieval-Augmented Generation (RAG) systems in QA. By advancing this area, future efforts can focus on creating metrics that better align with human judgment, ultimately driving the field toward more robust and reliable evaluation methods.

Bibliography

[1] Marah Abdin et al. “Phi-3 technical report: A highly capable language model locally on your phone”. In: *arXiv preprint arXiv:2404.14219* (2024).

[2] Josh Achiam et al. “Gpt-4 technical report”. In: *arXiv preprint arXiv:2303.08774* (2023).

[3] Vaibhav Adlakha et al. “Evaluating correctness and faithfulness of instruction-following models for question answering”. In: *arXiv preprint arXiv:2307.16877* (2023).

[4] Nimet Aksoy, Zekeriya Anıl Güven, and Murat Osman Ünalır. “Architecting and Evaluating a RAG based Question Answering System for SQuAD Dataset”. In: *ICAMSigma’24* (2024), p. 213.

[5] Ali Mohamed Nabil Allam and Mohamed Hassan Haggag. “The question answering systems: A survey”. In: *International Journal of Research and Reviews in Information Sciences (IJRRIS)* 2.3 (2012).

[6] Chenxin An et al. “L-eval: Instituting standardized evaluation for long context language models”. In: *arXiv preprint arXiv:2307.11088* (2023).

[7] Simone Balloccu et al. “Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms”. In: *arXiv preprint arXiv:2402.03927* (2024).

[8] Satanjeev Banerjee and Alon Lavie. “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments”. In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005, pp. 65–72.

[9] Anna Bavaresco et al. “LLMs instead of Human Judges? A Large Scale Empirical Study across 20 NLP Evaluation Tasks”. In: *arXiv preprint arXiv:2406.18403* (2024).

[10] Stella Biderman et al. “Lessons from the Trenches on Reproducible Evaluation of Language Models”. In: *arXiv preprint arXiv:2405.14782* (2024).

[11] Viktor Botev, Kaloyan Marinov, and Florian Schäfer. “Word importance-based similarity of documents metric (WISDM) Fast and scalable document similarity metric for analysis of scientific documents”. In: *Proceedings of the 6th international workshop on mining scientific publications*. 2017, pp. 17–23.

[12] Tom B Brown et al. “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1877–1901.

[13] Jannis Bulian et al. “Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation”. In: *arXiv preprint arXiv:2202.07654* (2022).

[14] Michael Caballero. “A brief survey of question answering systems”. In: *International Journal of Artificial Intelligence & Applications (IJAIA)* 12.5 (2021).

[15] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. "Evaluation of text generation: A survey". In: *arXiv preprint arXiv:2006.14799* (2020).

[16] Yupeng Chang et al. "A survey on evaluation of large language models". In: *ACM Transactions on Intelligent Systems and Technology* 15.3 (2024), pp. 1–45.

[17] Anthony Chen et al. "Evaluating question answering evaluation". In: *Proceedings of the 2nd workshop on machine reading for question answering*. 2019, pp. 119–124.

[18] Steffi Chern et al. "Can large language models be trusted for evaluation? scalable meta-evaluation of llms as evaluators via agent debate". In: *arXiv preprint arXiv:2401.16788* (2024).

[19] Cheng-Han Chiang and Hung-yi Lee. "Can large language models be an alternative to human evaluations?" In: *arXiv preprint arXiv:2305.01937* (2023).

[20] Sabrina Chiesurin et al. "The Dangers of trusting Stochastic Parrots: Faithfulness and Trust in Open-domain Conversational Question Answering. arXiv 2023". In: *arXiv preprint arXiv:2305.16519* ().

[21] Paul F Christiano et al. "Deep reinforcement learning from human preferences". In: *Advances in neural information processing systems* 30 (2017).

[22] Elizabeth Clark, Asli Celikyilmaz, and Noah A Smith. "Sentence mover's similarity: Automatic evaluation for multi-sentence texts". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 2748–2760.

[23] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. routledge, 2013.

[24] Eduardo Gabriel Côrtes. "Beyond accuracy: completeness and relevance metrics for evaluating long answers". In: (2024).

[25] Pradeep Dasigi et al. "A dataset of information-seeking questions and answers anchored in research papers". In: *arXiv preprint arXiv:2105.03011* (2021).

[26] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019, pp. 4171–4186.

[27] Yuyao Duan and Vilgot Lundborg. *A Method for Automated Assessment of Large Language Model Chatbots: Exploring LLM-as-a-Judge in Educational Question-Answering Tasks*. 2024.

[28] Abhimanyu Dubey et al. "The llama 3 herd of models". In: *arXiv preprint arXiv:2407.21783* (2024).

[29] Shahul Es et al. "Ragas: Automated evaluation of retrieval augmented generation". In: *arXiv preprint arXiv:2309.15217* (2023).

[30] Jinlan Fu et al. "Gptscore: Evaluate as you desire". In: *arXiv preprint arXiv:2302.04166* (2023).

[31] Matteo Gabburo et al. "Knowledge transfer from answer ranking to answer generation". In: *arXiv preprint arXiv:2210.12865* (2022).

[32] Mingqi Gao et al. "Llm-based nlg evaluation: Current status and challenges". In: *arXiv preprint arXiv:2402.01383* (2024).

[33] Yunfan Gao et al. "Retrieval-augmented generation for large language models: A survey". In: *arXiv preprint arXiv:2312.10997* (2023).

[34] Cristina Garbacea et al. “Judge the judges: A large-scale evaluation study of neural language models for online review generation”. In: *arXiv preprint arXiv:1901.00398* (2019).

[35] Hamed Babaei Giglou et al. “Scholarly Question Answering using Large Language Models in the NFDI4DataScience Gateway”. In: *arXiv preprint arXiv:2406.07257* (2024).

[36] Zishan Guo et al. “Evaluating large language models: A comprehensive survey”. In: *arXiv preprint arXiv:2310.19736* (2023).

[37] Kelvin Guu et al. “Retrieval augmented language model pre-training”. In: *International conference on machine learning*. PMLR. 2020, pp. 3929–3938.

[38] Rujun Han et al. “RAG-QA Arena: Evaluating Domain Robustness for Long-form Retrieval Augmented Question Answering”. In: *arXiv preprint arXiv:2407.13998* (2024).

[39] Chao-Chun Hsu et al. “Answer generation for retrieval-based question answering systems”. In: *arXiv preprint arXiv:2106.00955* (2021).

[40] Taojun Hu and Xiao-Hua Zhou. “Unveiling LLM Evaluation Focused on Metrics: Challenges and Solutions”. In: *arXiv preprint arXiv:2404.09135* (2024).

[41] Hui Huang et al. “An empirical study of llm-as-a-judge for llm evaluation: Fine-tuned judge models are task-specific classifiers”. In: *arXiv preprint arXiv:2403.02839* (2024).

[42] Gautier Izacard et al. “Atlas: Few-shot learning with retrieval augmented language models”. In: *Journal of Machine Learning Research* 24.251 (2023), pp. 1–43.

[43] Ehsan Kamalloo et al. “Evaluating open-domain question answering in the era of large language models”. In: *arXiv preprint arXiv:2305.06984* (2023).

[44] Jared Kaplan et al. “Scaling laws for neural language models”. In: *arXiv preprint arXiv:2001.08361* (2020).

[45] Vladimir Karpukhin et al. “Dense passage retrieval for open-domain question answering”. In: *arXiv preprint arXiv:2004.04906* (2020).

[46] Seungone Kim et al. “Prometheus: Inducing fine-grained evaluation capability in language models”. In: *The Twelfth International Conference on Learning Representations*. 2023.

[47] Matt Kusner et al. “From word embeddings to document distances”. In: *International conference on machine learning*. PMLR. 2015, pp. 957–966.

[48] Jakub Lála et al. “Paperqa: Retrieval-augmented generative agent for scientific research”. In: *arXiv preprint arXiv:2312.07559* (2023).

[49] Md Tahmid Rahman Laskar et al. “A Systematic Survey and Critical Review on Evaluating Large Language Models: Challenges, Limitations, and Recommendations”. In: *arXiv preprint arXiv:2407.04069* (2024).

[50] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. “Latent retrieval for weakly supervised open domain question answering”. In: *arXiv preprint arXiv:1906.00300* (2019).

[51] Yoonjoo Lee et al. “QASA: advanced question answering on scientific articles”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 19036–19052.

[52] Mike Lewis et al. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *arXiv preprint arXiv:1910.13461* (2019).

[53] Patrick Lewis et al. “Retrieval-augmented generation for knowledge-intensive nlp tasks”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9459–9474.

[54] Zhen Li et al. "Leveraging Large Language Models for NLG Evaluation: Advances and Challenges". In: (2024).

[55] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[56] Stephanie Lin, Jacob Hilton, and Owain Evans. "Truthfulqa: Measuring how models mimic human falsehoods". In: *arXiv preprint arXiv:2109.07958* (2021).

[57] Chia-Wei Liu et al. "How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation". In: *arXiv preprint arXiv:1603.08023* (2016).

[58] Yang Liu et al. "G-eval: Nlg evaluation using gpt-4 with better human alignment". In: *arXiv preprint arXiv:2303.16634* (2023).

[59] Yang Liu et al. "Trustworthy LLMs: A survey and guideline for evaluating large language models' alignment". In: *arXiv preprint arXiv:2308.05374* (2023).

[60] Yuxuan Liu et al. "Calibrating llm-based evaluator". In: *arXiv preprint arXiv:2309.13308* (2023).

[61] Larry Medsker and Lakhmi C Jain. *Recurrent neural networks: design and applications*. CRC press, 1999.

[62] Tomas Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *arXiv preprint arXiv:1301.3781* (2013).

[63] Shervin Minaee et al. "Large language models: A survey". In: *arXiv preprint arXiv:2402.06196* (2024).

[64] Shervin Minaee et al. "Large language models: A survey". In: *arXiv preprint arXiv:2402.06196* (2024).

[65] Amit Mishra and Sanjay Kumar Jain. "A survey on question answering systems with classification". In: *Journal of King Saud University-Computer and Information Sciences* 28.3 (2016), pp. 345–361.

[66] Benjamin Muller et al. "Cross-lingual open-domain question answering with answer sentence generation". In: *arXiv preprint arXiv:2110.07150* (2021).

[67] Sai Munikoti et al. "Evaluating the Effectiveness of Retrieval-Augmented Large Language Models in Scientific Document Reasoning". In: *arXiv preprint arXiv:2311.04348* (2023).

[68] Preksha Nema and Mitesh M Khapra. "Towards a better metric for evaluating question generation systems". In: *arXiv preprint arXiv:1808.10192* (2018).

[69] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[70] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[71] Gabriel Peyré, Marco Cuturi, et al. "Computational optimal transport: With applications to data science". In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.

[72] Maja Popović. "chrF: character n-gram F-score for automatic MT evaluation". In: *Proceedings of the tenth workshop on statistical machine translation*. 2015, pp. 392–395.

[73] Maja Popović. “chrF++: words helping character n-grams”. In: *Proceedings of the second conference on machine translation*. 2017, pp. 612–618.

[74] Lawrence Rabiner and Biinghwang Juang. “An introduction to hidden Markov models”. In: *ieee assp magazine* 3.1 (1986), pp. 4–16.

[75] Alec Radford et al. “Improving Language Understanding by Generative Pre-Training”. In: *OpenAI* (2018).

[76] N Reimers. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *arXiv preprint arXiv:1908.10084* (2019).

[77] Paul Robert and Yves Escoufier. “A unifying tool for linear multivariate statistical methods: the RV-coefficient”. In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 25.3 (1976), pp. 257–265.

[78] Gerard Salton and Christopher Buckley. “Term-weighting approaches in automatic text retrieval”. In: *Information Processing & Management* 24.5 (1988), pp. 513–523.

[79] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. “BLEURT: Learning robust metrics for text generation”. In: *arXiv preprint arXiv:2004.04696* (2020).

[80] Thibault Sellam et al. “Learning to evaluate translation beyond English: BLEURT submissions to the WMT metrics 2020 shared task”. In: *arXiv preprint arXiv:2010.04297* (2020).

[81] Chenglei Si, Chen Zhao, and Jordan Boyd-Graber. “What’s in a name? answer equivalence for open-domain question answering”. In: *arXiv preprint arXiv:2109.05289* (2021).

[82] Matthew Snover et al. “A study of translation edit rate with targeted human annotation”. In: *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. 2006, pp. 223–231.

[83] Andrea Sottana et al. “Evaluation metrics in the era of GPT-4: reliably evaluating large language models on sequence to sequence tasks”. In: *arXiv preprint arXiv:2310.13800* (2023).

[84] C Spearman. “The american journal of psychology”. In: *Am. J. Psychol* 15 (1904), p. 88.

[85] Akchay Srivastava and Atif Memon. “Towards Robust Evaluation: A Comprehensive Taxonomy of Datasets and Metrics for Open Domain Question Answering in the Era of Large Language Models”. In: *IEEE Access* (2024).

[86] Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. “Large language models are inconsistent and biased evaluators”. In: *arXiv preprint arXiv:2405.01724* (2024).

[87] Prakash Chandra Sukhwal, Atreyi Kankanhalli, and Vaibhav Rajan. “Evaluation Dimensions for Assessing Question Answer Systems for Lay Users: The Case of DiseaseGuru”. In: *Proceedings of the AAAI Symposium Series*. Vol. 1. 1. 2023, pp. 98–102.

[88] Tianxiang Sun et al. “BERTScore is unfair: On social bias in language model-based metrics for text generation”. In: *arXiv preprint arXiv:2210.07626* (2022).

[89] Gemma Team et al. “Gemma 2: Improving open language models at a practical size”. In: *arXiv preprint arXiv:2408.00118* (2024).

[90] Hugo Touvron et al. “Llama 2: Open foundation and fine-tuned chat models”. In: *arXiv preprint arXiv:2307.09288* (2023).

[91] Hugo Touvron et al. “LLaMA: Open and Efficient Foundation Language Models”. In: *arXiv preprint arXiv:2302.13971* (2023).

[92] A Vaswani. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017).

[93] Cunxiang Wang et al. "Evaluating open-qa evaluation". In: *Advances in Neural Information Processing Systems* 36 (2024).

[94] Jiaan Wang et al. "Is chatgpt a good nlg evaluator? a preliminary study". In: *arXiv preprint arXiv:2303.04048* (2023).

[95] Jacob White. "PubMed 2.0". In: *Medical reference services quarterly* 39.4 (2020), pp. 382–387.

[96] Yi-Ting Yeh, Maxine Eskenazi, and Shikib Mehri. "A comprehensive assessment of dialog evaluation metrics". In: *arXiv preprint arXiv:2106.03706* (2021).

[97] Weizhe Yuan, Graham Neubig, and Pengfei Liu. "Bartscore: Evaluating generated text as text generation". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 27263–27277.

[98] Tianyi Zhang et al. "Bertscore: Evaluating text generation with bert". In: *arXiv preprint arXiv:1904.09675* (2019).

[99] Wayne Xin Zhao et al. "A survey of large language models". In: *arXiv preprint arXiv:2303.18223* (2023).

[100] Lianmin Zheng et al. "Judging llm-as-a-judge with mt-bench and chatbot arena". In: *Advances in Neural Information Processing Systems* 36 (2024).

[101] Zhuang Ziyu et al. "Through the lens of core competency: Survey on evaluation of large language models". In: *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 2: Frontier Forum)*. 2023, pp. 88–109.

Appendix Methodology

M.1 Evaluation Module (Metrics Implementation)

The evaluation module is designed to automatically compute the performance of specified metrics on each instance of a given dataset, generating detailed statistics and correlation values for these metrics. The development of this module prioritizes reproducibility, user-friendliness, and adaptability, making it suitable for diverse datasets and evaluation scenarios in both research and practical applications.

The module primarily consists of two main classes, **EvaluationPipeline** and **GenerationEvaluation**, which coordinate the evaluation process. These classes are supported by several data types, functions, and metric implementations to streamline the evaluation workflow.

The **EvaluationPipeline** class orchestrates the evaluation process by dynamically loading and configuring the specified metrics based on user-provided settings. This class is initialized with a configuration object (`EvaluationPipelineConfig`) that defines which metrics and environment variables are needed for the evaluation.

- Initialization: During the initialization, the module loads the specified metrics using their class names and configures them accordingly. Metrics are categorized as general metrics or those requiring a language model for evaluation.

- Metric Preparation: The `init` method prepares the metrics, setting up any necessary connections to language models when required. For instance, some metrics, like `ContextSensitivity`, are configured with a model and a prompt template, whereas simpler metrics are initialized directly.

- Running Evaluation: The `run` method evaluates the provided data using the initialized metrics, generating scores for each metric. The method handles exceptions gracefully, ensuring that missing data or evaluation errors do not interrupt the overall process.

The **GenerationEvaluation** class handles the core data processing and evaluation tasks, including reading input data, evaluating metrics, and saving the results. This class interacts with the dataset, converting it into a format suitable for evaluation and performing the computations.

- Data Processing: the `read_data` method converts input data into a list of `RAGExample` instances, which encapsulate each question, its associated context, the generated answer, and the correct answer. This structured approach ensures that all necessary information is available for evaluation.

- Input Handling: the `read_input` method supports reading data from CSV and JSON files, logging the process to maintain transparency and traceability. This flexibility allows users to work with different data formats, facilitating easy integration into various workflows.

- Data Evaluation: the `evaluate_data` method is the core evaluation component, computing metric scores for each data instance. It calculates various statistical values, such as mean, standard deviation, minimum, and maximum scores for each metric. Additionally, it computes correlations between the metrics using Spearman's rank correlation and agreement scores using Cohen's kappa, offering insights into the relationships and consistencies between different evaluation criteria. The method is designed to handle missing or invalid data gracefully, ensuring that the results are robust and reliable.

The module supports a wide range of metrics, each with specific implementation details. Table 12 summarizes the source and implementation approach for each metric:

Metric	Implementation Details
Bleu	SacreBLEU library available on PyPi.
Rouge	ROUGE library available on PyPi. We considered only the f-score.
Meteor	"meteor_score" function of the NLTK library, available on PyPi.
Chrf (++)	SacreBLEU library. For the chrF++ version, the word order was set to 2.
Ter	SacreBLEU library available on PyPi.
WMD and SMD	In the repository, both Mover's Distance metrics are implemented as follows. MoversDistance class is a general framework for computing the movers distance between two sets of strings. This process involves two primary steps: calculates a cost matrix representing the distance between each string in the first set and each string in the second set, utilizing Euclidean Distance; and uses the linear sum assignment algorithm from the Scipy library to find the optimal one-to-one mapping between elements of the two sets, minimizing the total cost. The class is designed to be extended and requires subclasses to implement two methods: get_embeddings (converts text into numerical embeddings), and parse_text (processes and tokenizes text). The WordMoversDistance and SentenceMoversDistance classes extend Movers Distance but differ in the embeddings they use: WMD embeddings for each token are obtained using a word analyzer; SMD embeddings for each sentence are obtained using a specified sentence transformer model. Finally, WordMoversSimilarity and SentenceMoversSimilarity are extensions of the respective distance classes designed to compute similarity scores instead of distance. The similarity score is simply one minus the distance score.
Wisdm	Internal implementation of Iris.AI company.
BertScore	It uses the SentenceTransformer model ("sentence-transformers/all-MiniLM-L6-v2") to obtain embeddings. Then, cosine similarity is calculated using the pytorch_cos_sim function.
Bleurt	BleuRT library available on GitHub.
BEM	BEM model is hosted on Kaggle, and the code to compute this score is based on the directions given in its documentation.
Prometheus	It uses the PrometheusEval class with the LiteLLM model initialized using groq/llama-3.1-70b-versatile. The specific criteria and score rubric is: criteria: Is the model proficient in generating a relevant, faithful, and complete answer to the question? score1_description: The generated answer is not relevant to the user query and reference answer. score2_description: The generated answer is similar to the reference answer but not relevant to the user query. score3_description: The generated answer is relevant to the user query and reference answer but contains mistakes. score4_description: The generated answer is relevant to the user query and is similar to the reference answer but is not as concise. score5_description: The generated answer is relevant to the user query and fully correct according to the reference answer.
BartScore	It uses the BART model ("facebook/bart-large-cnn") from the Transformers library.
LLMscore	class LLMscore allows choosing an OpenAI or Meta and prompt it to perform an evaluation. We used LLAMA 3.1 70 billion. The exact prompt for Experiment 1 was: "You will be provided with a question and a context from a scientific paper. Then, you will be given a reference right answer and a candidate answer. Your task is to evaluate how good the candidate answer is in relation to this question, taking into consideration the original question, the context, and the reference answer. 1. Question: {question} 2. Context: {context} 3. Reference answer: {sentence1} 4. Answer to evaluate: {sentence2} Output only a numerical value from 0 (poor answer) to 1 (excellent answer)" (for Experiment 3 reference was not provided
RAGAS	All the code used to compute these metrics comes from the ragas library available on GitHub.
TONIC	All the code used to compute these metrics comes from the tonic_validate library available on GitHub.

Table 12: **Metrics Implementation.** The table describes how the code to compute each of the metrics was obtained or developed.

M.2 Datasets Table

Table 13 shows a brief summary of information from different QA datasets that were considered to be used on this thesis.

M.3 Datasets Processing

The generation and processing of all the data is available in the repository on the folder "data", organized as follows:

"Qasa and Qasper" folder contains: 1. the original .json files; 2. the notebooks where data was load as pandas data frame, processed, and outputted as .csv file; 3. the notebooks where the .csv files were populated with answers generated by the LLAMA 2 model; 4. the notebooks where the .csv files were shuffled to pair questions with random answers; 5. the notebooks where .csv files were populated with the LLAMA 3.1 model with 8 and 70 billion parameters.

- **QASPER :** The data was downloaded as two .json files from their Hugging Face repository. Then, it was load and processed as follows: from the files were extracted the content of the keys "question" as the question, "free form answer" as the answer and "evidence" as the context; consequently, the final data frame from QASPER data finally had 968 instances; correct answers and contexts that were originally stored as lists were merged into one single answer and context string for each row; it was saved as qasper.csv file.
- **QASA:** The data was downloaded as a `.json` file from their GitHub repository. Then, it was loaded and processed as follows: unnecessary columns were removed (`'paper_id'`, `'title'`, `'question_id'`, `'question_section'`, `'question_trigger_sentence'`, `'arxiv_id'`, `'s2orc_url'`, `'arxiv_url'`, `'s2orc_id'`); columns `'composition'` and `'evidential_info'` were renamed as `'correct_answer'` and `'context'` for clarity; the contexts, which were saved as a dictionary with multiple pieces of evidential information, were unified into a single string for each row; in the `'question_type'` column, values' names were unified to `'complex question'`, `'shallow question'`, and `'testing question'`; lines with missing information (mainly where no context was provided) were removed. In conclusion, the final data frame included 1543 examples (589 complex, 486 shallow, and 468 testing) with the columns: question, question type, context, and correct answer; it was saved as the `qasa.csv` file.

The code used to populate `qasa.csv` and `qasper.csv` with answers generated by the Iris Chat Tool is located in the `\populate data: generate answers"` folder of the `\data"` section in the repository, under the filename `\generate_answer_iris.py"`. This script generates answers for questions based on provided contexts using the Iris model deployed on AWS SageMaker. The process begins with loading AWS credentials and configuring the Llama 2 model. The script then defines data structures to handle examples with context and annotated examples, reads data from CSV files, generates answers for each example, and post-processes these answers to remove unnecessary text. Finally, the script saves the annotated examples with model-generated answers back into CSV files, specifically `populated_qasper.csv` and `populated_qasa.csv`. Additionally, these files were cleaned to remove initial phrases generated by the model that were not part of the actual answers, including: `\Sure! Here's the answer to your question:"`, `\Sure! Here's my answer:"`, `\The answer is:"`, and `\Sure! Here's the answer to your question based on the given/provided context:"`.

M.4 Sensitivity Prompts

Table 14 provides the prompts utilized on the generation of the sensitivity dataset used on Experiment 2.

Element	Generation Prompt	Validation Prompt
Fact Checking question	Given the following abstract from a scientific paper, generate three fact-checking questions that can be answered with a specific piece of text from this abstract. Each question should be directly answerable with a specific line or statement from the abstract. Abstract: {abstract} Please generate the three fact-checking questions and return a list containing them as strings. Do not include the answers.	You will be given a context and a question. Your task is to categorize the question according to the types listed below, based solely on the relationship between the question and the provided context. • fact: the answer is a composition based on only one substring of the context, that can be used directly as it is written in the context. • reasoning: answer is a composition based on a more elaborated understanding made from more than one substring of the context. substrings can not be used exactly as they are written, it requires a deeper understanding of the content. • unrelated question: the answer cannot be derived from the context provided. Context: {abstract} Question: {question} Please provide only the category name of the question based on the definitions provided.
Reasoning question	Given the following abstract from a scientific paper, generate three deep reasoning questions that require synthesis of multiple parts of the abstract to form a comprehensive answer. These questions should necessitate an understanding of the overall content, and the answer should include a paraphrase of more than one substring of the abstract. Abstract: {abstract} Please generate the three deeper reasoning questions and return a list containing them as strings. Do not include the answers.	
Correct perfect	You will be given an abstract from a scientific paper and a question related to it. Considering the abstract, generate a short answer for the question. If the answer is only a name or a number write a full sentence, do not return only one name or number. Abstract: {abstract} Question: {question} Please output only the accurate short answer.	-no validation-
Correct similar	You will be given a sentence. Your task is to write a new sentence that is a paraphrase similar to the original one. It should keep the same meaning but using some different words. Make sure that the original and the new sentence share at least 3 content words. Sentence: {ground_truth}	Function that: • Tokenize the two answers (correct perfect and correct similar or different) • Count the overlap of tokens between them • Returns True if correct similar has 3 or more overlap tokens, and correct different has 2 or less overlap tokens. Otherwise, returns False.
Correct different	You will be given a sentence. Your task is to write a new sentence that is a paraphrase completely different to the original one. It should keep the same meaning but using completely different words. Make sure that the original and the new sentence do not have any content word in common, not even 1. Example 1: Original sentence: The car-trained network showed a drop in performance for inverted versus upright cars. New sentence: This CNN demonstrated decreased behavior with upside-down items in comparison to those oriented correctly. Example 2: Original sentence: This paper tries to demonstrate, first, that the behavioral signatures associated to human face recognition can be explained as a result of the optimization of the task. Also, they show that it is not so "special", as this behavior can be found in CNNs trained to recognize other objects (like cars). New sentence: This article aims to demonstrate that the distinct characteristics linked to recognizing people's visages may stem from enhancement of the activity. Furthermore, it illustrates that this pattern is not unique, as similar ones occur in neural networks developed to identify different items, such as vehicles. Your task: Original sentence: {ground_truth} New sentence: Please output only the new sentence.	
Incorrect similar	You will be provided a question. Your task is to write a short sentence on the topic of the question that is not the answer. Question: {question} Please output only your answer.	You will be presented with a question, its correct answer, and a candidate sentence. Your task is to categorize the candidate sentence based on how it relates to the correct answer and the content of the question. Choose from the categories below: • Incorrect Similar: The sentence resembles the correct answer but contains incorrect information or misinterpretations. • Incorrect Related: The sentence is topically related to the question but does not address or correctly answer it. • Incorrect Unrelated: The sentence has no relevance to the topic or context of the question. • Rigth: The sentence is a possible rigth answer to the question. Question: {question} Correct Answer: {ground_truth} Candidate Sentence: {answer} Please provide only the category name for the candidate sentence based on the definitions provided.
Incorrect related	You will be provided a question and its correct answer. Your task is to make a small change on the answer to make it completely wrong. The new answer should be very similar to the original one, but being completely wrong. Question: {question} Answer: {answer} Please output only your answer.	

Table 14: **Sensitivity Dataset Prompt.** This table presents the prompts used to generate data on the Sensitivity Dataset generation module.

name	year	authors	size	description	based on	source questions	source answers	type / length
QASPER	**2021**	Allen Institute	5K	andcrafted QA based on scientific papers	NLP papers	Questions were written by NLP practitioners after reading only the title and abstract of the paper.	Answers were provided by a separate set of NLP practitioners who also provided supporting evidence.	nformation-seeking QA over academic research papers
QASA	**2023**	LG AI Research	1798 QA pairs	A handcrafted by experts from Scientific papers	AI/ML papers	collected from regular readers of AI/ML papers and the authors of the papers themselves. Annotators read entire papers and generate questions based on a schema that includes surface, testing, and deep questions.	provided by AI/ML experts who read the relevant sections of the papers. They generate evidential rationales from selected paragraphs and compose comprehensive answers.	advanced question answering tasks requiring full-stack reasoning on scientific articles, specifically in AI and ML fields
BioASQ	**2022**	National Center for Scientific Research (NCSR) "Demokritos"	4,721	handcrafted by experts / biomedical / unlike most previous QA benchmarks that contain only exact answers, the BioASQ-QA dataset also includes ideal answers (in effect summaries)	MEDLINE, various biomedical ontologies, and linked data sources	-		Yes/no, factoid, list, and summary questions
MC test	**2013**	Microsoft Research	2k	question based on fictional story and 4 short answers (multiple choice). 1 correct and 3 distractors.	fictional stories	crowd-sourcing on Amazon Mechanical Turk (AMT)	crowd-sourcing on Amazon Mechanical Turk (AMT)	
SQuAD	**2016**	Standford	107,785 QA pairs	question base on Wikipedia passage, with short answer.	Wikipedia	crowdworkers based on a collection of Wikipedia articles	spans of text from the corresponding articles.	factual, descriptive, and explanatory questions across domains. short (one word/idea)
SQuAD-V2	**2018**		50k	adds un-answerable				Un-answerable!
HotpotQA	**2018**	Standford and Google AI	113k	question base on Wikipedia passage, with short answer.	Wikipedia	crowd workers based on Wikipedia articles.	derived from multiple supporting Wikipedia documents, provided by crowd workers.	factoid comparison questions, chain reasoning, and yes/no questions.
ELI5	**2019**	Facebook AI Research	270K	QA pair from reddit. Long-form.	Reddit	Reddit forum "Explain Like I'm Five"		
Natural Questions	**2019**	Google Research		google search questions + wikipedia answers	Wikipedia	The questions are real anonymized, aggregated queries issued to the Google search engine. Simple heuristics are used to filter questions from the query stream, focusing on longer queries as they are more complex	Annotators use a custom interface to identify long and short answers within Wikipedia pages.	factual information across various domains
GOOAQ	**2021**	Allen Institute	3 M	google questions and answers	Google	Questions from Google search autocomplete	Answers from Google answer boxes.	
TruthfulQA	**2021**	Oxford and OpenAI	817	handcrafted QA pairs, with true and false answer	Magazines and Wikipedia	Crafted by authors; sources include Wikipedia, Cleveland Clinic, National Geographic		
Science QA	**2022**	Prague and India Unis	100,000	generated + manually cheked	Articles	The questions were generated using an answer-aware question generator (QG) model trained on a combination of SQuAD and ScholarlyRead datasets. The generated questions were then manually checked by human annotators.	The answers were derived from noun phrases in the abstracts of research articles, processed through the Stanford constituency parser. Implausible phrases were discarded, and plausible answers were manually evaluated.	factoid questions. answer is just a extraction of the text.
AE	**2022**	Google		Q + truth + generated answers + human annotation (answer equivalence)				Annotators were presented with the question, context, reference answer, and candidate answer. They answered a series of questions to determine if the candidate answer was completely different, equivalent, removed important information, or added misleading/superfluous information
ASQA	**2023**	Google Research + Unis	6k	long form answer to ambiguos questoin.		from AMBIGQA dataset	crowdsourced	long-form answers to ambiguous factoid questions
SciEval	**2023**	Shanghai Jiao Tong University	18k	generated + manually checked	Socratic QA (website where people ask questions)	Socratic Q&A, MedQA, PubMedQA, Reagent Selection from ChemLLMBench, and dynamically generated data based on scientific principles	ground-truth responses using GPT-4 for simplification and generation of multiple-choice options, with manual verification for quality assurance.	covers four dimensions of scientific research ability: Basic Knowledge, Knowledge Application, Scientific Calculation, and Research Ability. Questions include multiple-choice, fill-in-the-blank, judgment questions, and subjective experimental questions.
EVOUNA	**2023**	Amazon	5k	Q from other data + generated ans		Natural Questions (NQ) and TriviaQA (TQ)	answers generated by FiD, GPT-3.5, ChatGPT-3.5, ChatGPT-4, and BingChat models.	Open-domain factual questions

Table 13: Information about some considerate datasets.

Appendix Results

R.1 Statistics QASPER and QASA

The following tables show the results for the first analysis of the Experiment 1 and 2, for QASPER and QASA datasets separately, Tables 15 and 16 respectively.

Metrics	aligned vs. shuffled			8b vs. 70b parameters		
	difference	Cohen's d	Win-rate %	difference	Cohen's d	Win-rate %
Bleu	**-0,70**	**-0,72**	41	**-5,00**	**-0,22**	7
Rouge-1	**0,01**	**0,07**	50	**-0,13**	**-0,22**	10
Rouge-2	**0,01**	**0,56**	30	**-0,05**	**-0,21**	6
Rouge-L	**0,05**	**1,27**	68	**-0,12**	**-0,20**	11
Chrf	**-2,71**	**-0,25**	37	**-15,73**	**-0,21**	21
Chrf++	**-2,26**	**-0,26**	38	**-15,28**	**-0,20**	17
Meteor	**0,07**	**1,27**	76	**-0,13**	**-0,24**	11
Ter	**0,00**	**1,16**	66	**-0,01**	**-0,34**	7
Bert	**0,30**	**1,99**	89	**-0,26**	**-0,27**	17
WMS	0,25	1,02	81	**-0,18**	0,01	26
SMS	0,39	2,20	92	**-0,25**	**-0,25**	18
Wisdm	0,25	1,32	83	**-0,02**	**-0,10**	27
Bleurt	0,29	1,56	96	**-0,02**	**-0,06**	38
BEM	0,38	1,68	94	**-0,02**	**-0,06**	38
Prometheus	2,41	4,25	94	**-0,23**	**-0,19**	15
Bart	0,01	0,61	77	0,00	**-0,03**	35
LLM	0,78	5,54	99	**-0,06**	**-0,13**	20
Faithfulness	0,58	1,89	83	0,79	0,21	19
Relevancy	0,15	1,01	95	0,04	0,10	35
Correctness	0,16	0,73	74	**-0,02**	**-0,08**	42
R Similarity	0,07	1,74	91	**-0,01**	**-0,23**	36
Consistency	0,68	2,37	88	**-0,16**	**-0,13**	11
T Similarity	2,67	2,37	88	**-0,95**	**-0,19**	13

Table 15: **QASPER results.** This table summarizes the performance of the evaluation metrics comparing aligned vs. shuffled data and the 8 billion vs. 70 billion parameter on the QASPER dataset. The results include the mean difference, Cohen's d, and win rate percentage for each metric (under 30%: red, 30-60%: yellow, over 60%: green), highlighting their ability to discriminate between better and worse conditions.

Metrics	aligned vs. shuffled			8b vs. 70b parameters		
	difference	Cohen's d	Win-rate %	difference	Cohen's d	Win-rate %
Bleu	6,22	0,89	80	**-1,36**	**-0,29**	17
Rouge-1	0,19	1,13	83	**-0,06**	**-0,42**	18
Rouge-2	0,10	1,06	70	**-0,02**	**-0,24**	13
Rouge-L	0,16	1,09	83	**-0,04**	**-0,38**	20
Chrf	18,25	1,07	86	**-5,60**	**-0,35**	26
Chrf++	17,55	1,12	87	**-5,03**	**-0,35**	27
Meteor	0,18	1,13	83	**-0,04**	**-0,34**	19
Ter	0,00	**-0,04**	64	0,00	**-0,39**	13
Bert	0,47	2,21	93	**-0,09**	**-0,34**	25
WMS	0,23	1,25	88	**-0,01**	**-0,36**	38
SMS	0,42	2,06	94	**-0,08**	**-0,32**	26
Wisdm	0,26	1,49	87	**-0,05**	**-0,30**	30
Bleurt	0,29	1,56	96	**-0,03**	**-0,10**	33
BEM	0,38	1,68	94	**-0,03**	**-0,10**	33
Prometheus	2,35	3,56	91	**-0,26**	**-0,24**	14
Bart	0,02	0,87	80	0,00	-0,08	31
LLM	0,78	5,08	98	0,02	0,04	26
Faithfulness	0,54	1,39	76	**-0,01**	**-0,04**	9
Relevancy	0,17	0,76	94	**-0,02**	**-0,06**	34
Correctness	0,16	0,73	73	**-0,04**	**-0,15**	41
R Similarity	0,13	2,27	94	**-0,02**	**-0,33**	28
Consistency	0,00	2,56	89	**-0,07**	**-0,13**	11
T Similarity	2,72	2,35	89	**-0,50**	**-0,30**	15

Table 16: **QASA results.** Performance of the evaluation metrics comparing aligned vs. shuffled data and the 8 billion vs. 70 billion parameter on the QASA dataset. The results include the mean difference, Cohen's d, and win rate percentage for each metric (under 30%: red, 30-60%: yellow, over 60%: green)

R.2 Examples Instances.

The following images provide the examples obtained when filtering the QASPER and QASA datasets to analyze the top 3 instances with the highest and lowest general score (average among all metrics); and the highest and lowest std based on the general score.

Ex.	Question	Reference Answer	Generated Answer
1	Where does the data come from?	['crowsourcing platform', 'ins scope data collectioncrowd workers provide questions commands related topic domains additional data rephrase scenario crowdsourcing tasks proposed bibref used . scope data collection workers mistakesqueries written one intents actually match intents using scoping scenario tasks prompts based topic areas found quora wikipedia elsewhere .']	Based on the text, the data comes from two sources:\n\n1. Scoping crowdsourcing tasks: The data is collected using crowdsourcing tasks that are designed to gather questions and commands related to specific topic domains. The tasks are designed to simulate the way a user might interact with an artificially intelligent assistant.\n2. Outside sources: The data is also collected from two outside sources:\n\t* Quora and Wikipedia: These sources are used to gather additional out-of-scope data by using the topic areas found on these platforms.\n\t* Elsewhere: This refers to other sources of data that are used to ensure the richness of the out-of-scope data.\n\nThe data is manually grouped and generated using the scoping tasks and intents, and the team has a limited budget to gather the data. The data is validated and tested using training and validation queries per intent
2	Which datasets are used?	['existential oneshape multishapes spacial twoshapes multishapes quantification count ratio datasets generated shapeworldice', 'shapeworldice datasets oneshape multishapes twoshapes multishapes count ratio']	Based on the text, the following datasets are used:\n\n1. ShapeWorldice datasets\n2. GTD (Generalized Truth Discovery) framework\n3. Existential oneshape dataset\n4. Spatial twoshapes dataset\n5. Four datasets with multiple abstract objects present in the visual scene
3	What was the baseline for this task?	['baseline system slc task simple logistic regression classifier default parameters . baseline flc task generates spans selects one techniques randomly .', '']	Based on the text, the baseline for the task is a simple logistic regression classifier with default parameters, which is used to represent input instances with a single feature length sentence. This is mentioned in the first and third sentences of the text

Figure 31: Top 3 instances with the highest general score - QAPER.

Ex.	Question	Reference Answer	Generated Answer
1	What are pros and cons of these models illustrated in Figure 2, and what are distinctions of the proposed model?	Using figure 2, \nThese increasingly expressive architectures are in tension. While interaction-based models (i.e., Figure 2 (b) and (c)) tend to be superior for IR tasks (Guo et al., 2019; Mitraet al., 2018), a representation-focused model—by isolating the computations among q and d—makes it possible to pre-compute document representations offline (Zamani et al., 2018), greatly reducing the computational load per query. In this work, we observe that the fine-grained matching of interaction-based models and the pre-computation of document representations of representation-based models can be combined by retaining yet judiciously delaying the query–document interaction. Figure 2 (d) illustrates an architecture that precisely does so. As illustrated, every query embedding interacts with all document embeddings via a MaxSim operator, which computes maximum similarity (e.g., cosine similarity), and the scalar outputs of these operators are summed across query terms. This paradigm allows ColBERT to exploit deep LM-based representations while shifting the cost of encoding documents offline and amortizing the cost of encoding the query once across all ranked documents. Additionally, it enables ColBERT to leverage vector-similarity search indexes (e.g., (Johnsonet al., 2017; Abuzaidet al., 2019)) to retrieve the top-k results directly from a large document collection, substantially improving recall over models that only re-rank the output of term-based retrieval.\n\n\n\n\n\n\nThe distinction of proposed model : 1) a highly-effective model is proposed that employs novel BERT-based query and document encoders within the late interaction paradigm.(2) We show how to leverage ColBERT both for re-ranking on top of a term-based retrieval model (3) and for searching a full collection using vector similarity indexes.(4)We evaluate ColBERT on MS MARCO and TREC CAR, two recent passage search collections.	Pros of the interaction-based models (Figure 2(b) and (c)):\n\n* Superior for IR tasks (Guo et al., 2019; Mitra et al., 2018)\n\nCons of the interaction-based models:\n\n* Computationally expensive\n* Limited by the number of documents that can be processed in a single batch\n\nPros of the representation-focused model (Figure 2(d)):\n\n* Pre-computes document representations offline, reducing computational load per query\n* Allows for efficient vector-similarity search indexes (Johnson et al., 2017; Abuzaid et al., 2019)\n\nCons of the representation-focused model:\n\n* May not be as effective for IR tasks as interaction-based models\n\nDistinctions of the proposed model (ColBERT):\n\n* Combines the fine-grained matching of interaction-based models with the pre-computation of document representations of representation-based models\n* Allows for efficient retrieval of top-k results directly from a large document collection, improving recall over models that only re-rank the output of term-based retrieval
2	What is the difference of RocketQAv1 and RocketQAv2 model?	RocketQAv1 trains dual-encoder and cross-encoder in a cascade manner, which leverages the powerful cross-encoder to empower the dual-encoder. While it inherits the parameters from RocketQAv1, RocketQAv2 extends the first version through a novel approach that jointly trains the dense passage retriever and passage re-ranker, and by using a large PLM for data augmentation and denoising (i.e., a distillation procedure).	RocketQAv2 is an improved version of RocketQAv1.\n\nRocketQAv1 trains dual-encoder and cross-encoder in a cascade manner, while RocketQAv2 jointly trains the dense passage retriever and passage re-ranker. Additionally, RocketQAv2 utilizes a large PLM for data augmentation and denoising, which can be regarded as a distillation procedure. This allows RocketQAv2 to improve the effectiveness of the model by leveraging explicit external knowledge
3	What characteristics of large-scale pre-trained language models made it remarkable successful for passage re-ranking task?	Large-scale pre-trained language models (PLMs) have been found to be successful for passage re-ranking due to their ability to learn semantic relevance in the latent space from massive textual corpus. PLMs obtain this ability from their expressive transformer architecture and the pretrain-then-finetune paradigm.	The expressive transformer structure and the pretrain-then-finetune paradigm of large-scale pre-trained language models (PLMs) have contributed to their remarkable success on the passage re-ranking task

Figure 32: Top 3 instances with the highest general score - QASA.

Ex.	**Question**	**Reference Answer**	**Iris Answer**
1	How large is their MNER SnapCaptions dataset?	[", "]	Based on the information provided in the two texts, the MNER SnapCaptions dataset contains:\n\n* "k" user-generated image snap textual caption pairs (where k is a positive integer)\n* Named entities are manually labeled by expert human annotators\n* Entity types per loc org misc. (which suggests that the dataset includes a variety of entity types, such as locations, organizations, and miscellaneous entities)\n\nHowever, the exact size of the dataset is not specified. The texts only mention that the dataset contains "many new emerging entities found in various surface forms, nicknames, and typos," and that it includes "expert-annotated named entities."\n\nTherefore, without more information, it is not possible to determine the exact size of the MNER SnapCaptions dataset
2	What useful information does attention capture?	[", 'alignment points pos tags .']	Based on the given context, attention captures the following useful information:\n\n1. Information beyond alignment: The analysis shows that attention models can capture information beyond traditional alignment cases, providing a more comprehensive understanding of the relationship between words in a sentence.\n2. Translational equivalents: Attention agrees with traditional alignments for high-degree case nouns, but captures information about translational equivalents for verbs, including auxiliary verbs and adverbs.\n3. Word prediction loss: Attention loss is related to translation quality, and analyzing the relationship between attention loss and word prediction loss can provide insights into the impact of attention on translation accuracy.\n4. Distribution of attention: Attention is not equally distributed across all parts of speech, and the distribution of attention can vary substantially across different pos tags.\n5. Dependency roles: Attention can capture the dependency roles of words in a sentence, such as subject-verb-object relationships, and this information can be useful for improving translation accuracy.\n6. Translating nouns and verbs: Attention can help identify the roles of nouns and verbs in a sentence, including adjectives, determiners, and auxiliary verbs, which can be useful for improving the accuracy of translations
3	On how many language pairs do they show that preordering assisting language sentences helps translation quality?	[", "]	Based on the given text, they show that preordering assisting language sentences helps translation quality for 6 language pairs:\n\n1. English-Hindi\n2. Bengali-Hindi\n3. Gujarati-Hindi\n4. Marathi-Hindi\n5. Malayalam-Hindi\n6. Tamil-Hindi\n\nThese language pairs are all Indo-Aryan languages, and Hindi is the target language for all of them. The text mentions that preordering the assisting language sentences in Hindi improves the translation quality for these language pairs

Figure 33: Top 3 instances with the lowest general score - QAPER.

Ex.	Question	Reference Answer	Generated Answer
1	If both queries and documents are short, is still the fine-granular interaction required?	During indexing, we use another server with the same CPU and system memory specifications but which has four Titan V GPUs attached, each with 12 GiBs of memory. Across all experiments, only one GPU is dedicated per query for retrieval (i.e., for methods with neural computations) but we use up to all four GPUs during indexing.	No
2	What is the total number of filters in squeeze convolution layer?	s1x1 is the number of filters in the squeeze layer and it is set s1x1 to be less than (e1x1 + e3x3) -the total number of filters in expand layer of the fire module- to limit the number of input channels to the 3x3 filters.	1
3	Would the performance be improved if the PLM model is pre-trained or fine-tuned on bio-medical domain datasets?	In their experiments, the authors showed that all of the models performed poorly on the biomedical domain due to the textual data of the domain not being covered widely in the PLMs' pretraining dataset. This lack of data can cause the PLM to struggle to reveal and capture knowledge specific to that domain. These results suggest that further training on bio-medical data could increase performance.	Yes

Figure 34: Top 3 instances with the lowest general score - QASA.

Ex.	Question	Reference Answer	Generated Answer
1	Which baselines did they compare against?	['various tree structured neural networks including variants treelstm treebased cnn rntn nontree models including variants lstms cnns residual selfattention based networks', 'sentence classification baselines rntn socher et al . adamcrntn dong et al . terntn qian et al . tbcnn mou et al . treelstm tai socher manning adahtlstmcm liu qiu huang dctreelstm liu qiu huang telstm huang qian zhu bicontree teng zhang gumbel treelstm choi yoo lee treenet cheng et al . cnn kim adasent zhao lu poupart lstmcnn zhou et al . bytemlstm radford jozefowicz sutskever bcn char cove mccann et al . bcn char elmo peters et al . . stanford natural language inference baselines latent syntax treelstm yogatama et al . treebased cnn mou et al . gumbel treelstm choi yoo lee nse munkhdalai yu reinforced self attention network shen et al . residual stacked encoders nie bansal bilstm generalized pooling chen ling zhu .']	Based on the text you provided, the authors compared their model (SATa-STM) against the following baselines:\n\n1. Previous treestructured models (e.g., Tree-LSTM, Tree-CNN)\n2. Latent treestructured models (e.g., Latent Tree-LSTM, Latent Tree-CNN)\n3. Models pretrained on large external corpora (e.g., BERT, RoBERTa)\n4. Nontree models (e.g., BilSTM, Residual Stacked Encoders)\n\nThey also compared their model with a variant of the SATa-STM model that uses a different type of self-attention mechanism (Gumbel TreelSTM) and a variant that uses a different type of pooling mechanism (Generalized Pooling)
2	Which datasets are used?	['existential oneshape multishapes spacial twoshapes multishapes quantification count ratio datasets generated shapeworldice', 'shapeworldice datasets oneshape multishapes twoshapes multishapes count ratio']	Based on the text, the following datasets are used:\n\n1. ShapeWorldice datasets\n2. GTD (Generalized Truth Discovery) framework\n3. Existential oneshape dataset\n4. Spatial twoshapes dataset\n5. Four datasets with multiple abstract objects present in the visual scene
3	What was the baseline for this task?	['baseline system slc task simple logistic regression classifier default parameters . baseline flc task generates spans selects one techniques randomly .', '']	Based on the text, the baseline for the task is a simple logistic regression classifier with default parameters, which is used to represent input instances with a single feature length sentence. This is mentioned in the first and third sentences of the text

Figure 35: Top 3 instances with the highest std - QAPER.

Ex.	Question	Reference Answer	Generated Answer
1	What are pros and cons of these models illustrated in Figure 2, and what are distinctions of the proposed model?	Using figure 2, \nThese increasingly expressive architectures are in tension. While interaction-based models (i.e., Figure 2 (b) and (c)) tend to be superior for IR tasks (Guo et al., 2019; Mitraet al., 2018), a representation-focused model—by isolating the computations among q and d—makes it possible to pre-compute document representations offline (Zamani et al., 2018), greatly reducing the computational load per query. In this work, we observe that the fine-grained matching of interaction-based models and the pre-computation of document representations of representation-based models can be combined by retaining yet judiciously delaying the query–document interaction. Figure 2 (d) illustrates an architecture that precisely does so. As illustrated, every query embedding interacts with all document embeddings via a MaxSim operator, which computes maximum similarity (e.g., cosine similarity), and the scalar outputs of these operators are summed across query terms. This paradigm allows ColBERT to exploit deep LM-based representations while shifting the cost of encoding documents offline and amortizing the cost of encoding the query once across all ranked documents. Additionally, it enables ColBERT to leverage vector-similarity search indexes (e.g., (Johnsonet al., 2017; Abuzaidet al., 2019)) to retrieve the top-k results directly from a large document collection, substantially improving recall over models that only re-rank the output of term-based retrieval.\n\n\n\n\n\n\nThe distinction of proposed model : 1) a highly-effective model is proposed that employs novel BERT-based query and document encoders within the late interaction paradigm.(2) We show how to leverage ColBERT both for re-ranking on top of a term-based retrieval model (3) and for searching a full collection using vector similarity indexes.(4)We evaluate ColBERT on MS MARCO and TREC CAR, two recent passage search collections.	Pros of the interaction-based models (Figure 2(b) and (c)):\n\n* Superior for IR tasks (Guo et al., 2019; Mitra et al., 2018)\n\nCons of the interaction-based models:\n\n* Computationally expensive\n* Limited by the number of documents that can be processed in a single batch\n\nPros of the representation-focused model (Figure 2(d)):\n\n* Pre-computes document representations offline, reducing computational load per query\n* Allows for efficient vector-similarity search indexes (Johnson et al., 2017; Abuzaid et al., 2019)\n\nCons of the representation-focused model:\n\n* May not be as effective for IR tasks as interaction-based models\n\nDistinctions of the proposed model (ColBERT):\n\n* Combines the fine-grained matching of interaction-based models with the pre-computation of document representations of representation-based models\n* Allows for efficient retrieval of top-k results directly from a large document collection, improving recall over models that only re-rank the output of term-based retrieval
2	What is the difference of RocketQAv1 and RocketQAv2 model?	RocketQAv1 trains dual-encoder and cross-encoder in a cascade manner, which leverages the powerful cross-encoder to empower the dual-encoder. While it inherits the parameters from RocketQAv1, RocketQAv2 extends the first version through a novel approach that jointly trains the dense passage retriever and passage re-ranker, and by using a large PLM for data augmentation and denoising (i.e., a distillation procedure).	RocketQAv2 is an improved version of RocketQAv1.\n\nRocketQAv1 trains dual-encoder and cross-encoder in a cascade manner, while RocketQAv2 jointly trains the dense passage retriever and passage re-ranker. Additionally, RocketQAv2 utilizes a large PLM for data augmentation and denoising, which can be regarded as a distillation procedure. This allows RocketQAv2 to improve the effectiveness of the model by leveraging explicit external knowledge
3	What characteristics of large-scale pre-trained language models made it remarkable successful for passage re-ranking task?	Large-scale pre-trained language models (PLMs) have been found to be successful for passage re-ranking due to their ability to learn semantic relevance in the latent space from massive textual corpus. PLMs obtain this ability from their expressive transformer architecture and the pretrain-then-finetune paradigm.	The expressive transformer structure and the pretrain-then-finetune paradigm of large-scale pre-trained language models (PLMs) have contributed to their remarkable success on the passage re-ranking task

Figure 36: Top 3 instances with the highest std - QASA.

Ex.	Question	Reference Answer	Generated Answer
1	What useful information does attention capture?	['', 'alignment points pos tags .']	Based on the information provided in the two texts, the MNER SnapCaptions dataset contains:\n\n* "k" user-generated image snap textual caption pairs (where k is a positive integer)\n* Named entities are manually labeled by expert human annotators\n* Entity types per loc org misc. (which suggests that the dataset includes a variety of entity types, such as locations, organizations, and miscellaneous entities)\n\nHowever, the exact size of the dataset is not specified. The texts only mention that the dataset contains "many new emerging entities found in various surface forms, nicknames, and typos," and that it includes "expert-annotated named entities."\n\nTherefore, without more information, it is not possible to determine the exact size of the MNER SnapCaptions dataset
2	Which other approaches do they compare their model with?	['akbik et al . link et al .', 'compare akbik et al . link et al . .']	Based on the given context, attention captures the following useful information:\n\n1. Information beyond alignment: The analysis shows that attention models can capture information beyond traditional alignment cases, providing a more comprehensive understanding of the relationship between words in a sentence.\n2. Translational equivalents: Attention agrees with traditional alignments for high-degree case nouns, but captures information about translational equivalents for verbs, including auxiliary verbs and adverbs.\n3. Word prediction loss: Attention loss is related to translation quality, and analyzing the relationship between attention loss and word prediction loss can provide insights into the impact of attention on translation accuracy.\n4. Distribution of attention: Attention is not equally distributed across all parts of speech, and the distribution of attention can vary substantially across different pos tags.\n5. Dependency roles: Attention can capture the dependency roles of words in a sentence, such as subject-verb-object relationships, and this information can be useful for improving translation accuracy.\n6. Translating nouns and verbs: Attention can help identify the roles of nouns and verbs in a sentence, including adjectives, determiners, and auxiliary verbs, which can be useful for improving the accuracy of translations
3	How large is their MNER SnapCaptions dataset?	['', '']	Based on the given text, they show that preordering assisting language sentences helps translation quality for 6 language pairs:\n\n1. English-Hindi\n2. Bengali-Hindi\n3. Gujarati-Hindi\n4. Marathi-Hindi\n5. Malayalam-Hindi\n6. Tamil-Hindi\n\nThese language pairs are all Indo-Aryan languages, and Hindi is the target language for all of them. The text mentions that preordering the assisting language sentences in Hindi improves the translation quality for these language pairs

Figure 37: Top 3 instances with the lowest std - QAPER.

Ex.	Question	Reference Answer	Generated Answer
1	If both queries and documents are short, is still the fine-granular interaction required?	During indexing, we use another server with the same CPU and system memory specifications but which has four Titan V GPUs attached, each with 12 GiBs of memory. Across all experiments, only one GPU is dedicated per query for retrieval (i.e., for methods with neural computations) but we use up to all four GPUs during indexing.	No
2	How does the performance change when a dense retriever is evaluated on out-of-domain queries and documents that are different from the domain on which the retriever was trained?	It is said that when evaluating a retriever trained on a source domain in an out-of-domain setting, the performance is obtained lower than BM25. Also, dense retrievers are said to be sensitive to domain shift and models that perform well on MS MARCO do not perform well on COVID-19 data. There have been many studies on unsupervised sentence embedding learning, but it is said that they do not work well in unsupervised dense retrieval. Therefore, the performance of the retriever in out-of-domain may be worse.	Poorly
3	What is the total number of filters in squeeze convolution layer?	s1x1 is the number of filters in the squeeze layer and it is set s1x1 to be less than (e1x1 + e3x3) -the total number of filters in expand layer of the fire module- to limit the number of input channels to the 3x3 filters.	1

Figure 38: Top 3 instances with the lowest std - QASA.

R.3 Spearman Correlation QASPER and QASA

The following figures show the heatmaps of the last analysis on Experiment 1 for QASPER and QASA datasets separately, Figures 39 and 40 respectively.

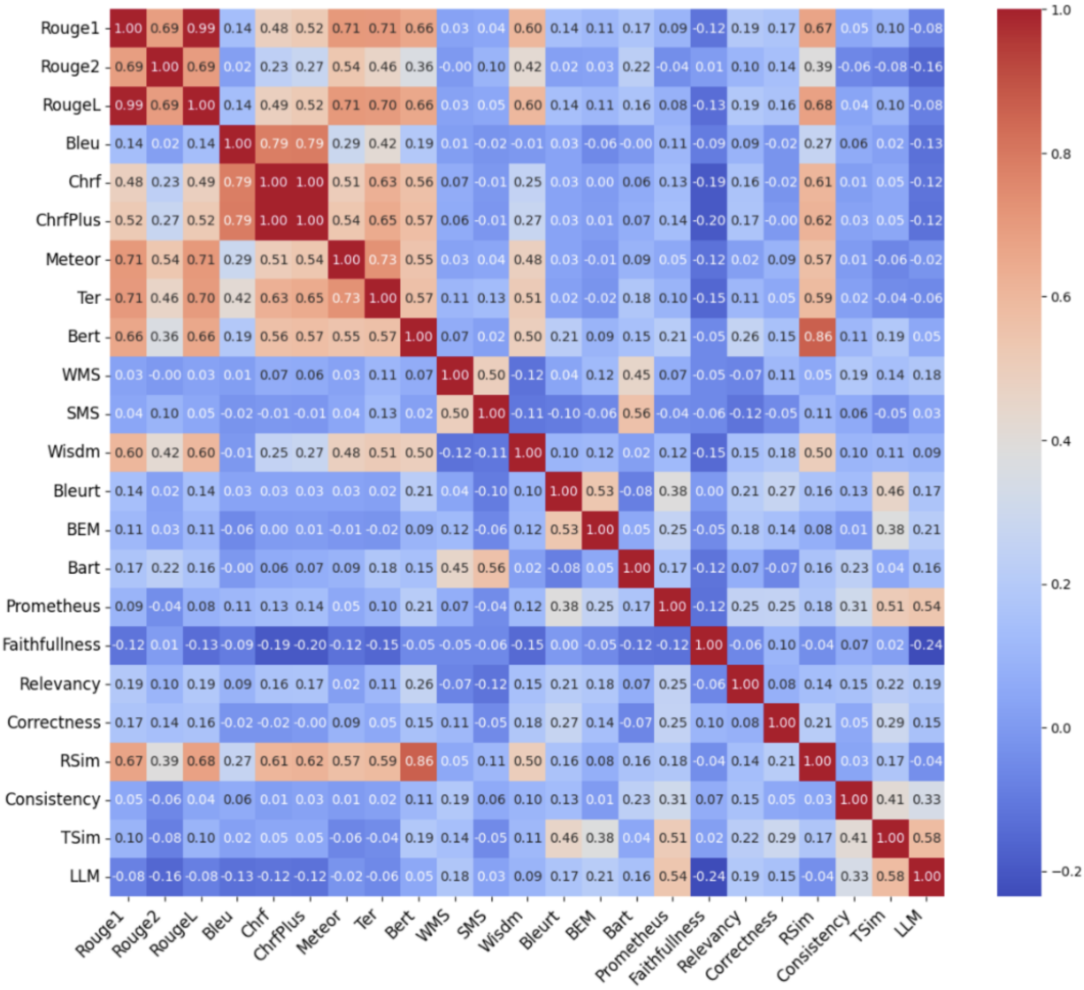

Figure 39: **Spearman Correlation QASPER dataset.** Spearman correlation heatmap for the QASPER dataset, showing the relationships between various evaluation metrics applied to answers generated by the LLAMA 2 model.

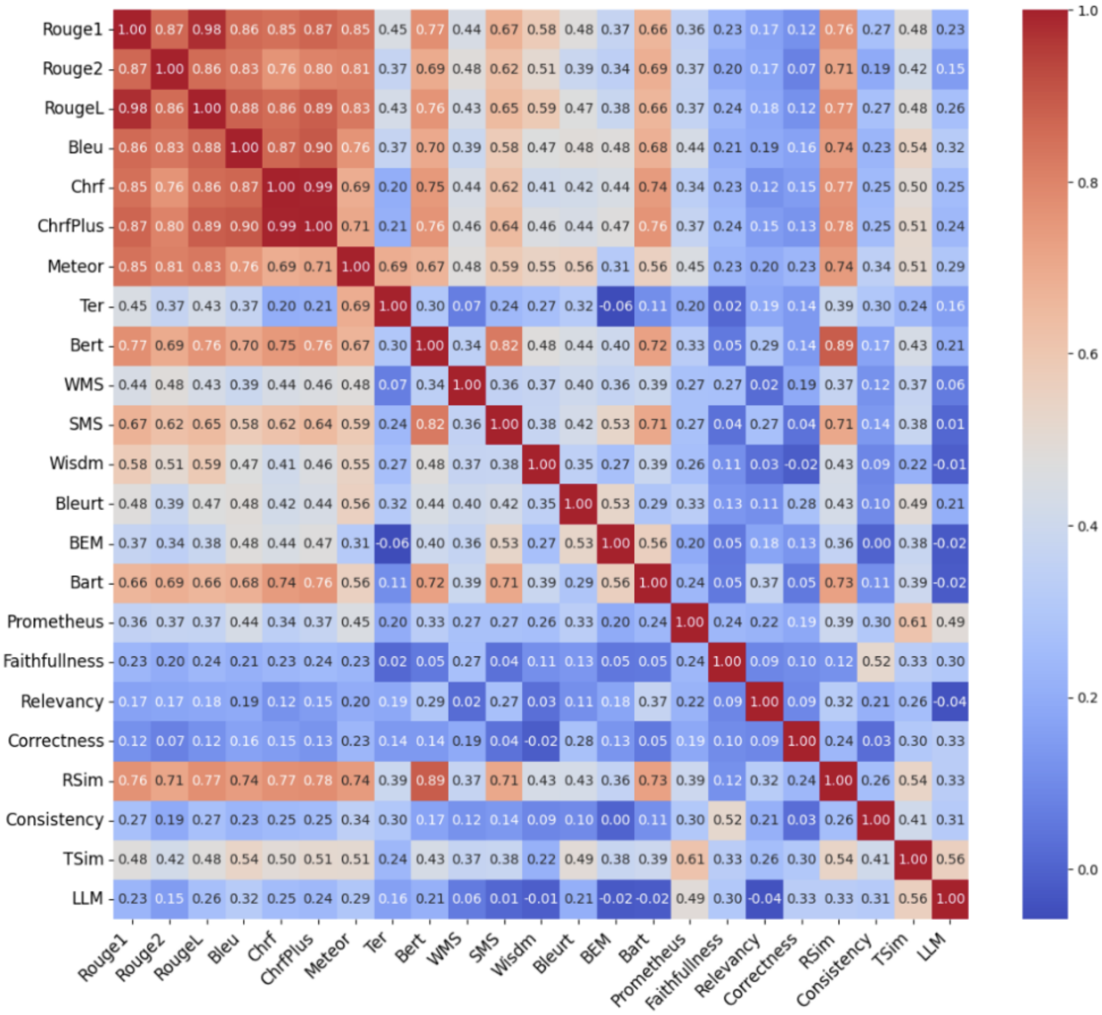

Figure 40: **Spearman Correlation QASA dataset.** Spearman correlation heatmap for the QASA dataset, showing the relationships between various evaluation metrics applied to answers generated by the LLAMA 2 model.

List of Figures

List of Tables